VPC-DI System Administration Guide, Release 21.16

First Published: 2019-10-24

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

CONTENTS

About this Guide

This preface describes the *VPC-DI System Administration Guide*, how it is organized and its document conventions.

Cisco Virtualized Packet Core Distributed Instance (VPC-DI) consists of a fully distributed network of multiple virtual machines (VMs) grouped to form a single StarOS instance with VMs performing management, input/output (I/O), and packet processing. The VMs run on commercial off-the-shelf (COTS) servers. This guide describes how to configure and administer the various components of the VPC-DI.

Note Control and User Plane Separation (CUPS) represents a significant architectural change in the way StarOS-based products are deployed in the 3G, 4G, and 5G networks. Unless otherwise specified, it should not be assumed that any constructs (including, but not limited to, commands, statistics, attributes, MIB objects, alarms, logs, services) referenced in this document imply functional parity with CUPS products. References to any CUPS products or features are for informational purposes only. Please contact your Cisco Account or Support representative for any questions about parity between this product and any CUPS products.

Note The documentation set for this product strives to use bias-free language. For purposes of this documentation set, bias-free is defined as language that does not imply discrimination based on age, disability, gender, racial identity, ethnic identity, sexual orientation, socioeconomic status, and intersectionality. Exceptions may be present in the documentation due to language that is hardcoded in the user interfaces of the product software, language used based on RFP documentation, or language that is used by a referenced third-party product.

Note The current release does not comply fully with Cisco's Security Development Lifecycle. Cisco has analyzed and identified the security vulnerabilities related to this release and closed the high-impacting vulnerabilities. Vulnerabilities will be disclosed in accordance with Cisco's Security Vulnerability Policy.

This guide describes how to generally configure and maintain StarOS running on an virtualized platform. It also includes information on monitoring system performance and troubleshooting. Supplemental information related to general StarOS operation and supported network gateway functions can be found in the StarOS documentation.

- Conventions Used, on page xxxii
- Related Documentation, on page xxxiii

• Contacting Customer Support, on page xxxiv

Conventions Used

The following tables describe the conventions used throughout this documentation.

Notice Type	Description
Information Note	Provides information about important features or instructions.
Caution	Alerts you of potential damage to a program, device, or system.
Warning	Alerts you of potential personal injury or fatality. May also alert you of potential electrical hazards.

Typeface Conventions	Description
Text represented as a `screen display`	This typeface represents displays that appear on your terminal screen, for example: `Login:`
Text represented as **commands**	This typeface represents commands that you enter, for example: **show ip access-list** This document always gives the full form of a command in lowercase letters. Commands are not case sensitive.
Text represented as a **command** *variable*	This typeface represents a variable that is part of a command, for example: **show card** *slot_number* *slot_number* is a variable representing the desired chassis slot number.
Text represented as menu or sub-menu names	This typeface represents menus and sub-menus that you access within a software application, for example: Click the **File** menu, then click **New**

Command Syntax Conventions	Description
{ keyword or *variable* **}**	Required keyword options and variables are those components that are required to be entered as part of the command syntax. Required keyword options and variables are surrounded by grouped braces { }. For example: **sctp-max-data-chunks { limit** *max_chunks* **| mtu-limit }** If a keyword or variable is not enclosed in braces or brackets, it is mandatory. For example: **snmp trap link-status**
[keyword or *variable* **]**	Optional keywords or variables, or those that a user may or may not choose to use, are surrounded by brackets.
|	Some commands support multiple options. These are documented within braces or brackets by separating each option with a vertical bar. These options can be used in conjunction with required or optional keywords or variables. For example: **action activate-flow-detection { intitiation | termination }** or **ip address [count** *number_of_packets* **| size** *number_of_bytes* **]**

Related Documentation

The most up-to-date information for this product is available in the product *Release Notes* provided with each software release.

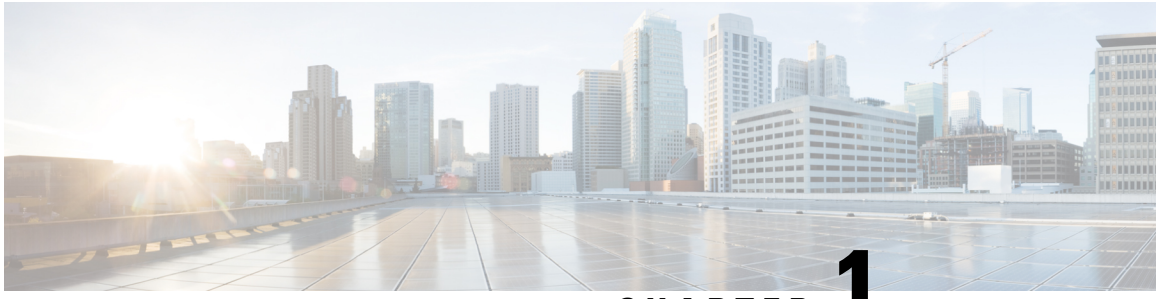
The following user documents are available on www.cisco.com:

- *AAA Interface Administration and Reference*
- *Command Line Interface Reference*
- *GTPP Interface Administration and Reference*
- *IPSec Reference*
- *Release Change Reference*
- *SNMP MIB Reference*
- *Statistics and Counters Reference*
- *Thresholding Configuration Guide*
- Product-specific and feature-specific Administration guides

Contacting Customer Support

Use the information in this section to contact customer support.

Refer to the support area of http://www.cisco.com for up-to-date product documentation or to submit a service request. A valid username and password are required to access this site. Please contact your Cisco sales or service representative for additional information.



CHAPTER 1

Introduction to VPC-DI

This chapter introduces Cisco Virtualized Packet Core—Distributed Instance (VPC-DI). VPC-DI supports scalability for virtualized cloud architectures by extending the boundaries beyond a single virtual machine (VM).

Product Description

This chapter describes the StarOS VPC-DI architecture and interaction with external devices.

VPC-DI distributes the virtualized StarOS beyond a single Virtual Machine (VM), enabling multiple VMs to act as a single StarOS instance with shared interfaces, shared service addresses, load balancing, redundancy, and a single point of management.

The system operates as a fully distributed network of multiple VMs grouped to form a StarOS instance, with VMs performing management and session processing with a standby VM for each type.

For UGP with Control and User Plane Separation (CUPS) VNF deployments, the CF and associated SF VNFCs comprise the Control Plane Function (CPF). The SAEGW service is used to provide Control Plane functionality for Sxa, Sxb, and Sxab (combined Sxa and Sxb) interfaces.

For CUPS-based VNF deployments, the UPF functions as an SAEGW-U. It is deployed as a VNFC running a single, stand-alone instance of the StarOS. Multiple UPF VNFCs can be deployed for scalability based on your deployment requirements.

For Day-N configuration procedures, refer to the *Ultra Packet Core CUPS Control Plane Administration Guide* and *Ultra Packet Core CUPS User Plane Administration Guide*.

Underlying Infrastructure for the System

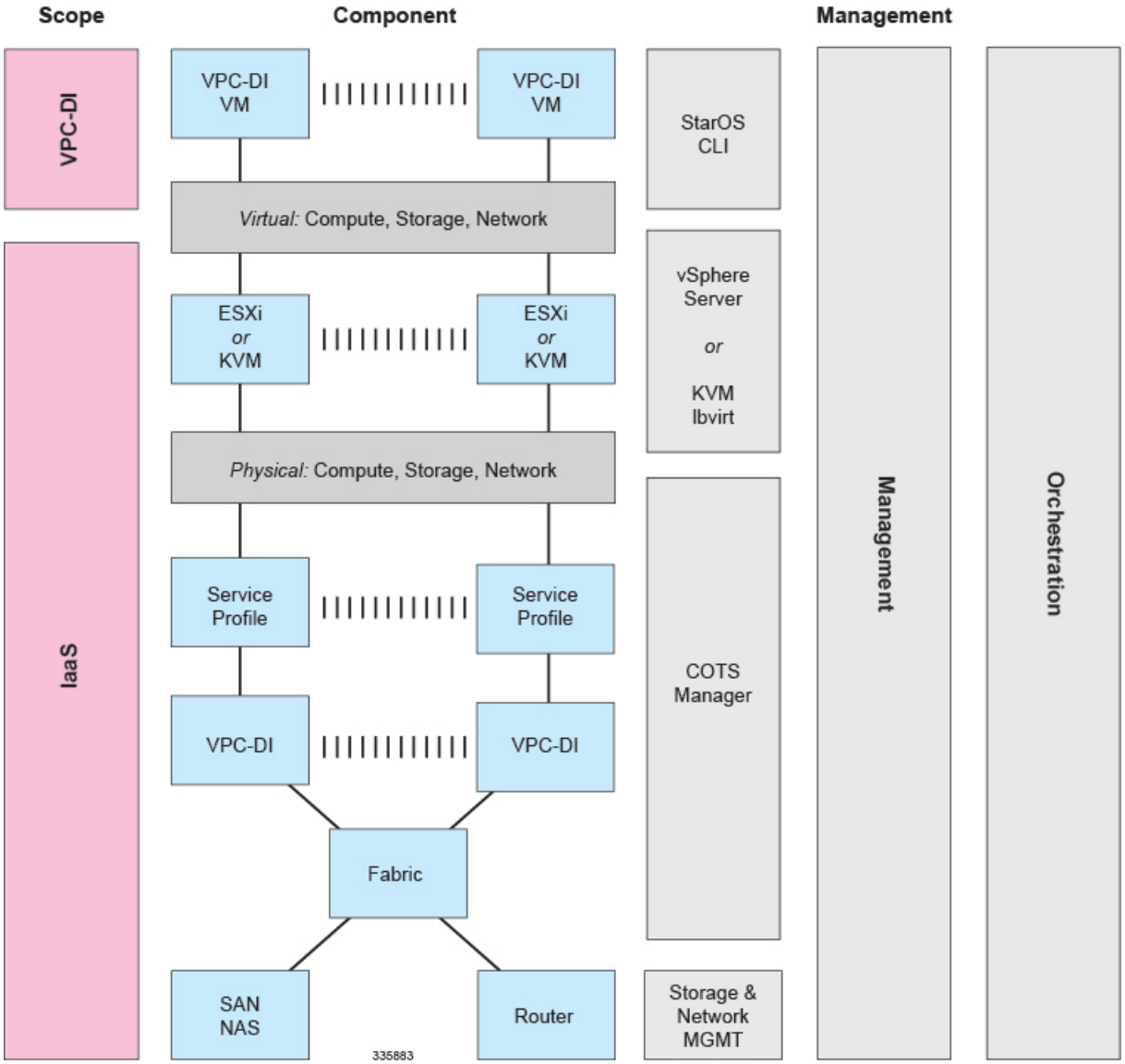
This virtualized system can be deployed into a new or existing Infrastructure as a Service (IaaS) cloud datacenter. VPC-DI runs in a set of virtual machines (VMs) using industry standard hypervisors on Commercial Off The Shelf (COTS) servers. This deployment model allows the management of physical infrastructure to remain outside of the scope of StarOS and VPC-DI.

A typical instance runs on the following NFVi (network function virtual infrastructure):

- IaaS components
  - COTS blade chassis, blades, and fabric interconnects
  - Manager Software
  - Network Attached Storage (NAS) or Storage Area Network (SAN)
  - VMware ESXi or KVM hypervisor on each blade or server
  - VMware vSphere Server or KVM OpenStack
- StarOS software installed within VMs
- Each VM must run on a separate blade so if a blade fails, only a single VM is affected (anti-affinity).
- Existing Management software (service, logging, statistics, etc.)
- Orchestration software (optional) [See Orchestration, on page 16]

A VPC-DI instance is a grouping of VMs that act as a single manageable instance of StarOS. VPC-DI consists of the following major components:

Figure 1: VPC-DI Scope



• Control Function (CF) VMs, on page 3
• Service Function (SF) VMs, on page 4
• DI Network, on page 5

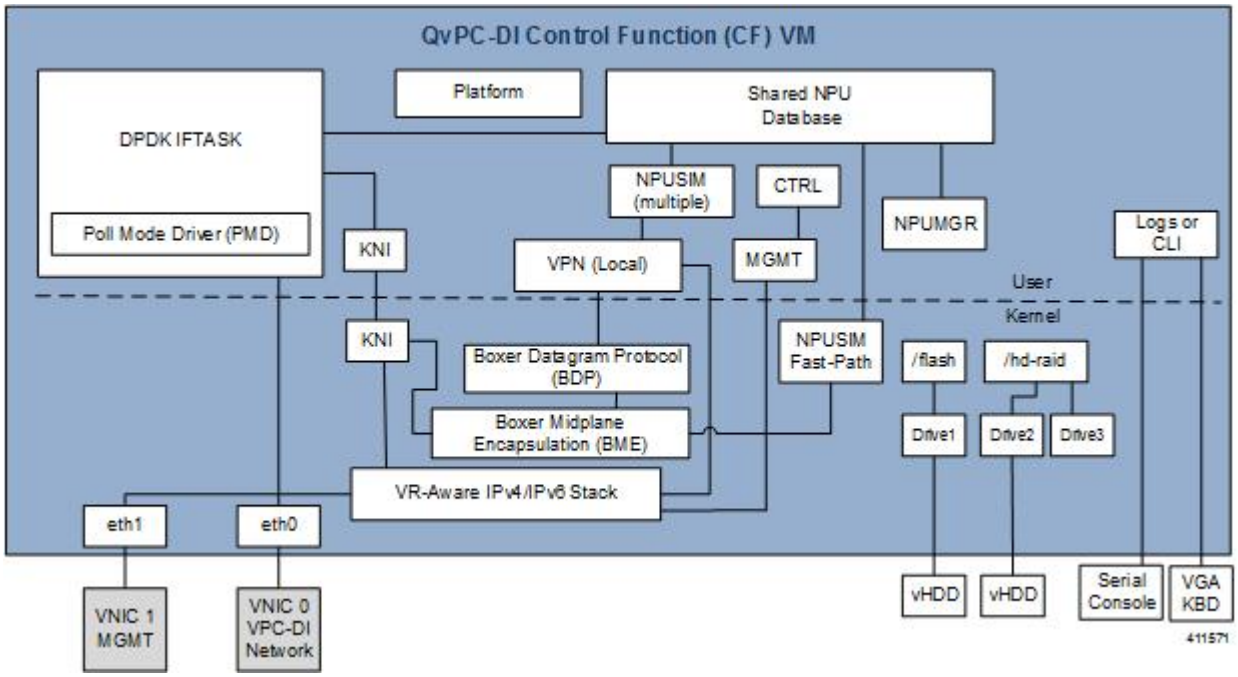
Control Function (CF) VMs

Two CF VMs act as an active:standby (1:1) redundant pair. The active CF is responsible for the following functions:

• Controller tasks

- Local context VPNMGR
- Local context (MGMT) and DI-Network vNICs
- System boot image and configuration storage on vHDD
- Record storage on vHDD
- Out-of-Band (OOB) management (vSerial and vKVM) for CLI and logging

Figure 2: Control Function VM



Service Function (SF) VMs

SF VMs provide service context (user I/O ports) and handle protocol signaling and session processing tasks. A VPC-DI instance can have a minimum of 4 SF VMs and maximum of 14 SF VMs, of which a maximum of 12 SF VMs can be active.

Each SF VM dynamically takes on one of three roles as directed by the CF:

- Demux VM (flow assignments)
- Session VM (traffic handling)
- Standby VM (n+1 redundancy)

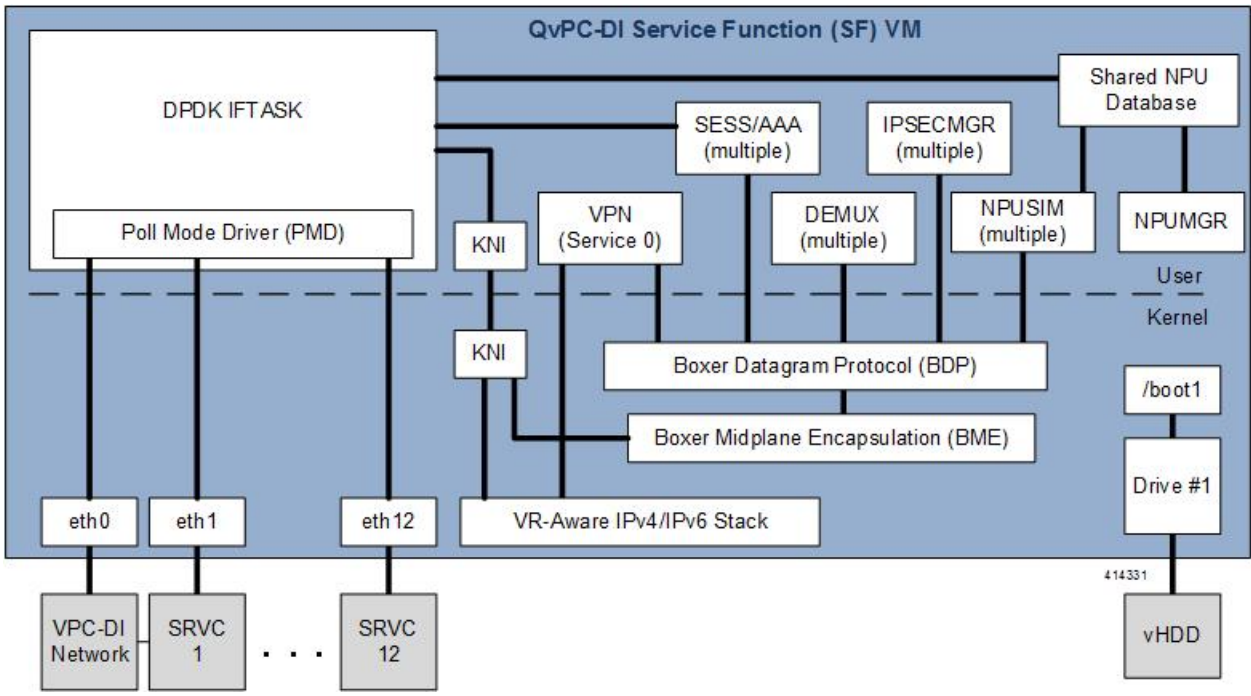
An SF is responsible for the following functions:

Function	Location Where Runs
NPUSIM fastpath/slow path (NPU emulation and routing to CPU)	Demux VM, Session VM, Standby VM

Function	Location Where Runs
IFTASK based on the Intel® Data Plane Development Kit (DPDK)	Demux VM, Session VM, Standby VM
Non-local context (SRVC) vNIC ports	Demux VM, Session VM, Standby VM
VPNMGR and Demux for service contexts (first VM)	Demux VM
SESSMGR and AAAMGR for session processing (additional VMs)	Session VM
Egress forwarding decisions	
Crypto processing	

A minimum configuration for a VPC-DI instance requires four SFs two active, one demux and one standby.

Figure 3: Service Function VM



DI Network

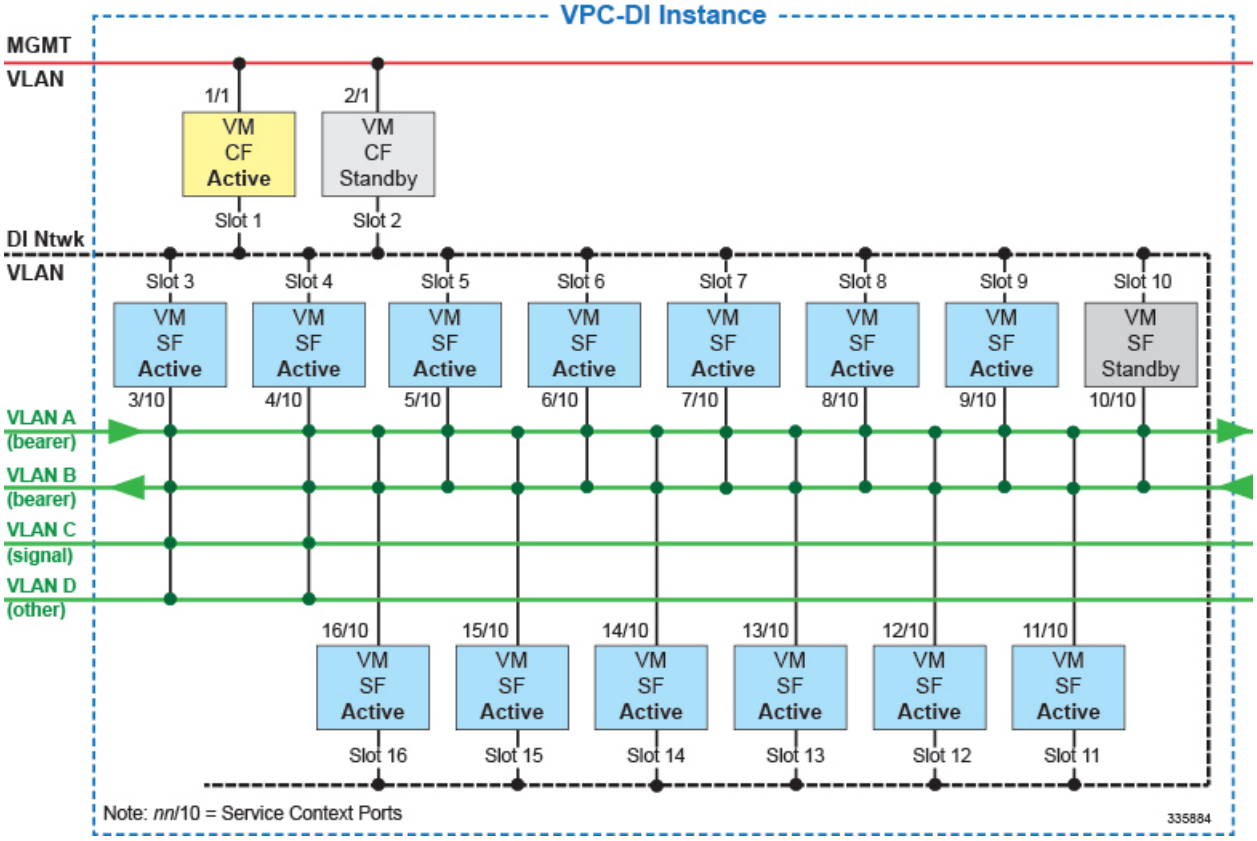
In order for the VMs within a VPC-DI instance to communicate with each other, each instance must have a private L2 network that interconnects the VMs. This network should utilize a VLAN within the IaaS/virtualization infrastructure and be exposed untagged to each VM as the first vNIC.

The DI network must be for the exclusive use of a single VPC-DI instance. No other devices may be connected to this network. It has to be a secure network that limits access.

Note If more than one instance is instantiated within the same datacenter, each instance must have its own DI network.

All the VMs within an instance must be physically located in the same site, ideally in the same few racks with minimal interconnecting devices. The reliability of the DI network is important for the stability of the VPC-DI instance. Using L2 tunneling protocols across a WAN or congested links is highly discouraged.

Figure 4: DI Network



Network Requirements

The reliability and performance of the DI network is critical to the reliability and performance of VPC-DI. The DI Network is used for internal control, signaling, and bearer traffic. Bearer traffic may traverse the DI network multiple times, so any packet loss in the DI network would impact the perceivable packet loss of VPC-DI as a whole.

For OpenStack deployments, both the DI-Network and Service network types should be configured as SR-IOV mode flat. For deployments on VMware, the DI-Network type should be configured as VMXNET3 or PCI-PT with bonding and VLAN tagged. On VMware, the Service network type should be configured as VMXNET3 or PCI-PT.

Note The infrastructure connecting the VMs should be 10 Gbps or higher between all VMs and have a redundant configuration. A redundant configuration can be provided in one of these ways:

- on the host using a vSwitch (for Virtio/VMXNET3 interfaces)
- on the hardware, such as the Cisco UCS virtual interface card (VIC)
- in the VPC-DI using network interface bonding

The IaaS/hypervisor must provide a DI network that can:

- Perform as a L2 Ethernet bridge/switch.
- Support jumbo frames up to at least 7100 bytes. If your installation does not support jumbo frames, you can still use the VPC-DI. You must set the appropriate parameter in the boot parameters file as described in Configure Support for Traffic Above Supported MTU, on page 36.

The infrastructure/hypervisor should provide a DI network that can:

- Support 802.1p priorities sent from the VMs with VID=0.
- Honor 802.1p priorities end-to-end between all VMs within the instance.
- Provide redundant L2 paths in all physical infrastructure or 802.1p priorities end-to-end between all system VMs within the instance.
- Provide a secure network that limits access. It has to be a secure network in a sense, that it is only locally available.

Specifically, the DI network should have the following minimum reliability requirements:

- Fully redundant L2 paths
- No outage longer than 1.5 seconds, including STP and LACP outages (if applicable)
- Packet prioritization
- Sufficient network bandwidth to minimize any control or bearer packet loss

Disruptions in the DI network or excessive packet loss may cause false failure detection or unpredictable behavior in the VPC-DI instance.

Each system VM monitors the reachability of the other VMs and the reliability of the DI network on an ongoing basis.

Jumbo Frames

We recommend that the DI network support jumbo frames up to at least 7100 bytes. On startup, each VM issues a series of ping commands on the network to determine that jumbo frames are supported. Support of jumbo frames provides better system performance.

If your installation does not support jumbo frames, you can still use the VPC-DI. You must set the appropriate parameter in the boot parameters file as described in Configure Support for Traffic Above Supported MTU, on page 36.

The CF and SF do not start if an MTU of less than 7100 is detected and the appropriate boot parameter is not set.

Service ports on SFs can also support a maximum MTU up to 9100 bytes in Release 21.4 and higher, or 2048 bytes in older releases if configured appropriately in the StarOS configuration.

Record Storage

Record storage is available on instance-wide storage devices available at **/records**. Both CF VMs are provisioned with a second vHDD (/hd-raid) of suitable size for record storage (minimum of 16GB). The CFs share a RAID configuration to mirror data between their vHDDs. The SFs send data records to the active CF over the DI network for transfer to external ephemeral storage that was created and mounted manually or orchestrated by the VNFM.

Packet Flows

SF ports are used to receive and transmit bearer and signaling packets. To simplify network settings and address usage, only VLANs for high-bandwidth (bearer) packets need to be connected to all SFs. Low-bandwidth interfaces (signaling) can be connected to just two SFs. In the diagrams below, the bearer VLANs are connected to all SFs, while signaling and other VLANs are only connected to the first two SFs.

Note This asymmetric arrangement means that fewer interfaces are needed, however careful consideration should be paid to failures since the loss of two VMs results in loss of services.

ECMP does load-sharing based on a hash and can send traffic to any SF VM.

On ingress, the SFs perform flow lookups and direct packets to the specific SESSMGR task on a specific SF. Some of this ingress traffic is processed by local SESSMGR tasks, otherwise it is relayed via the DI network to the correct SF. On egress, each SF sends out packets from its local port (provided ECMP is used). In most cases, the number of VMs that packets traverse is less than two. However, ACLs and tunneling may increase the number of hops for specific flows depending on the EPC configuration.

Packets Received on SF Demux VM

On the demux and standby SF, all session traffic received is relayed to another SF for session processing. The figure below shows how ingress packets are distributed via the Demux SF to other session SFs for processing.

Item	Description
1	Receive NPU does flow lookup to determine SESSMGR, (Ingress)
2	SESSMGR processes packet.
3	Transmit NPU sends packet out local port. (Egress)

Packets Received on SF Session VM

The figure below shows how ingress packets received by a session SF are distributed to other session SFs for processing.

Item	Description
1	Receive NPU does flow lookup to determine SESSMGR, (Ingress)
2	SESSMGR processes packet.
3	Transmit NPU sends packet out local port. (Egress)

Feature Set

Interfaces and Addressing

Each VM in a VPC-DI instance is represented as a virtual card with a single CPU subsystem. This makes many CLI commands, logs, and functions work similarly to StarOS running on ASR 5500 platform.

StarOS concepts of contexts, services, pools, interfaces, cards, and ports exist on each VM just as on ASR 5500 platform.

When the VM boots, the vNICs configured in the VM profile are detected and an equivalent number of "Virtual Ethernet" type ports appear in the StarOS CLI.

Refer to Creating a Boot Parameters File, on page 21 to manually specify the order of the vNICs.

By default, the system assigns the vNIC interfaces in the order offered by the hypervisor.

- CF VMs (slots 1 and 2)
  - First interface offered (1/0 or 2/0) is for the DI Network.
  - Second interface offered (1/1 or 2/1) is for the management network.
- SF VMs (Slots 3 through 16)
  - First interface offered (*slot*/0) is for the DI Network.
  - Traffic Interfaces *slot*/10 through *slot*/21 are for IaaS VLAN control and data traffic.

Note StarOS supports up to 12 service ports, but the actual number of ports may be limited by the hypervisor.

It is critical to confirm that the interfaces listed in the supported hypervisors line up with the KVM bridge group or VMware vSwitch in the order in which you want them to match the VM interfaces.

Note You cannot be guaranteed that the order of the vNICs as listed in the hypervisor CLI/GUI is the same as how the hypervisor offers them to the VM. On initial setup you must use the **show hardware** CLI command to walk through the MAC addresses shown on the hypervisor vNIC configuration and match them up with the MAC addresses learned by the VMs. This confirms that the VM interfaces are connected to the intended bridge group or VMware vSwitch.

Encryption

VMs within a VPC-DI instance perform software-based encryption and tunneling of packets (as opposed to the higher-throughput hardware-based services). Call models that make heavy use of encryption for bearer packets or have significant PKI (Public Key Infrastructure) key generation rates may require significant compute resources.

If your COTS server hardware uses the Coleto Creek chipset based on the Intel 89xx chip, the system automatically utilizes this hardware chip for encryption and decryption of packets. However, all service function VMs must use this chipset in order for the system to use the hardware chipset for encryption and decryption.

Security

Security of external traffic including tunneling, encryption, Access Control Lists (ACLs), context separation, and user authentication function as on existing StarOS platforms. User ports and interfaces on the CFs and SFs are protected through StarOS CLI configuration.

The virtual system adds additional security concerns on the customer because network communication travel over the DI network on datacenter equipment.

The DI network must be isolated from other hosts within the datacenter by limiting membership in the system network's VLAN to VMs within that specific VPC-DI instance. Unauthorized access to the DI network through other hosts being inadvertently added to that network or the compromise of a router, switch or hypervisor could disrupt or circumvent the security measures of StarOS. Such disruptions can result in failures, loss of service, and/or exposure of control and bearer packets. Properly securing access to the DI network is beyond the control of StarOS.

Communication between DI network component (e.g. CF and SF) VMs is now only possibley via authentication over externally supplied SSH keys. In addition, the system enforces public/private key-based SSH authentication for logins within the DI network. No passwords, keys or LI information are stored or sent in clear text.

If an operator requires physical separation of networks, such as management versus bearer versus LI (Lawful Intercept), then physical separation of the DI network should also be done since it carries sensitive data. In a virtualized environment, the physical separation of networks may not be possible or practical. Operators that have these requirements may need to qualify their hypervisor and infrastructure to confirm that it will provide sufficient protection for their needs.

Redundancy and Availability

Platform Requirements

The virtual system relies on the underlying hardware and hypervisor for overall resource redundancy and availability. The StarOS handles the system redundancy on top of that.

The hardware and hypervisor should provide:

- Redundant hardware components where practical (such as power supplies, disks)
- Redundant network paths (dual fabric/NICs, with automatic failover)
- Redundant network uplinks (switches, routers, etc.)

High availability can only be achieved if the underlying infrastructure (hosts, hypervisor, and network) can provide availability and reliability that exceeds expected values. The system is only as reliable as the environment on which it runs.

Interchassis Session Recovery (ICSR) is also recommended to improve availability and recovery time in the case of a non-redundant hardware failure (such as CPU, memory, motherboard, hypervisor software). ICSR provides redundancy at the session level for gateways only.

CF Redundancy

The two CF VMs are 1:1 redundant for control of the VPC-DI instance and the local context/management port.

The management port vNIC on both CFs are 1:1 redundant for each other and must be placed in the same VLAN in the infrastructure. Only one management port is active at a time.

Note The two CF VMs must not run on the same physical host (server or blade) to achieve redundancy in case of the failure of the host or hypervisor.

SF Redundancy

Each SF VM provides network connectivity for service ports. Each SF provides one or more ports and associated interfaces, but the SFs do not provide 1:1 redundancy as they are not paired together.

Redundancy of SF ports should be achieved using ECMP or another supported L3 protocol.

The total throughput required of the instance should not exceed N-2 SFs with session recovery enabled so that any single SF can fail while the others take over its load. Use of loopback interfaces for service IP addresses is highly recommended.

It is recommended to use BFD for detection of path failures between an SF and the peer router so ECMP paths are excluded in the event of a failure.

When session recovery is enabled, one VM becomes the VPN/Demux and the remainder are session processing VMs. A standby SF can provide redundancy for any other SF.

Note Each SF VM must run on a different physical host to achieve redundancy in case of the failure of the host or hypervisor.

Iftask Redundancy

If Active standby task gets killed or stopped the transition to Standby Iftask takes place. This transition process involves initialization of certain parameters on the standby Iftask to active Iftask. At the current heartbeat interval, the initialization might take more than three seconds and leads to card reboot.

To avoid the card reboot, set the HAT HB interval to high value.

```
For example
configure
   [deafult] high-availability fault-detection card hb-loss  value
```

ICSR Support

VPC-DI supports ICSR between two instances for services that support ICSR in the StarOS software release. When more than one service type is in use, only those services that support ICSR will be able to use ICSR.

Note The two VPC-DI instances must be on a different DI-Network.

ICSR supports redundancy for site/row/rack/host outages, and major software faults. To do so, the two instances should be run on non-overlapping hosts and network interconnects. ICSR is supported only between like-configured instances. ICSR between a VPC-DI instance and another type of platform is not supported.

L3 ICSR is supported.

For additional information, refer to the *Interchassis Session Recovery* chapter in this guide.

Hypervisor Requirements

VPC-DI has been qualified to run under the following hypervisors:

- OpenStack based virtualized environments
- VMware ESXi
  - Version 6.0: Supported in releases prior to Release 21.8
  - Version 6.5: Supported in Release 21.8 and 21.9
  - Version 6.7: Supported from Release 21.10 onwards

Heat templates (for OpenStack) and OVF/OVA templates (for VMware vSphere) are provided for CF and SF VMs.

A VMware vApp bundles an entire VPC-DI instances's VMs together for easy deployment.

It is recommended to use Cisco Elastic Services Controller (ESC) to deploy VPC-DI with OpenStack. Refer toOnboarding the VPC-DI with ESC on OpenStack, on page 37 .

Note Deviations from the supplied templates must be approved by Cisco engineering in order to maintain performance and reliability expectations.

CF VM Configuration

The system requires that each CF VM be configured with:

- 8 vCPUs

- 16 GB RAM
- First vNIC is the DI Network.
- Second vNIC is the management port.
- First vHDD is for boot image and config storage (/flash, non-RAID, 4GB recommended).
- Second vHDD is for record storage [optional] (hd-local1, RAID, 16GB minimum recommended).

Note Both CF VMs must be identically configured.

SF VM Configuration

The system requires that each SF VM be configured with:

- 12 or more vCPUs (see vCPU and vRAM Options, on page 13).

  Note An SF VM will not boot and will report the following error if this minimum vCPU requirements is not met.

  ```
  Found hardware disparate in minimum number of cores, found n cores,
   minimum expected cores are 12.
  ```

- 32 GB or more vRAM (see vCPU and vRAM Options, on page 13).
- First vNIC is the DI Network.
- Second and subsequent vNICs are service ports. The system supports up to 12 vNICs, but this number may be limited by the hypervisor.
- vHDD is for boot image, 2 GB recommended.

Note All SF VMs must be identically configured. Refer to VM Hardware Verification, on page 94 for information on monitoring the VM hardware configuration.

vCPU and vRAM Options

A CPU is a single physical computer chip that can have more than one physical CPU core that is fully capable of running the entire system and applications by itself. Virtual core technology supports multiple logical processors (vCPUs) per physical core. The total number of vCPUs supported on a specific platform varies based on the number of available physical cores and the type of virtual core technology implemented in each core.

It is best practice to avoid spanning VM across NUMA nodes. The vCPU shall align with the underlying CPU. A VM should not span across NUMA node for performance optimization.

CF and SF run within VMs that are assigned a number of vCPUs, each supporting one thread (sequence of instructions). The number of available vCPUs supported by the platform CPU may exceed the maximum number of vCPUs that can be assigned to the VM via the hypervisor.

Note The number vCPUs per VM should never exceed the maximum number of vCPUs supported by the platform CPU.

To maximize performance, it may be desirable to adjust the number of vCPUs or vRAM to align with the underlying hardware. SF supports varied vCPU and vRAM combinations, however all SFs must share the same combination within an instance.

Software will determine the optimal number of SESSMGR tasks per SF on startup of the SF based on the number of vCPUs and amount of vRAM on that SF.

Note Dynamic resizing of vCPU count, vRAM size or vNIC type/count (via hotplug, ballooning, etc.) is not supported. If these values need to be changed after provisioning, all VMs must be shut down and reconfigured. Reconfiguration can be performed only on all VMs at once. VMs cannot be reconfigured one at a time since the CPUs and RAM would not match the other instances.

vNIC Options

In this release the supported vNIC options include:

- VMXNET3—Paravirtual NIC for VMware
- VIRTIO—Paravirtual NIC for KVM
- ixgbe—Intel 10 Gigabit NIC virtual function
- enic—Cisco UCS NIC
- SR-IOV—Single root I/O virtualization ixgbe and enic interfaces

Support for vhost-net and vhost-user

The system implements a Virtio front-end based on a DPDK-based user application which can interact with both vhost-net and vhost-user based back-end mechanisms. Vhost-user and vhost-net provide an implementation of the vhost paradigm of using shared memory based, event, and interrupt descriptors. The DPDK-based front-end driver in combination with vhost-net and vhost-user provide a higher performance data path compared to a Linux bridge-based data path.

- Vhost-user provides packet handling completely in the user space, which improves performance. The system implements the front-end in a DPDK-based user space application, while a host user space application implements the back-end based on the vhost-user interface.
- Vhost-net provides the kernel-level back-end for Virtio networking that reduces virtualization overhead by moving Virtio packet processing tasks out of the user space (the QEMU process) and into the kernel (the vhost-net driver). This allows device emulation code to directly call into kernel subsystems, instead of performing system calls from the user space.

The system supports single queue in vhost-user.

Hard Drive Storage

In additional to the mandatory /flash (non-RAID) drive, the system supports RAID1 under a virtual machine (VM). For each VM, Virtual SCSI disks can be created, on CF only, matching the SCSI ID shown in this table. The minimum disk size must be greater than 16 GB.

Table 1: Disk Mapping

Type	/flash (non-RAID)	hd-local1	Notes
KVM	SCSI 0:0:0:0	SCSI 0:0:1:0	Raw disk hd-local1 uses RAID1
VMware	SCSI 0:0:0:0	SCSI 0:0:1:0	Raw disk hd-local1 and hd-remote1 use RAID1

For record storage (CDRs and UDRs) the CF VM should be provisioned with a second vHDD sized to meet anticipated record requirements (minimum 16GB). Records will be written to /records on the second vHDD.

DPDK Internal Forwarder

The Intel Data Plane Development Kit (DPDK) is an integral part of the VPC architecture and is used to enhance system performance. The DPDK Internal Forwarder (IFTASK) is a software component that is responsible for packet input and output operations and provides a fast path for packet processing in the user space by bypassing the Linux kernel. It is required for system operation. Upon CF or SF instantiation, DPDK allocates a certain proportion of the CPU cores to IFTASK depending on the total number of CPU cores. The remaining CPU cores are allocated to applications.

To determine which CPU cores are used by IFTASK and view their utilization, use the **show npu utilization table** command as shown here:

```
[local]mySystem# show npu utilization table

Wednesday July 06 10:53:55 PDT 2017
          -------iftask-------
   lcore     now   5min  15min
--------  ------ ------ ------
  01/0/1     38%    53%    52%
  01/0/2     51%    55%    55%
  02/0/1     66%    72%    68%
  02/0/2     66%    63%    67%
  03/0/1     57%    55%    55%
  03/0/2     51%    47%    45%
  03/0/3     89%    89%    89%
  03/0/4     88%    88%    89%
  04/0/1     67%    59%    58%
  04/0/2     54%    40%    48%
  04/0/3     89%    89%    90%
  04/0/4     90%    89%    89%
  05/0/1     55%    55%    56%
  05/0/2     68%    45%    45%
```

```
  05/0/3     90%    89%    89%
  05/0/4     90%    89%    89%
  06/0/1     50%    58%    58%
  06/0/2     24%    24%    25%
  06/0/3     89%    90%    90%
  06/0/4     91%    90%    90%
```

To view CPU utilization for the VM without the IFTASK cores, use the **show cpu info** command. For more detailed information use the **verbose** keyword.

```
[local]mySystem# show cpu info card 6
Tuesday July 05 10:39:52 PDT 2017
Card 6, CPU 0:
  Status                : Active, Kernel Running, Tasks Running
  Load Average          : 7.74, 7.62, 7.54 (9.44 max)
  Total Memory          : 49152M
  Kernel Uptime         : 4D 5H 7M
  Last Reading:
    CPU Usage           : 25.4% user, 7.8% sys, 0.0% io, 0.1% irq, 66.7% idle
    Poll CPUs           : 4 (1, 2, 3, 4)
    Processes / Tasks   : 177 processes / 35 tasks
    Network            : 164.717 kpps rx, 1025.315 mbps rx, 164.541 kpps tx, 1002.149 mbps
 tx
    File Usage          : 8256 open files, 4941592 available
    Memory Usage        : 21116M 43.0% used
  Maximum/Minimum:
    CPU Usage           : 32.9% user, 8.9% sys, 0.0% io, 0.4% irq, 59.1% idle
    Poll CPUs           : 4 (1, 2, 3, 4)
    Processes / Tasks   : 184 processes / 36 tasks
    Network            : 178.388 kpps rx, 1270.977 mbps rx, 178.736 kpps tx, 1168.999 mbps
 tx
    File Usage          : 8576 open files, 4941272 available
    Memory Usage        : 21190M 43.1% used
```

Orchestration

When a VPC-DI instance is deployed, there are several expectations of the environment on which VPC-DI is running that are beyond the control of StarOS. Most of these fall into requirement of the Orchestration System.

- Provisioning of VPC-DI VMs including install and parameters assignment: configuration, connectivity, and persistent block storage for each VM.
- L2 provisioning of the DI network to ensure that the DI network meets reliability requirements.
- Policy enforcement of network separation, if applicable.
- Physical placement of VMs that enforce redundancy rules.
- Providing useful monitoring tools for physical resources, such as CPU, RAM, NIC, etc.

If an orchestration system is not used to deploy a VPC-DI instance, these requirements must still be maintained. However, they should be enforced manually or through other means. Refer to VM Hardware Verification, on page 94 for information on monitoring the VM hardware configuration.

Provisioning

Provisioning of a VPC-DI instance has two phases:

- VMs and network interconnections are created and linked.
- VPC-DI instance is configured for services.

IaaS administrators set up and interconnect the servers and use hypervisor VM templates or orchestration software to create a set of VMs, the DI network, and the redundancy configuration to meet Service Level Agreement (SLA) requirements.

Deploying a VPC-DI instance requires a detailed configuration plan that addresses the operator's deployment requirements.

Boot Sequence

StarOS is installed on each VM using pre-installed disk templates in QCOW2 format. Slot numbers are managed by ESC and OpenStack. A slot number is assigned as part of the VM configuration. The slot number is auto-detected during install. Installation of the installer image is completely automated provided that the slot number can be detected from the hypervisor. For additional information, see Software Installation and Network Deployment, on page 20.

For information regarding how to control the configuration of vNICs from the VM, refer to Creating a Boot Parameters File, on page 21.

Each VM will reboot and attempt to join the VPC-DI Instance. A bootloader boots the instance via automated (scripted), network or manual booting.

Upon completion of the virtual BIOS, the VM boots from its local vHDD and runs CFE (Common Firmware Environment). CFE looks on the vHDD for the presence of the parameters file that was created during installation. If this file is found and parses correctly, CFE takes different paths depending on the VM's type and slot number. In all cases, the first vNIC becomes the interface for the network.

CF Boot Sequence

The CF performs the following functions during its boot sequence:

- Checks to see if the other CF is alive (via the DI network).
- If other CF is alive, attempts to boot from it.
  - Tries to obtain parameters and a boot image from the other CF.
  - If successful, transfers the boot image and runs it.
- If the other CF is not alive or booting from it fails, boots independently.
  - Finds and parses a boot.sys file on the local vHDD for boot/config priorities.
  - Performs a boot via instructions in the boot.sys unless interrupted by a user (via the Management network or local vHDD).

CFE on a CF supports downloading a starfile (bootable image) from the peer CF, via the CF management vNIC to an external HTTP or TFTP server, or from a local file on its vHDD. This is driven by the boot.sys and the StarOS **boot** CLI command.

Note HTTP and TFTP booting are only supported on VIRTIO and VMXNET3 interface types.

A network protocol on the DI network determines which CF is master. Mastership is then communicated over the DI network to SF VMs.

SF Boot Sequence

An SF boots from its vHDD. It then contacts the active CF via the DI network to determine if it booted the correct software version. If the SF did not boot the correct software version, it transfers the correct version from the CF and reboots itself. Once it boots the correct the software version, the boot sequence is complete.

Bandwidth Requirements

Modeling of bandwidth requirements on the L2 switches that host a VPC-DI instance is required for each operator deployment.

In addition to the predominant bearer traffic, the DI network also passes session signaling and internal control data between the VMs.

Internal control traffic will be heavy during redundancy operations, but significantly less under normal operation. Heavy use of control traffic occurs during:

- Migrations of tasks from an active SF VM to the standby SF
- Startup or restart of a standby SF
- Startup or restart of an SF
- Startup or restart of an SF or standby CF
- Heavy signaling traffic (high Call Events per Second [CEP] rate)
- Significant CLI and/or Bulkstats usage

Depending on the CEPS rate, configuration, and management operations, each VM places a load on its DI network interface regardless of bearer throughput. This load is expected to be highly variable, but average less than 1 Gbps per VM with some VMs having higher usage than others.

Capacity, CEPS and Throughput

Sizing a VPC-DI instance requires modeling of the expected call model.

Initial software versions support up to 2 CFs and 14 SFs.

Many service types require more resources than others. Packet size, throughput per session, CEPS (Call Events per Second) rate, IPSec usage (site-to-site, subscriber, LI), contention with other VMs, and the underlying hardware type (CPU speed, number of vCPUs) will further limit the effective number of maximum subscribers. Qualification of a call model on equivalent hardware and hypervisor configuration is required.

Software-based transmit batching greatly enhances the system performance.

Diagnostics and Monitoring

Because VPC-DI runs within VMs, no hardware diagnostics or monitoring are provided. Retrieval of hardware sensor data (temperature, voltage, memory errors) are accomplished via the hypervisor and external monitoring systems. To determine the configuration of the underlying VMs, refer to VM Hardware Verification, on page 94.

VPC-DI monitors and exports vCPU, vRAM, and vNIC usage per VM through existing mechanisms including CLI **show** commands, bulkstats and MIB traps. However, an operator may find that monitoring physical CPU, RAM, and NIC values per host in the hypervisor is more useful.

Because vNICs have a variable max throughput (not defined as 1 Gbps or 10 Gbps for example), counters and bulkstats that export utilization as a percentage of throughput may have little value. Absolute values (bps) can be obtained from the VM, but where possible physical infrastructure utilization should be obtained from the hypervisor. This would not apply to pass-through PF NICs, as those have a fixed maximum throughput.

Cisco Prime Analytics

The Cisco Prime for Mobility suite of analytics provides scalable management of a VPC-DI instance.

Cisco Prime for Mobility supports the following:

- Integrated operator workflows across the Radio Access Network (RAN) backhaul and packet core
- Centralized network visibility and advanced troubleshooting and diagnostics
- Pre-integrated network management software components that reduce time and resources required for integration

For additional information, contact your Cisco account representative.

StarOS VPC-DI Build Components

The following StarOS build filename types are associated with VPC-DI:

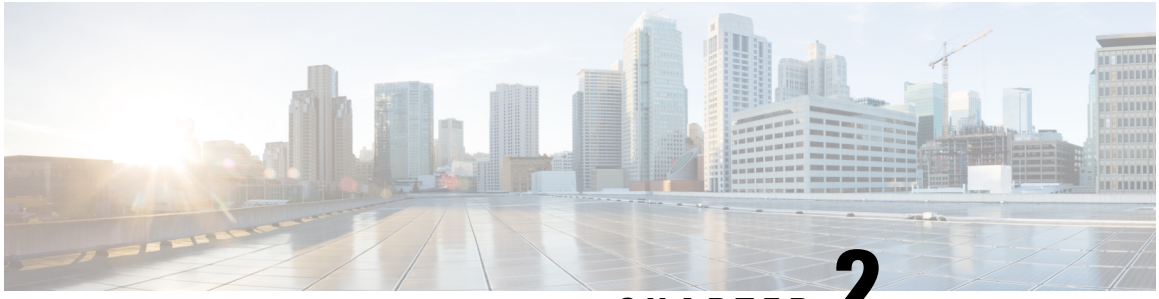
- **.qvpc-di-<version>.iso** initial installation or startover ISO file.
- **.qvpc-di-<version>.bin** update, upgrade or recovery file for a system that is already running.
- **.qvpc-di-template-libvirt-kvm-<version>.tgz** KVM libvirt template plus ssi_install.sh.
- **.qvpc-di.qcow2.tgz** KVM QCOW2 disk template.
- **.qvpc-di-template-vmware.tgz** VMware files.
- **.qvpc-di-template-vmware-<version>.ova** VMware OVA template.

Software Installation and Network Deployment

This guide assumes that components of VPC-DI have been properly installed to run in virtual machines (VMs) on commercial off-the shelf (COTS) servers. For additional information, see Provisioning, on page 17.

The DI network must also be provisioned within the datacenter to meet the requirements specified in DI Network, on page 5 and Bandwidth Requirements, on page 18.

For additional information on supported operating system and hypervisor packages, as well as platform configurations, please contact your Cisco representative. The Cisco Advanced Services (AS) group offer consultation, installation and network deployment services for the VPC-DI product.



CHAPTER 2

VPC-DI Installation Notes

This guide assumes that components of VPC-DI have been properly installed to run in virtual machines (VMs) on commercial off-the shelf (COTS) servers. This chapter provides some installation notes that may assist in the installation process.

Creating a Boot Parameters File

The boot parameters file provides a means to pass configuration items to StarOS before it boots. The parameters are typically necessary to successfully load StarOS and specify items such as virtual slot number, VM type, NIC assignment and network bonding configuration.

By default, VPC-DI assigns the vNIC interfaces in the order offered by the hypervisor. To configure your vNICs manually according to a specific order, you need to create a boot parameters file. You also must create a boot parameters file if you want to enable a VNFM interface.

The boot parameters are sourced in multiple ways, with all methods using the same parameter names and usage. The first location for the boot parameters file is on the first partition of the first VM drive, for example, */boot1/param.cfg*. The second location searched is on the configuration drive, which is a virtual CD-ROM drive. If you are using OpenStack, specify the target boot parameters file name as *staros_param.cfg*. If you are not using OpenStack, create an ISO image with *staros_param.cfg* in the root directory and attach this ISO to the first virtual CD-ROM drive of the VM.

As the VM boots, the param.cfg file is parsed first by the preboot environment known as CFE. Once the VM starts Linux, the virtual CD-ROM drive is accessed to parse the *staros_param.cfg* file. If there are any conflicts with values stored in the */boot1/param.cfg* file, parameters in staros_param.cfg take precedence.

If you do not create a boot parameters file, the default file is used. If you create a boot parameters file, all parameters described in Configuring Boot Parameters, on page 27 must be defined.

Format of the Boot Parameters File

The structure of the boot parameters file is:

VARIABLE_NAME = VALUE

Specify one variable per line with a newline as the end of the line terminator (UNIX text file format). Variable names and values are case insensitive. Invalid values are ignored and an error indication is displayed on the VM console. If there are duplicate values for a variable (two different values specified for the same variable name), the last value defined is used.

Numeric values do not need to be zero padded. For example a PCI_ID of 0:1:1.0 is treated the same as 0000:01:01.0.

Network Interface Roles

Network interfaces serve specific roles depending on whether the VM is used for a CF or SF.

All system VMs have a network interface connection to the DI internal network. This network links all the VMs in a VPC-DI instance together. This network must be private to a VPC-DI instance and is configured by the system software.

All VMs have the option of configuring a network interface that is connected to the virtual network function (VNF) manager (VNFM) if it exists. This interface can be configured via DHCP or static IP assignment and is used to talk to a VNFM or higher level orchestrator. This interface is enabled before the main application starts.

On CFs, one additional interface connects to the management network interface. This interface is typically configured in StarOS and should be part of the Day 0 configuration. The management interface supports static address assignment through the main StarOS configuration file.

On SFs, an additional 0 to 12 network interfaces serve as service ports. These interfaces are configured by StarOS. Typically these ports are configured as trunk ports in the VNF infrastructure (VNFI).

Table 2: Network Interface Roles

Interface Role	Description
DI_INTERFACE	Interface to the DI internal network, required for all VM types
MGMT_INTERFACE	Interface to the management port on the CF VM
SERVICE#_INTERFACE	Service port number # on the SF VM, where # can be from 1 to 12.
VNFM_INTERFACE	Optional network interface to the VNFM or orchestrator, valid for all VM types

Note Although VIRTIO interfaces can be used for the DI_INTERFACE role and the SERVICE#_INTERFACE roles, they are not recommended.

Network Interface Identification

By default the first NIC found by a VPC-DI VM is assigned the DI internal network role. Additional ports serve as either the management interface on the CF or service ports on the SF. No interface is used as the VNFM interface by default.

VPC-DI assigns the vNIC interfaces in the order offered by the hypervisor. You cannot be guaranteed that the order of the vNICs as listed in the hypervisor CLI/GUI is the same as how the hypervisor offers them to the VM.

The order that VPC-DI finds the vNICs is subject to the PCI bus enumeration order and even paravirtual devices are represented on the PCI bus. The PCI bus is enumerated in a depth first manner where bridges are explored before additional devices at the same level. If all the network interfaces are of the same type then knowing the PCI topology is sufficient to get the vNIC order correct. If the network interfaces are of different types, then the order is dependent on the PCI topology plus the device driver load order inside the VM. The device driver load order is not guaranteed to be the same from software release to release but in general paravirtual devices are prior to pass-through devices.

There are several methods available to identify NICs.

- MAC address: MAC address of the interface
- Virtual PCI ID
- Bonded interfaces: When using network device bonding, network interfaces are identified to serve as the slave interface role. The slave interfaces in the bond are identified using MAC, PCI ID, or Interface type.
- Interface type and instance number.

Virtual PCI ID

Devices on a PCI bus are identified by a unique tuple known as the domain, bus, device, and function numbers. These identifiers can be identified in several ways.

Inside the guest, the **lspci** utility shows the bus configuration:

```
# lspci

00:00.0 Host bridge: Intel Corporation 440FX - 82441FX PMC [Natoma] (rev 02)
00:01.0 ISA bridge: Intel Corporation 82371SB PIIX3 ISA [Natoma/Triton II]
00:01.1 IDE interface: Intel Corporation 82371SB PIIX3 IDE [Natoma/Triton II]
00:01.2 USB controller: Intel Corporation 82371SB PIIX3 USB [Natoma/Triton II] (rev 01)
00:01.3 Bridge: Intel Corporation 82371AB/EB/MB PIIX4 ACPI (rev 03)
00:02.0 VGA compatible controller: Cirrus Logic GD 5446
00:03.0 System peripheral: Intel Corporation 6300ESB Watchdog Timer
00:04.0 Unclassified device [00ff]: Red Hat, Inc Virtio memory balloon
00:05.0 Ethernet controller: Red Hat, Inc Virtio network device
00:06.0 Ethernet controller: Red Hat, Inc Virtio network device
```

The domain, bus, device, and function numbers for this virtual bus are shown here:

Table 3: Virtual PCI IDs

Line	Domain	Bus	Device	Function
00:00.0 Host bridge: Intel Corporation 440FX - 82441FX PMC [Natoma] (rev 02)	0	0	0	0
00:01.0 ISA bridge: Intel Corporation 82371SB PIIX3 ISA [Natoma/Triton II]	0	0	1	0

Line	Domain	Bus	Device	Function
00:01.1 IDE interface: Intel Corporation 82371SB PIIX3 IDE [Natoma/Triton II]	0	0	1	1
00:01.2 USB controller: Intel Corporation 82371SB PIIX3 USB [Natoma/Triton II] (rev 01)	0	0	1	2
00:01.3 Bridge: Intel Corporation 82371AB/EB/MB PIIX4 ACPI (rev 03)	0	0	1	3
00:02.0 VGA compatible controller: Cirrus Logic GD 5446	0	0	2	0
00:03.0 System peripheral: Intel Corporation 6300ESB Watchdog Timer	0	0	3	0
00:04.0 Unclassified device [00ff]: Red Hat, Inc Virtio memory balloon	0	0	4	0
00:05.0 Ethernet controller: Red Hat, Inc Virtio network device	0	0	5	0
00:06.0 Ethernet controller: Red Hat, Inc Virtio network device	0	0	6	0

For libvirt-based virtual machines, you can get the virtual PCI bus topology from the **virsh dumpxml** command. Note that the libvirt schema uses the term *slot* for the device number. This is a snippet of the xml description of the virtual machine used in the previous example:

```
<interface type='bridge'>
  <mac address='52:54:00:c2:d0:5f'/>
  <source bridge='br3043'/>
  <target dev='vnet0'/>
  <model type='virtio'/>
  <driver name='vhost' queues='8'/>
  <alias name='net0'/>
  <address type='pci' domain='0x0000' bus='0x00' slot='0x05' function='0x0'/>
</interface>
<interface type='bridge'>
  <mac address='52:54:00:c3:60:eb'/>
  <source bridge='br0'/>
  <target dev='vnet1'/>
  <model type='virtio'/>
  <alias name='net1'/>
  <address type='pci' domain='0x0000' bus='0x00' slot='0x06' function='0x0'/>
</interface>
```

Interface Type and Instance Number

Here the NIC is identified by its type using its Linux device driver name (virtio_net, vmxnet3, ixgbe, i40e, etc) and its instance number. The instance number is based on PCI enumeration order for that type of interface starting at instance number 1. The interface type is available to identify both paravirtual types as well as pass-through interfaces and SR-IOV virtual functions. The PCI enumeration order of devices on the PCI bus can be seen from the **lspci** utility, which is on the host OS.

For example, a CF with the following guest PCI topology indicates that virtio_net interface number1 is the Ethernet controller at 00:05.0 and virtio_net interface number 2 is the Ethernet Controller at 00:06.0. The output is from the **lspci** command executed in the guest:

```
# lspci

00:00.0 Host bridge: Intel Corporation 440FX - 82441FX PMC [Natoma] (rev 02)
00:01.0 ISA bridge: Intel Corporation 82371SB PIIX3 ISA [Natoma/Triton II]
00:01.1 IDE interface: Intel Corporation 82371SB PIIX3 IDE [Natoma/Triton II]
00:01.2 USB controller: Intel Corporation 82371SB PIIX3 USB [Natoma/Triton II] (rev 01)
00:01.3 Bridge: Intel Corporation 82371AB/EB/MB PIIX4 ACPI (rev 03)
00:02.0 VGA compatible controller: Cirrus Logic GD 5446
00:03.0 System peripheral: Intel Corporation 6300ESB Watchdog Timer
00:04.0 Unclassified device [00ff]: Red Hat, Inc Virtio memory balloon
00:05.0 Ethernet controller: Red Hat, Inc Virtio network device
00:06.0 Ethernet controller: Red Hat, Inc Virtio network device
```

Here is the complete list of the supported Linux drivers:

Table 4: Supported Linux Drivers

Type	PCI Vendor / Device ID	Driver Name
VIRTIO (paravirtual NIC for KVM)	0x10af / 0x1000	virtio_net
VMXNET3 (paravirtual NIC for VMware)	0x15ad / 0x07b0	vmxnet3

Type	PCI Vendor / Device ID	Driver Name
Intel 10 Gigabit Ethernet	0x8086 / 0x10b6 0x8086 / 0x10c6 0x8086 / 0x10c7 0x8086 / 0x10c8 0x8086 / 0x150b 0x8086 / 0x10dd 0x8086 / 0x10ec 0x8086 / 0x10f1 0x8086 / 0x10e1 0x8086 / 0x10db 0x8086 / 0x1508 0x8086 / 0x10f7 0x8086 / 0x10fc 0x8086 / 0x1517 0x8086 / 0x10fb 0x8086 / 0x1507 0x8086 / 0x1514 0x8086 / 0x10f9 0x8086 / 0x152a 0x8086 / 0x1529 0x8086 / 0x151c 0x8086 / 0x10f8 0x8086 / 0x1528 0x8086 / 0x154d 0x8086 / 0x154f 0x8086 / 0x1557	ixgbe
Intel 10 Gigabit NIC virtual function	0x8086 / 0x10ed 0x8086 / 0x1515	ixgbevf
Cisco UCS NIC	0x1137 / 0x0043 0x1137 / 0x0044 0x1137 / 0x0071	enic

Type	PCI Vendor / Device ID	Driver Name
Mellanox ConnectX-5	0x15b3 / 0x1017 0x15b3 / 0x1018	mlx5_core
Intel XL 710 family NIC (PF)	0x8086 / 0x1572 (40 gig) 0x8086 / 0x1574 (40 gig) 0x8086 / 0x1580 (40 gig) 0x8086 / 0x1581 (40 gig) 0x8086 / 0x1583 (40 gig) 0x8086 / 0x1584 (40 gig) 0x8086 / 0x1585 (40 gig) 0x8086 / 0x158a (25 gig) 0x8086 / 0x158b (25 gig)	i40e**
Intel XL 710 family NIC virtual function	0x8086 / 0x154c	i40evf

** Note: A known issue exists where MAC address assignment does not occur dynamically for SRIOV VFs created on the host when using the **i40e** driver. MAC address assignment is necessary to boot the StarOS VM. As a workaround, MAC address assignment must be configured from the host. Refer to the following link for more information:https://www.intel.com/content/dam/www/public/us/en/documents/technology-briefs/xl710-sr-iov-config-guide-gbe-linux-brief.pdf

Configuring Boot Parameters

If you do not create a boot parameters file, the default file is used. If you create a boot parameters file, all parameters described in this task must be defined.

Before you begin

Refer to Network Interface Roles, on page 22 and Network Interface Identification, on page 22 for more information on determining the interface identifiers for your VM interfaces.

Step 1 **CARDSLOT=***slot-number*

slot_number is an integer between 1 and 32 that indicates the slot number or VM. CF slots can be 1 or 2. SF slots can range from 3 to 48.

Step 2 **CARDTYPE=***card-type*

card-type identifies whether the VM is a CF or SF.

- Use 0x40010100 for Control Function.
- Use 0x42020100 for Service Function.

Step 3 *interface-role***_INTERFACE=***interface-id*

Valid values for *interface-role* are:

- DI
- MGMT
- SERVICE#, where # can range from 1 to 12
- VNFM

For example, DI_INTERFACE=interface-id.

Refer to Network Interface Roles, on page 22 for more information on interface roles.

Valid values for *interface-id* are:

- MAC: xx:xx:xx:xx:xx:xx
- PCI_ID:xxxx:xx:xx.x (Domain:Bus:Device.Function)
- TYPE:*drive-name–instance-number*
- BOND: *slave-interface-A,slave-interface-B*

Refer to Network Interface Identification, on page 22 for information on determining the interface identifier.

Example:

This example identifies the interface by its MAC address:

```
DI_INTERFACE=MAC:00:01:02:03:04:05
```

This example identifies the interface by its guest PCI address:

```
DI_INTERFACE=PCI_ID:0000:01:02.0
```

This example identifies the interface by its interface type (1st virtio interface):

```
DI_INTERFACE=TYPE:enic-1
```

Example:

This example identifies the interfaces as a network bond interface. The example illustrates identifying the interface using MAC address, PCI identifier and interface type:

```
DI_INTERFACE=BOND:MAC:00:01:02:03:04:05,MAC:00:01:02:03:04:06
# or
DI_INTERFACE=BOND:PCI_ID:0000:01:01.0,PCI_ID:0000:01:02.0
# or
DI_INTERFACE=BOND:TYPE:enic-1,TYPE:enic-2
```

Configuring Network Interface Bonding

The system supports configuring pairs of network interfaces into an active/standby bonded interface. Only one interface is active at a time and failure detection is limited to the loss of the physical link. Use this task to configure bonded interfaces.

All bonding variable names use the format *interface-role*_BOND. Refer to Network Interface Roles, on page 22 for information on interface roles.

Before you begin

All boot parameters described in this task are optional. If these parameters are required, add them to the boot parameters file together with the required parameters described in Configuring Boot Parameters, on page 27.

Step 1 *interface-role*_BOND_PRIMARY=*interface-id*

Configures the primary slave interface if you have a preference for a particular interface to be active the majority of the time. The default bond configuration does not select a primary slave.

Refer to Network Interface Roles, on page 22 for information on interface roles; refer to Network Interface Identification, on page 22 for information regarding interface identifiers.

Note By default, the reversion policy is that the bond only reverts back to the primary interface on a subsequent failure of the new active link.

By default, the failure detection method is that the bond uses the driver state to poll link status of the underlying interfaces.

Example:

This example specifies the primary interface using a MAC address:

```
DI_INTERFACE_BOND_PRIMARY=MAC:00:01:02:03:04:05
```

This example specifies the primary interface using a PCI identifier:

```
DI_INTERFACE_BOND_PRIMARY=BOND:PCI_ID:0000:01:01.0
```

This example specifies the primary interface using an interface type identifier:

Example:

```
DI_INTERFACE_BOND_PRIMARY=BOND:TYPE:enic-1
```

Step 2 *interface-role*_BOND_MII_POLL = *poll-interval*

Specifies the poll interval, in milliseconds, to use when MII is used for link detection. The poll interval can range from 0 to 1000. The default is 100.

Step 3 *interface-role*_BOND_MII_UPDELAY=*slave-enable-delay*

Specifies how long to wait for the link to settle before enabling a slave interface after a link failure, when MII is used for link detection. The link state can bounce when it is first detected. This delay allows the link to settle before trying to use the interface and thereby avoids excessive flips in the active slave for the bond interface.

The slave enable delay must be a multiple of the MII poll interval. Values are in milliseconds and the default is 0.

Step 4 *interface-role*_BOND_MII_DOWNDELAY=*slave-disable-delay*

Optional. When used, it allows the bond to wait before declaring that the slave interface is down, when MII is used for link detection. The slave disable delay must be a multiple of the MII poll interval. Values are in milliseconds and the default is 0.

Configuring a VNFM Interface

A virtual network function management (VNFM) interface is designed to communicate between each VM and a VNFM. This interface is brought up before the main application and can be configured only using the boot parameters. The VNFM interface is disabled by default.

Use this task to configure a VNFM interface:

Before you begin

All boot parameters described in this task are optional. If these parameters are required, add them to the boot parameters file together with the required parameters described in Configuring Boot Parameters, on page 27.

Step 1 VNFM_IPV4_ENABLE={true | false}

Enables the VNFM interface.

Step 2 VNFM_CARTRIDGE_AGENT={true | false}

Enables the cartridge agent. This must be enabled if the VNFM is using the cartridge agent.

Step 3 VNFM_IPV4_DHCP_ENABLE={true | false}

Enables DHCP on the VNFM.

Step 4 VNFM_IPV4_ADDRESS=*x.x.x.x*

Specifies the IP address for the VNFM where DHCP is not used.

Step 5 VNFM_IPV4_NETMASK=*x.x.x.x*

Specifies the netmask for the IP address of the VNFM where DHCP is not used.

Step 6 VNFM_IPV4_GATEWAY=*x.x.x.x*

Specifies the gateway for the IP address of the VNFM where DHCP is not used.

VNFM Interface Options

Note These configuration options are optional.

The virtual network functions manager (VNFM) interface is designed to communicate between each VM and a VNFM. The VNFM interface initializes before the main application and only boot parameters can configure the interface.

The VNFM interface is disabled by default.

Enable VNFM IPv4 Interface

The default value is False (disabled).

Variable	Valid Values
VNFM_IPV4_ENABLE	True or False

Configure IPv4 DHCP Client

Variable	Valid Values
VNFM_IPV4_DHCP_ENABLE	True or False

Configure IPv4 Static IP

Note If IPv4 DHCP client is enabled, static configuration parameters are ignored.

Variable	Valid Values
VNFM_IPV4_ADDRESS	x.x.x.x
VNFM_IPV4_NETMASK	x.x.x.x
VNFM_IPV4_GATEWAY	x.x.x.x

Enable VNFM IPv6 Interface.

Variable	Valid Values
VNFM_IPV6_ENABLE	True or False

Enable IPv6 Static IP Configuration

Variable	Valid Values
VNFM_IPV6_STATIC_ENABLE	True or False

If set to true, static IP parameters configuration applies to the interface as shown in the following section. If set to false, the interface attempts to use both stateless autoconfiguration (RFC4862) and DHCPv6 to configure the address of the interface.

Configure IPv6 Static IP

Note If the "VNFM_IPV6_ENABLE" parameter value is set to false, the static configuration parameters are ignored. The IPv6 address field should conform to RFC 5952. Prefix is fixed at /64.

Variable	Valid Values
VNFM_IPV6_ADDRESS	x:x:x:x:x:x:x:x
VNFM_IPV6_GATEWAY	x:x:x:x:x:x:x:x

Configuring the DI Network VLAN

The DI network requires a unique and isolated network available for its use. When using pass-through interfaces, a VLAN ID can be configured to allow for easier separation of the VPC-DI instances in the customer network. Optionally, the DI Network VLAN can also be tagged on the host or even the L2 switch, if there are dedicated ports on the host.

Use this task to configure the VLAN.

Before you begin

All boot parameters described in this task are optional. If these parameters are required, add them to the boot parameters file together with the required parameters described in Configuring Boot Parameters, on page 27.

DI_Internal_VLANID=*vlan-id*

Specifies a VLAN ID for the internal DI network. Values can range from 1 to 4094.

Example:

```
DI_INTERNAL_VLANID=10
```

Configuring IFTASK Tunable Parameters

By default, DPDK allocates 30% of the CPU cores to the Internal Forwarder Task (IFtask) process. You can configure the resources allocated to IFTASK using these boot parameters. Use the **show cpu info** and **show cpu verbose** commands to display information regarding the CPU core allocation for IFTASK.

Note These are optional parameters that should be set with extreme care.

Step 1 (Optional) **IFTASK_CORES=***percentage-of-cores*

Specify the percentage of CPU cores to allocate to IFTASK. Values can range from 0 to 100 percent. The default is 30.

Step 2 (Optional) **IFTASK_MCDMA_CORES=***percentage-of-iftask-cores*

Sets the number of cores assigned to Multi-channel Direct Memory Access (MCDMA) to be a percentage of total iftask cores. You must first define **IFTASK_CORES** parameter above.

Note When NUMA optimization is enabled and also this MCDMA core count setting is configured (**IFTASK_CORES=***percentage-of-cores*), you must set *percentage-of-cores* to be an even number. This ensures that the MCDMA threads are evenly distributed across the NUMA nodes.

Step 3 (Optional) **MCDMA_THREAD_DISABLE=***percentage-of-iftask-cores*

Set the MCDMA_THREAD_DISABLE parameter to 1 to run PMDs on all cores, rather than using an MCDMA - VNPU split.

Step 4 (Optional) **IFTASK_SERVICE_TYPE=***value*

Specifies the service type being deployed in order to calculate the service memory and enable service-specific features. The following service types can be specified:

- 0 = VPC service type
- 1 = GiLAN service type
- 2 = ePDG service type
- 3 = CUPS controller service type
- 4 = CUPS forwarder service type

The default is 0.

Step 5 (Optional) **IFTASK_CRYPTO_CORES=***value*

When **IFTASK_SERVICE_TYPE** is configured to "2" (EPDG), this parameter specifies the percentages of iftask cores to allocate to crypto processing. Values can range from 0 to 50 percent, though the cores dedicate will be capped at 4. The default is 0.

Note This parameter should only be used if the **IFTASK_SERVICE_TYPE** is set to "2" (EPDG). If it is set to any other service type, then this parameter should be set to "0".

Step 6 (Optional) **IFTASK_SW_TX-RSS=***value*

Use this setting to disable the transmit path throughput enhancements on the system.

- 0 = enabled (Default)
- 1 = disabled

Step 7 (Optional) **IFTASK_DISABLE_NUMA_OPT=***value*

Use this setting to disable the NUMA optimizations, even though more than 1 NUMA node is presented to the VM by the host. This option can be set when NUMA optimizations are not desirable for whatever reason.

- NO = enabled (default)
- YES = disabled

NUMA optimization is enabled by default, except for the following cases:

- The number of NUMA nodes/cells does not equal 2.
- Card type is Control Function (CF), Application Function (AF), or Network Function (NF). Only Service Function (SF) VMs support NUMA.
- The service type of the VM is not VPC. NUMA is only supported for VPC service type.
- This setting is explicitly set to YES (**IFTASK_DISABLE_NUMA_OPT=YES**).

Step 8 (Optional) **IFTASK_VNPU_TX_MODE=***value*

The compute nodes in an Ultra M deployment have 28 cores. Two of these cores are reserved for use by the host. When 26 cores are utilized, this results in an unequal distribution of MCDMA channels across the cores used to perform MCDMA work.

When this setting is enabled, the MCDMA function cores in iftask are split equally as MCDMA cores and VNPU TX lookup cores.

- 0 = disabled (Default)
- 1 = enabled

Step 9 (Optional) **MULTI_SEG_MBUF_ENABLE=***value*

By default in release 21.6 and higher, the system enables the use of multi-segmented transmission/reception with smaller size buffers in all memory pools for Ixgbe pf/vf drivers. This feature reduces the overall memory size of IFTASK and makes it more suitable for small deployments.

- 1 = true (default for Ixgbe NICs).
- 0 = false (default for all other NICs).

Important Care must be taken when upgrading to 21.6 on systems which use Ixgbe NICs as this feature by default is enabled.

This feature is automatically disabled for any system not using the Ixgbe vf/pf NICs NICs.

Example

Use the StarOS command **show cloud hardware iftask** *card_number* to verify that the boot parameters took effect:

```
[local]mySystem# show cloud hardware iftask 4
Card 4:
  Total number of cores on VM:      24
  Number of cores for PMD only:     0
  Number of cores for VNPU only:    0
  Number of cores for PMD and VNPU: 3
  Number of cores for MCDMA:        4
  Number of cores for Crypto        0
Hugepage size:                2048 kB
Total hugepages:           3670016 kB
NPUSHM hugepages:                0 kB
CPU flags: avx sse sse2 ssse3 sse4_1 sse4_2
Poll CPU's: 1 2 3 4 5 6 7
KNI reschedule interval: 5 us
```

Increased Maximum IFtask Thread Support

Feature Summary and Revision History

Summary Data

Applicable Product(s) or Functional Area	All
Applicable Platform(s)	VPC-DI
Feature Default	Enabled - Always-on
Related Changes in This Release	Not applicable
Related Documentation	*VPC-DI System Administration Guide*

Revision History

Important Revision history details are not provided for features introduced before releases 21.2 and N5.1.

Revision Details	Release
From this release, the maximum number of IFtask threads configuration supported is increased to 22 cores.	21.8
First introduced.	Pre 21.2

Feature Changes

When the number of DPDK Internal Forwarder (IFTask) threads configured (in /tmp/iftask.cfg) are greater than 14 cores, the IFTask drops packets or displays an error.

Previous Behavior: Currently, the maximum number of IFtask threads configuration is limited to only 14 cores.

New Behavior: From Release 21.8, the maximum number of IFtask threads configuration supported is increased to 22 cores.

Configure MTU Size

By default, the IFTASK process sets the maximum interface MTU as follows:

- Service interfaces: 2100 bytes
- DI network interface: 7100 bytes

These default can be modified by setting the following parameters in the param.cfg file:

Table 5: MTU Size Parameters

Parameter Name	Range	Default Value
DI_INTERFACE_MTU=	576-9100	7100
SERVICE_INTERFACE_MTU=	576-9100	2100

Refer to Configure Support for Traffic Above Supported MTU, on page 36 for configuring the MTU size for a system which does not support jumbo frames.

Configure Support for Traffic Above Supported MTU

By default, jumbo frame support is required for the system to operate. If your infrastructure does not support jumbo frames, you can still run the system, however you must specify the MTU for the DI internal network to be 1500 in the boot parameters file. This allows the IFTASK to process DI network traffic that is above the supported MTU.

Before you begin

All boot parameters described in this task are optional. If these parameters are required, add them to the boot parameters file together with the required parameters described in Configuring Boot Parameters, on page 27.

DI_INTERFACE_MTU=*1500*

Specifies that the DI internal network does not support jumbo frames so that the software handles jumbo frames appropriately.

Boot Parameters File Examples

This example shows a boot parameters file for a CF in slot 1 with two VIRTIO interfaces:

```
CARDSLOT=1
CARDTYPE=0x40010100
DI_INTERFACE=TYPE:enic-1
MGMT_INTERFACE=TYPE:virtio_net-2
```

This example shows a boot parameters file for an SF in slot 3 with three VIRTIO interfaces:

```
CARDSLOT=3
CARDTYPE=0x42020100
DI_INTERFACE=TYPE:enic-1
SERVICE1_INTERFACE=TYPE:enic-3
SERVICE2_INTERFACE=TYPE:enic-4
```

This example shows a boot parameters file for a CF with pass-through NICs, bonding configured and a DI internal network on a VLAN:

```
CARDSLOT=1
```

```
CARDTYPE=0x40010100
DI_INTERFACE=BOND:TYPE:enic-1,TYPE:enic-2
MGMT_INTERFACE=BOND:TYPE:ixgbe-3,TYPE:ixgbe-4
DI_INTERNAL_VLANID=10
```

VPC-DI Onboarding using ESC

You can use ESC to start an instance of the VPC-DI.

Onboarding the VPC-DI with ESC on OpenStack

This procedure describes how to onboard the VPC-DI on an instance of ESC in an OpenStack environment.

Before you begin

This procedure assumes that you have ESC created in a functioning OpenStack environment running release Juno or later with network access. For the detailed ESC installation procedure, refer to the *Cisco Elastic Services Controller 2.3 Install and Upgrade Guide*: http://www.cisco.com/c/en/us/td/docs/net_mgmt/elastic_services_controller/2-3/install/guide/Cisco-Elastic-Services-Controller-Install-Upgrade-Guide-2-3.html. For a description of the ESC configuration, refer to the *Cisco Elastic Services Controller 2.3 User Guide* : http://www.cisco.com/c/en/us/td/docs/net_mgmt/elastic_services_controller/2-3/user/guide/Cisco-Elastic-Services-Controller-User-Guide-2-3.html

Refer to the *Release Notes* to determine the version of Elastic Services Controller supported for this release.

Step 1 Get the qcow images of the VPC-DI instances for CF and SF.

The tarball file with the images is named something similar to `production.xxxxx.qvpc-di.qcow2.tgz`, depending on the release number. This archive includes two images for the CF and SF respectively: qvpc-di-cf.qcow2 and qvpc-di-xf.qcow2.

Step 2 Create the VPC-DI images in glance using the command **glance image-create**.

Example:

```
$ glance image-create --file qvpc-di-cf.qcow2 --container-format bare --disk-format
    qcow2 --is-public true --name cisco-qvpc-cf

$ glance image-create --file qvpc-di-xf.qcow2 --container-format bare --disk-format
    qcow2 --is-public true --name cisco-qvpc-xf
```

Step 3 Get the VPC-DI sample initialization tarball (vpc_esc_sample.tgz).

Step 4 Copy the VPC-DI sample initialization tarball to the ESC VM at the admin home (/home/admin/).

Step 5 Untar the VPC-DI sample initialization tarball to the vnf directory: opt/cisco/vnfs/cisco-qvpc/.

Step 6 Create the artifacts using the command line API command **esc_nc_cli edit-config** .

Example:

```
esc_nc_cli edit-config  /opt/cisco/vnfs/cisco-qvpc/dep/artifacts.xml
```

Step 7 Deploy the VPC-DI using the command **esc_nc_cli edit-config**.

Note Make sure to delete an existing deployment before redeploying the ESC. See Step 10.

The VPC-DI is deployed using the dep.xml file located in `/opt/cisco/vnfs/cisco-qvpc/dep/`. In general, you can use the default dep.xml file. If you need to customize the deployment, make any required changes to this file. For example, you can edit the chassis key that is used to create the chassis ID for your VPC-DI by editing the appropriate section in the dep.xml file:

```
<property>
  <name>CHASSIS_KEY</name>
  <value>164c03a0-eebb-44a8-87fa-20c791c0aa6d</value>
</property>
```

For a complete description of the dep.xml file, refer to: http://www.cisco.com/c/en/us/td/docs/net_mgmt/elastic_services_controller/2-2/deployment/Cisco-Elastic-Services-Controller-2-2-Deployment-Attributes.pdf

Example:

```
esc_nc_cli edit-config /opt/cisco/vnfs/cisco-qvpc/dep/dep.xml
```

This command may create a new tenant Core, if it does not already exist. Depending on which dep*.xml is used, the deployment might hit a "quota exceeded" error. If all SF and CFs do not boot, verify the default tenant quota and quota for tenant Core in OpenStack. The command to do this is `$ nova quota-defaults; nova quota-show --tenant Core`

Step 8 Verify the deployment status in the log file `var/log/esc/yangesc.log`.

Step 9 Wait for the VPC-DI to converge.

Step 10 To delete the VPC-DI deployment, use the ESC CLI command: **esc_nc_cli delete-dep** *Tenant deployment-name*

Example:

```
$ ./esc_nc_cli delete-dep Core cisco-qvpc
```

Step 11 (Optional) To use custom monitoring, before deploying the VPC-DI (Step 7) do the following:

a) Copy the mib file.

Example:

```
sudo cp /opt/cisco/vnfs/cisco-qvpc/config/starent.my /usr/share/snmp/mibs/
```

b) Add dynamic mapping metrics for the VPC-DI to any existing mappings. Merge the content of the file /opt/cisco/vnfs/cisco-qvpc/config/dynamic_mappings_snippet.xml to /opt/cisco/esc/esc-dynamic-mapping/dynamic_mappings.xml. Extra care should be taken for this step. For each SF, there needs to be one dynamic mapping in /opt/cisco/esc/esc-dynamic-mapping/dynamic_mappings.xml.

Customizing the VPC-DI Onboarding with ESC

After onboarding the VPC-DI with ESC, all VPC-DI files are located in the directory /opt/cisco/vnfs/cisco-qvpc/. There are numerous changes that can be made to the installation of your VPC-DI system. The following files can be altered as required.

VNF Deployment File

The deployment file is copied to your disk to the location: /opt/cisco/vnfs/cisco-qvpc/dep/.

You can create additional files as required.

Boot Parameter Files

You have several different boot parameter files for the various CF and SF VMs placed on your disk when you onboard the VPC-DI with ESC. These files are copied to: /opt/cisco/vnfs/cisco-qvpc/config/.

- param.cfg
- param_sf.cfg

These can be customized according to your needs. Refer to Creating a Boot Parameters File, on page 21 for more information on the boot parameters file format.

An example boot parameter file for a CF is shown here:

```
CARDSLOT=$SLOT_CARD_NUMBER
CPUID=0
CARDTYPE=$CARD_TYPE_NUM

DI_INTERFACE=TYPE:enic-1
VNFM_INTERFACE=TYPE:virtio_net-2
MGMT_INTERFACE=TYPE:virtio_net-3

VNFM_IPV4_ENABLE=true
VNFM_IPV4_DHCP_ENABLE=true
```

An example boot parameter file for an SF is shown here:

```
CARDSLOT=$SLOT_CARD_NUMBER
CPUID=0
CARDTYPE=$CARD_TYPE_NUM

DI_INTERFACE=TYPE:enic-1
VNFM_INTERFACE=TYPE:enic-2
SERVICE1_INTERFACE=TYPE:enic-3

VNFM_IPV4_ENABLE=true
VNFM_IPV4_DHCP_ENABLE=true
```

Configuration File

You can customize the configuration file located at /cisco/images/system.cfg. An example of a standard configuration file is shown here:

```
config
system hostname $VPC_HOSTNAME
clock timezone $TIMEZONE
context local
administrator admin password $ADMIN_PASS ftp

interface LOCAL1
ip address $CF_VIP_ADDR $CF_VIP_NETMASK
ip route 0.0.0.0 0.0.0.0 $NICID_1_GATEWAY LOCAL1
ip domain-lookup
ip domain-name $CF_DOMAIN_NAME
ip name-servers $CF_NAME_SERVER
ssh generate key
server sshd
subsystem sftp
port ethernet 1/1
```

```
bind interface LOCAL1 local
no shutdown
snmp community $SNMP_COMMUNITY read-only
end
```

Refer to Understanding Configuration Files, on page 67 for more information.

OpenStack Performance Optimizations

Cisco ESC allows a number of hypervisor optimizations using OpenStack Kilo release, such as non-uniform memory access (NUMA) node configuration, support for huge pages and the pinning of guest vCPUs.

- vCPU pinning—ability for guest vCPUs to be strictly pinned to a set of host physical CPUs. This prevents the potential wait by a vCPU for physical resources to become available.
- Large pages—allocation of larger blocks of memory to virtual resources.
- PCI-based NUMA scheduling—ability to assign a PCI device to an instance in OpenStack.

Note These OpenStack performance optimizations are supported using OpenStack Kilo version only.

OpenStack Kilo exposes various hardware acceleration features using a variety of mechanisms. One such mechanism involves the setting of key-value attributes on flavor objects. VM instantiation requests which reference a given flavor effectively request the corresponding hardware acceleration features. Provided the necessary OpenStack configuration is in place on the OpenStack control and compute nodes, the OpenStack compute service ("nova") selects an appropriate compute node, and assigns the corresponding resources.

To use hardware acceleration on the VPC-DI you must create new flavors in Cisco ESC and add the required metadata attributes using the NETCONF or REST interface. For information regarding flavors, refer to *Managing Flavors* in the *Cisco Elastic Services Controller User Guide*: http://www.cisco.com/c/en/us/support/cloud-systems-management/elastic-services-controller-2-1/model.html. You may also require a number of OpenStack configuration changes. The procedures for implementing hardware acceleration are described in these tasks.

Note Adding metadata is only supported for new flavors, not for existing ones. If metadata attributes need to be added to existing flavors, you must interact with OpenStack directly.

Configuring CPU Pinning

Step 1 On each OpenStack control node, configure the scheduler.

a) Configure the scheduler filters to include the NUMAToplogy and AggregateInstanceExtraSpec filters:

Example:

```
$ sudo vim /etc/nova/nova.conf
...
scheduler_default_filters=RetryFilter,AvailabilityZoneFilter,RamFilter,ComputeFilter,
ComputeCapabilitiesFilter,ImagePropertiesFilter,CoreFilter,NUMATopologyFilter,
```

```
AggregateInstanceExtraSpecsFilter
```

b) Restart the nova scheduler service.

Example:

```
$ sudo systemctl restart openstack-nova-scheduler.service
```

Step 2 On each relevant OpenStack compute node, configure which hypervisor processes are used for guests and which are not to be used for guests.

a) Ensure that hypervisor processes do not run on cores reserved for guests.

For example, to reserve cores 2, 3, 6 and 7 for guests, update the grub bootloader and reboot.

Example:

```
$ sudo grubby --update-kernel=ALL --args="isolcpus=2,3,6,7"
$ sudo grub2-install /dev/sda
$ sudo reboot
```

b) Verify the kernel command line reflects the change.

Example:

```
$ cat /proc/cmdline
...  isolcpus=2,3,6,7 ...
```

c) Configure guest virtual machine instances so that they are only allowed to run on specific cores and reserve RAM for hypervisor processes. For example, to use cores 2, 3, 6 and 7 and reserve 512 MB for hypervisor processes:

Example:

```
$ sudo vim /etc/nova/nova.conf
...
vcpu_pin_set=2,3,6,7
...
reserved_host_memory_mb=512
...
```

d) Restart the nova compute service.

Example:

```
$ sudo systemctl restart openstack-nova-compute.service
```

Step 3 Configure the global parameters.

a) Create a `performance-pinned` host aggregate for hosts that received pinning requests, and add an arbitrary attribute `pinned=true` to identify it.

Example:

```
$ nova aggregate-create performance-pinned
$ nova aggregate-set-metadata performance-pinned pinned=true
```

b) Create the normal aggregate for all other hosts and add the same arbitrary attribute, but set `pinned=false` to identify it.

Example:

```
$ nova aggregate-create normal
$ nova aggregate-set-metadata normal pinned=false
```

c) Add the previously enabled compute nodes to the `performance-pinned` host aggregate, and add all other compute nodes to the `normal` host aggregate.

Example:

```
$ nova aggregate-add-host normal            compute100.cloud.com
$ nova aggregate-add-host normal            compute101.cloud.com
$ nova aggregate-add-host normal            compute102.cloud.com
$ nova aggregate-add-host performance-pinned compute103.cloud.com
$ nova aggregate-add-host performance-pinned compute104.cloud.com
$ nova aggregate-add-host performance-pinned compute105.cloud.com
```

Step 4 Configure the flavor attributes using the Cisco ESC northbound API.

a) Set the flavor attributes.

- hw:cpu_policy=dedicated
- aggregate_instance_extra_specs:pinned=true

All instances created using this flavor are sent to hosts in host aggregates with pinned=true in their aggregate metadata.

Example:

```
 version='1.0' encoding='ASCII'?>
<esc_datamodel xmlns="http://www.cisco.com/esc/esc">
  <flavors>
    <flavor>
      <name>testfl6</name>
      <vcpus>1</vcpus>
      <memory_mb>2048</memory_mb>
      <root_disk_mb>10240</root_disk_mb>
      <ephemeral_disk_mb>0</ephemeral_disk_mb>
      <swap_disk_mb>0</swap_disk_mb>
      <properties>
        <property>
          <name>hw:cpu_policy</name><value>dedicated</value>
          <name>aggregate_instance_extra_specs:pinned</name><value>true</value>
        </property>
      </properties>
    </flavor>
  </flavors>
</esc_datamodel>
```

b) Update all other flavors to so that their extra specifications match them to the compute hosts in the normal aggregate.

Example:

```
$ nova flavor-key <flavor_Id> set aggregate_instance_extra_specs:pinned=false
```

c) To verify: launch a VM instance using the modified flavor and locate the compute node where the VM instance was started.

Example:

```
$ nova boot --image <test-image> --flavor <modified-flavor> test-instance
$ nova show test-instance | egrep
'OS-EXT-SRV-ATTR:hypervisor_hostname|OS-EXT-SRV-ATTR:instance_name'
| OS-EXT-SRV-ATTR:hypervisor_hostname  | compute3.cloud.com
```

```
| OS-EXT-SRV-ATTR:instance_name        | instance-00000cee
```

d) Log into the returned compute node and use the **virsh** tool to extract the XML of the returned instance.

Example:

```
$ ssh compute3.cloud.com
...
$ sudo virsh dumpxml instance-00000cee
...
  <vcpu placement='static' cpuset='2-3,6-7'>1</vcpu>
```

Configuring Huge Pages

Use this procedure to configure huge pages on your OpenStack configuration.

Step 1 Edit /etc/sysctrl.conf to configure the number of pages.

Example:

```
vm.nr_hugepages = 32768
```

Step 2 Edit /proc/meminfo to set the page size.

Example:

```
Hugepagesize: 2048 kB
```

Step 3 Create a flavor with hw:mem_page_size=2048 using the Cisco ESC northbound API.

Example:

```
<?xml version='1.0' encoding='ASCII'?>
<esc_datamodel xmlns="http://www.cisco.com/esc/esc">
  <flavors>
    <flavor>
      <name>testfl6</name>
      <vcpus>1</vcpus>
      <memory_mb>2048</memory_mb>
      <root_disk_mb>10240</root_disk_mb>
      <ephemeral_disk_mb>0</ephemeral_disk_mb>
      <swap_disk_mb>0</swap_disk_mb>
      <properties>
        <property>
          <name>hw:mem_page_size</name><value>2048</value>
        </property>
      </properties>
    </flavor>
  </flavors>
</esc_datamodel>
```

Configuring PCI Passthrough

The Intel VT-d extensions provide hardware support for directly assigning a physical device to a guest. This allows virtual devices to avoid the throughput loss and reduced packet forwarding capacity that would be involved in traversing multiple switches or bridges to reach a physical interface.

The VT-d extensions are required for PCI passthrough with Red Hat Enterprise Linux. The extensions must be enabled in the BIOS. Some system manufacturers disable these extensions by default.

This procedure does not cover enabling VT-d from the BIOS perspective; refer to your server manufacture BIOS configuration guide to enable VT-d. From the Linux kernel perspective, VT-d is enabled by adding "intel_iommu=on" to grub config.

Before you begin

VT-d must be enabled on the Intel chipset to support PCI Passthrough.

Step 1 Enable VT-d support on Red hat Enterprise Linux.

a) Edit "/etc/default/grub" and add "intel_iommu=on" to the end of the line: GRUB_CMDLINE_LINUX.

Example:

```
GRUB_TIMEOUT=5
GRUB_DEFAULT=saved
GRUB_DISABLE_SUBMENU=true
GRUB_TERMINAL_OUTPUT="console"
GRUB_CMDLINE_LINUX="rd.lvm.lv=rhel/swap crashkernel=auto rd.lvm.lv=rhel/root rhgb quiet
intel_iommu=on"
GRUB_DISABLE_RECOVERY="true"
```

b) Regenerate grub.conf and reboot your server by running the grub2-mkconfig command.

Example:

```
grub2-mkconfig -o /boot/grub2/grub.cfg on BIOS systems
```

or

```
grub2-mkconfig -o /boot/efi/EFI/redhat/grub.cfg on UEFI systems
```

c) Check that IOMMU gets activated by running the command **dmesg | grep -iE "dmar|iommu"**

Example:

Sample output from the **dmesg** command with IOMMU enabled:

```
[ 0.000000] Kernel command line: BOOT_IMAGE=/vmlinuz-3.10.0-229.el7.x86_64
root=/dev/mapper/rhel-root
ro rd.lvm.lv=rhel/swap crashkernel=auto rd.lvm.lv=rhel/root rhgb quiet intel_iommu=on
[ 0.000000] Intel-IOMMU: enabled
```

Step 2 Unbind the device from the Linux kernel.

a) For PCI passthrough to work, a device must be unbound from the Linux kernel driver. To accomplish this, pci_stub module is used.

Example:

```
Load pci_stub module "modprobe pci_stub"
```

b) Locate the network adapter you want to use for PCI passthrough. Run **lspci** and note the PCI address of the desired network card.

Example:

In this example, it is desired to use PCI device 15:00.0 for PCI passthrough.

```
12:00.0 Ethernet controller: Cisco Systems Inc VIC Ethernet NIC (rev a2)
13:00.0 Ethernet controller: Cisco Systems Inc VIC Ethernet NIC (rev a2)
14:00.0 Ethernet controller: Cisco Systems Inc VIC Ethernet NIC (rev a2)
15:00.0 Ethernet controller: Cisco Systems Inc VIC Ethernet NIC (rev a2)
```

c) Identify the Vendor ID and Device ID. Run **lspci -n**.

Example:

In this partial output, the Vendor ID 1137 and Device ID 0071 are identified.

```
11:00.0 0c04: 1137:0071 (rev a2)
12:00.0 0200: 1137:0071 (rev a2)
13:00.0 0200: 1137:0071 (rev a2)
14:00.0 0200: 1137:0071 (rev a2)
15:00.0 0200: 1137:0071 (rev a2)
```

d) Use this configuration to unbind the desired device from the Linux kernel driver.

Note that highlighted text must be changed for your device information.

Example:

```
echo "1137 0071" > /sys/bus/pci/drivers/pci-stub/new_id
echo 0000:15:00.0 > /sys/bus/pci/devices/0000:15:00.0/driver/unbind
echo 0000:15:00.0 > /sys/bus/pci/drivers/pci-stub/bind
```

e) Verify the success of these command by running **dmesg | grep stub**.

Example:

```
[ 276.705315] pci-stub 0000:15:00.0: claimed by stub
```

f) To make the changes persistent, add the pci-stub.ids to grub CMDLINE, update grub and reboot the host.

Note Note that this code applies to all vNICs with the specifiedVender/Device ID (1137:0071 in this example).

Example:

```
edit /etc/default/grub
GRUB_CMDLINE_LINUX="..pci-stub.ids=1137:0071"
grub2-mkconfig -o /boot/grub2/grub.cfg
reboot
```

Step 3 Configure nova.conf in OpenStack.

a) Specify which PCI devices are available for PCI passthrough, either using (vendorID, productID) combinations to allow any PCI device with that (vendorid, productid) combination to be passed through, or by specifying PCI addresses of devices allowed to be passed through.

```
pci_passthrough_whitelist={"vendor_id": "1137", "product_id": "0071"}
```

or

```
pci_passthrough_whitelist = [ {"address": "01:00.1"}, {"address": "02:00.1"} ]
```

b) Specify the PCI alias to (product ID, vendor ID) combination mapping. Note that this setting does currently not support PCI addresses, so all PCI devices on the whitelist have the same name.

Example:

```
pci_alias={"vendor_id":"1137", "product_id":"0071", "name":"nic1"}
```

c) Make the following additional changes to nova.conf

Example:

```
scheduler_driver=nova.scheduler.filter_scheduler.FilterScheduler
scheduler_available_filters=nova.scheduler.filters.all_filters
scheduler_available_filters=nova.scheduler.filters.pci_passthrough_filter.PciPassthroughFilter
scheduler_default_filters=RamFilter,ComputeFilter,AvailabilityZoneFilter,ComputeCapabilitiesFilter,ImagePropertiesFilter
```

d) Restart nova.

Example:

```
openstack-service restart nova
```

Step 4 Create a flavor with the attribute pci_passthrough:alias set to <PCI_DEVICE_ALIAS>:<NUM_DEVICES_REQUESTED>. The PCI_DEVICE_ALIAS references values from the pci_alias setting in /etc/nova/nova.conf.

Example:

```
$ cat fl.xml
<?xml version='1.0' encoding='ASCII'?>
<esc_datamodel xmlns="http://www.cisco.com/esc/esc">
    <flavors>
        <flavor>
            <name>testfl6</name>
              <vcpus>1</vcpus>
              <memory_mb>2048</memory_mb>
              <root_disk_mb>10240</root_disk_mb>
              <ephemeral_disk_mb>0</ephemeral_disk_mb>
              <swap_disk_mb>0</swap_disk_mb>
              <properties><property>
                  <name>pci_passthrough:alias</name><value>nic1g:1</value>
              </property></properties>
        </flavor>
    </flavors>
</esc_datamodel>
$ sudo /opt/cisco/esc/esc-confd/esc-cli/esc_nc_cli edit-config ./fl.xml
```

Onboarding the VPC-DI with Heat Orchestration Templates (HOT) in OpenStack

VPC-DI can be deployed as a Virtual Network Function (VNF) in an Openstack environment. VPC-DI runs as a collection of Virtual machines and the VMs have specific requirements with respect to storage, networking,

and configuration. In the Openstack environment, the Orchestrator is responsible for creating the objects required to bring up the VPC-DI VMs. The orchestrator is also responsible for creating and terminating the VMs and their associated objects. The orchestrator interfaces with Openstack services to create such entities in Openstack.

Openstack provides a service called HEAT orchestration templates (HOT) which defines the network, compute and storage topology of a VNF via a template. The HEAT template can be used as the blueprint for deploying an instance of the VNF.

The format of the templates and an example of the ENV parameters file are provided later in this section.

Step 1 Get the qcow images of the VPC-DI instances for CF and SF.

The tarball file with the images is named something similar to production.xxxxx.qvpc-di.qcow2.tgz, depending on the release number. When the tarball file is open there should be two images for the CF and SF: qvpc-di-cf.qcow2 and qvpc-di-xf.qcow2.

Step 2 Create all VPC-DI images in glance using the command **glance image-create**.

Example:

```
$ glance image-create --file qvpc-di-cf.qcow2 --container-format bare --disk-format
    qcow2 --is-public true --name cisco-qvpc-cf

$ glance image-create --file qvpc-di-sf.qcow2 --container-format bare --disk-format
    qcow2 --is-public true --name cisco-qvpc-sf
```

Step 3 Get the VPC-DI sample initialization tarball for HOT (vpc_HOT_sample.tgz).

Step 4 Copy the VPC-DI sample initialization tarball to your local machine.

Step 5 Untar the VPC-DI sample initialization tarball to any directory. There are two files with extensions **.yml** and **.env**.

Step 6 Edit the ENV file according to your openstack deployment.

Provide the values for your networks, availability zone, etc.

Browse to your local directory and click the **.yml** file for *Template source* and **.env** file for *Environment Source*

Step 7 Perform one of these:

- Deploy the VPC-DI using the OpenStack Dashboard by navigating to **Project>Orchestration>Stacks>Launch Stack**.
- Use the CLI to deploy the VPC-DI using HEAT with the command **heat stack-create -f di.yml -e di.env**.

Step 8 Verify the Status field is Complete, which indicates that there are Errors .

Step 9 Wait for the VPC-DI to converge.

Step 10 To delete the VPC-DI deployment, select the check box next to the stack name and click **Delete Stack**.

VPC-DI Heat Orchestration Templates

This section describes the format of the Heat templates. The VPC-DI HOT version is 2013-05-23. The template has these four sections: parameter-groups, parameters, resources, and outputs.

VPC-DI HOT Parameter Groups

The parameter_groups section allows for specifying how the input parameters should be grouped and the order to provide the parameters in. These groups are used to describe expected behavior for downstream user interfaces.

Table 6: HOT Parameters

Parameter Definition in Template	Notes

`- label: images`
`  description: CF and SF images in qvpc-di`
`  parameters:`
`  - qvpc_image_cf`
`  - qvpc_image_sf`
	List of images to be defined in the template

`- label: networks`
`  description: network configuration for DI`
`  parameters:`
`  - network_di_mgmt`
`  - network_di_internal`
`  - network_public`
`  - network_service1`
`  - network_service2`
`  - network_service3`
`  - network_service4`
	List of networks to be defined in the template
	Note This example lists four SF service ports. Up to 12 SF service portss can be defined.

VPC-DI HOT Parameters

The heat template defines a number of parameters for which you must provide values in an ENV file. Each of these parameters is described here. Each parameter definition is contained within the **parameters** section of the heat template. A sample ENV file follows the parameter descriptions.

Table 7: HOT Parameters

Parameter Definition in Template	Notes

`flavor_cf:`
`  type: string`
`  description: Flavor for Control Function VM`
`  default: m1.large`
	The name of the flavor to be used to create CFs. This can be one of the five default flavors, or a custom flavor that has been defined in OpenStack.

`flavor_sf:`
`  type: string`
`  description: Flavor for Service Function VM`
`  default: m1.large`
	The name of the flavor to be used to create SFs.

`availability_zone:`
`  type: string`
`  description: Availability_zone where the VNF`
`  should be created`
`  default: nova`
	Location in OpenStack where the VNF is created.

Parameter Definition in Template	Notes

`qvpc_image_cf:`
`  type: string`
`  label: Active CF image file in glance`
`  description: Active CF image ID or file in glance`
`  default: qvpc-di-<version>-cf.qcow2`
`  constraints:`
`  - custom_constraint: glance.image`
	Name of the VPC-DI image file for the CFs. This file must have been uploaded to glance.

`qvpc_image_xf:`
`  type: string`
`  label: SF image file in glance`
`  description: SF image ID or file in glance`
`  default: qvpc-di-<version>-xf.qcow2`
`  constraints:`
`  - custom_constraint: glance.image`
	Name of the VPC-DI image file for the service function VMs, that has been uploaded to glance.

Parameter Definition in Template	Notes

`network_public:`
`  type: string`
`  description: Network ID or Network Name of external network`
 	
`  default: public`
`  constraints:`
`  - custom_constraint: neutron.network`
	Network ID or name of the external network

`network_cf_mgmt:`
`  type: string`
`  description: Management Network ID or Name`
`  default: private`
`  constraints:`
`  - custom_constraint: neutron.network`
	Name or identifier for the VPC-DI management network.

`network_di_internal:`
`  type: string`
`  description: Unique QVPC-DI internal Network`
`  associated with this VNF`
`  default: private`
`  constraints:`
`  - custom_constraint: neutron.network`
	Name or identifier of the DI internal network. This is a private L2 network that interconnects the VMs in the VPC-DI.

`network_service#:`
`  type: string`
`  description: Network ID or Network Name of network`
`  to use for SF service ports`
`  default: cflocal`
`  constraints:`
`  - custom_constraint: neutron.network`
	Name or identifier of the service port. You can define between one and 12 service ports in each SF, where # represents this number. Each service port can perform a different service.

`  network_core:`
`  type: string`
`  description: core network for keepalives`
`  default: core`
`  constraints:`
`  - custom_constraint: neutron.network`
	Name or identifier of the core network used for keepalive messages.

Parameter Definition in Template	Notes

`qvpc_vip_addr:`
`  type: string`
`  description: OAM IP Address shared between CF01 and CF02`
 	
`  default: <value>`
`  constraints:`
`  - custom_constraint: ip_addr`
	Virtual IP address used between the CFs.

`qvpc_vip_gateway:`
`  type: string`
`  description: IP Address of Default Gateway for OAM Network`
 	
`  default: <value>`
`  constraints:`
`  - custom_constraint: ip_addr`
	The default gateway for the management network.

`vnf_name:`
`  type: string`
`  description: Unique name for this VNF instance`
`  default: qvpc_di`
	Unique name for this VNF instance. This name is used to identify the VNF.

`vnf_id:`
`  type: string`
`  description: Unique ID for this VNF instance`
`  default: 0`
	ID for the VNF instance.

`admin_password:`
`  type: string`
`  description: Default Administrator password for DI Access`
 	
`  default: Cisco123`	

`snmp_community:`
`  type: string`
`  description: READ SNMP string for this VPC instance`
`  default: public`	

`timezone:`
`  type: string`
`  description: TimeZone for this VF instance`
`  default: us-pacific`	

`cf_domain_name:`
`  type: string`
`  description: Domain for this VF instance`
`  default: localdomain`	

`az_cf<#>:`
`  type: string`
`  description: CF availability zone`
`  default: <value>`
	The availability zone for each of the two CF instances.

`az_sf<#>:`
`  type: string`
`  description: CF availability zone`
`  default:<value>`
	The availability zone for each of the SF instances.

Each of these parameters is defined for your VNF instance using an ENV file. An example ENV file is shown here:

```
parameters:
  # flavor defined for CF and SF in AIC
  flavor_cf: vsaegw_cf
  flavor_sf: vsaegw_sf

  # availability zone where the VNF instance should be deployed
  availability_zone: avzone-kvm-az01

  # vPC-DI glance images in qcow2
  qvpc_image_cf01: QVPCCF
  qvpc_image_sf: QVPCSF

  # Neutron Networks attached to vSAEGW instancenetwork_di_mgmt: oam_protected_net
  network_di_internal: saegw_di_internal_active_net
  network_service1: saegw_gn_net
  network_service2: saegw_sgi_net
  network_service3: saegw_support_net
  network_service4: saegw_icsr_li_net

  # VNF Instance Name
  vnf_name: qvpcDI_vsaegw

  # VNF Instance ID
  vnf_id: 01

  # Administrator user password
  admin_password: cisco123
```

```
parameters:
  flavor_cf:
    type: string
    description: Flavor for Control Function VM
    default: cisco-qvpc-cf
  flavor_sf:
    type: string
    description: Flavor for Service Function VM
    default: cisco-qvpc-xf
  qvpc_image_cf:
    type: string
    label: CF image file in glance
    description: CF image ID or file in glance
    default: qvpc-di-68031-cf.qcow2
    constraints:
      - custom_constraint: glance.image
  qvpc_image_sf:
    type: string
    label: SF image file in glance
    description: SF image ID or file in glance
    default: qvpc-di-68031-xf.qcow2
    constraints:
      - custom_constraint: glance.image
  network_public:
    type: string
    description: Network ID or Network Name of external network
    default: public
    constraints:
      - custom_constraint: neutron.network
  network_cf_mgmt:
    type: string
    description: Management Network ID or Name
    default: cf-mgmt
    constraints:
      - custom_constraint: neutron.network
  network_di_internal:
```

```
    type: string
    description: Unique QVPC-DI internal Network associated with this VNF
    default: di-internal
    constraints:
      - custom_constraint: neutron.network
  network_service1:
    type: string
    description: Transport Interface (Gn/S11/S1-u/S5) in to SAEGW Context
    default: service1
    constraints:
      - custom_constraint: neutron.network
  network_service2:
    type: string
    description: Transport Interface (Data, Voice, LI VLANs) in SGi Context
    default: service2
    constraints:
      - custom_constraint: neutron.network
   network_core:
    type: string
    description: core network for keepalives
    default: core
    constraints:
      - custom_constraint: neutron.network

# vip_addr and vip_gateway are automatically retrieved from the management network
  qvpc_vip_addr:
    type: string
    description: OAM IP Address shared between CF01 and CF02
    default: 172.16.181.2
    constraints:
      - custom_constraint: ip_addr
  qvpc_vip_gateway:
   type: string
   description: IP Address of Default Gateway for OAM Network
   default: 172.16.181.1
   constraints:
      - custom_constraint: ip_addr
  vnf_name:
    type: string
    description: Unique name for this VNF instance
    default: qvpc_di
  vnf_id:
    type: string
    description: Unique ID for this VNF instance
    default: 0
  admin_password:
    type: string
    description: Default Administrator password for DI Access
    default: Cisco123
  snmp_community:
    type: string
    description: READ SNMP string for this VPC instance
    default: public
  timezone:
    type: string
    description: TimeZone for this VF instance
    default: us-pacific
  cf_domain_name:
    type: string
    description: Domain for this VF instance
    default: localdomain
  az_cf1:
    type: string
    description: CF availability zone
```

```
    default: conway1
  az_cf2:
    type: string
    description: CF availability zone
    default: conway2
  az_sf3:
    type: string
    description: SF3 availability zone
    default: conway3
  az_sf4:
    type: string
    description: SF6 availability zone
    default: conway4
```

VPC-DI HOT Resources

The resources section of the template defines the control function (CF) and service function (SF) VMs as well as each of their ports.

Management Network

```
# Create port on management network and reserve a virtual IP address
qvpc_vip_port:
  type: OS::Neutron::Port
  properties:
      network: {get_param: network_di_mgmt}
      fixed_ips:
        - subnet_id: {get_param: subnet_id_di_mgmt}

# Associate a floating IP address to the virutal port
qvpc_vip_floating_ip:
  type: OS::Neutron::FloatingIP
  properties:
      floating_network: {get_param: network_public}
      port_id: {get_resource: qvpc_vip_port}
```

The VIP port is the virtual IP port used to access the VPC-DI. The VIP port IP address is configurable in the Day 0 configuration.

HOT Resources for CF

The heat template must define each of the two CF VMs being used by the VNF. This definition includes configuring the port that connects to the DI internal network, as well as the port that connects to the CF management network, specifying the StarOS boot parameter file and the StarOS Day 0 configuration file. The definition of the first CF is shown here with an explanation; the second CF is defined in a similar way.

CF DI Internal Network

This section creates the CF DI internal network. Use this section twice, once for each of the two CFs that must be configured. # is either 1 or 2.

```
# Port connected to unique DI-network
qvpc_cf_0#_port_int:
  type: OS::Neutron::Port
  properties:
    network: {get_param: network_di_internal}
    allowed_address_pairs:
```

```
      -ip_address: "172.16.0.0/18"
```

qvpc_cf_#_port_int is port connected to the DI internal network. The value of the network is extracted from the parameter **network_di_internal** which is retrieved from the ENV file.

The property allowed_address_pairs must be in each di-internal port. Because the di_internal port is assigned an IP address by the VPC-DI in the 17.16.0.0/18 network which is different from its address in neutron, we need to configure the allowed_address_pairs property to allow traffic on those address to pass through the port. The allowed address pair extension extends the port attribute to enable you to specify arbitrary MAC address or IP address (CIDR) pairs that are allowed to pass through a port regardless of the subnet associated with the network.

CF Management Network

This section creates the CF management network. Use this section twice, once for each of the two CFs that must be configured. # is either 1 or 2.

```
# Port connected to the management network
qvpc_cf_0#_port_mgmt:
  type: OS::Neutron::Port
  properties:
    network: {get_param: network_di_mgmt}
    allowed_address_pairs:
      - ip_address: {get_param: qvpc_vip_addr}
```

qvpc_cf_#_port_mgmt represents the port definition of the port connected to the OAM network. The value is extracted from the parameter **network_di_mgmt** which is retrieved from the ENV file.

SSH Keys

DI inter-VM communication is now only possible via authentication through externally supplied SSH keys. These keys are passed as part of the HEAT deployment. Public and private keys are required.

Generate the public and private SSH keys. Create a file called *user_key.pub* containing the public key. Create a file called *user_key* containing the private key. Ensure that both of these files are stored on the configuration drive. These files are referenced by HEAT:

```
      personality:
        "user_key.pub": |
                    ssh-rsa
<public_key>
        "user_key": |
                    -----BEGIN RSA PRIVATE KEY-----
<private_key>
                    -----END RSA PRIVATE KEY-----
```

Create CF VM

This section creates the CF VM. Use this section twice, once for each of the two CFs that must be created. # is either 1 or 2.

```
qvpc_cf_0#:
  type: OS::Nova::Server
  properties:
    # Create VM of format "<vnf_name>_cf_0#"
    name:
```

```
  str_replace:
    template: ${VF_NAME}_cf_0#
    params:
      ${VF_NAME}: {get_param: vnf_name}
# Use active CF image and CF Flavor
image: {get_param: qvpc_image_cf1 }
flavor: {get_param: flavor_cf }
networks:
  - port: {get_resource: qvpc_cf_0#_port_int}
  - port: {get_resource: qvpc_cf_0#_port_mgmt}
config_drive: True
```

The CF VM (**qvpc_cf_#**) is created with the previously defined parameters and named according to the convention "<vnf_name>_cf_#". The **vnf_name** is retrieved from the ENV file as are the image and flavor to be used to create the VNF.

StarOS Day 0 Configuration

The Day 0 configuration provided here configures the DI interface, system hostname and enables SSH and SFTP access using *personality* properties.

```
# Metadata to provide cloud-init capability to VPC-DI
  personality:
    "staros_param.cfg":
        str_replace:
            template: |
                CARDSLOT=$CARD_NUMBER
                CARDTYPE=$CARD_TYPE
                CPUID=$UUID
                DI_INTERFACE_MTU=1500
                DI_INTERFACE=TYPE:virtio_net-1
                MGMT_INTERFACE=TYPE:virtio_net-2
                VNFM_INTERFACE=TYPE:virtio_net-3
                VNFM_IPV4_ENABLE=true
                VNFM_IPV4_DHCP_ENABLE=true
                VNFM_PROXY_ADDRS=192.168.180.92,192.168.180.91,192.168.180.93
            params:
                $CARD_NUMBER: 1
                $CARD_TYPE: "0x40030100"
                $UUID: 0
    "staros_config.txt":
        str_replace:
            template: |
                config
                    system hostname $VF_NAME-cf-$CARD_NUMBER
                    clock timezone $TIMEZONE
                    ssh key-gen wait-time 0
                    context local
                        administrator admin password $ADMIN_PASS ftp
                            interface LOCAL1
                                ip address $CF_VIP_ADDR 255.255.255.0
                        #exit
                        ip route 0.0.0.0 0.0.0.0 $CF_VIP_GATEWAY LOCAL1
                        ip domain-lookup
                        ip domain-name $CF_DOMAIN_NAME
                        ip name-servers $CF_VIP_GATEWAY
                        ssh generate key
                        server sshd
                            subsystem sftp
                        #exit
                        server confd
```

```
                                confd-user admin
                            #exit
                            port ethernet 1/1
                                bind interface LOCAL1 local
                                no shutdown
                            #exit
                            snmp community $SNMP_COMMUNITY read-only
                        end
                    params:
                        $CARD_NUMBER: 1
                        $VF_NAME: {get_param: vnf_name}
                        $TIMEZONE: {get_param: timezone}
                        $ADMIN_PASS: {get_param: admin_password}
                        $SNMP_COMMUNITY: {get_param: snmp_community}
                        $CF_DOMAIN_NAME: {get_param: cf_domain_name}
                        $SLOT_CARD_NUMBER: 1
                        #$CF_VIP_ADDR: {get_attr: [qvpc_vip_port, fixed_ips, 0, ip_address]}
                        $CF_VIP_ADDR: 172.16.181.2
                        #$CF_VIP_GATEWAY: { get_attr: [qvpc_vip_port, subnets, 0, gateway_ip]
}
                        $CF_VIP_GATEWAY: 172.16.181.1
        "user_key.pub": |
                        ssh-rsa
<public_key>
        "user_key": |
                        -----BEGIN RSA PRIVATE KEY-----
<private_key>
                        -----END RSA PRIVATE KEY-----
```

$CARD_NUMBER refers to the number of the slot, which here is 1 but is 2 for the second CF.

HOT Resources for SF

Use the heat template to define each of the service function (SF) VMs that you want to deploy in the VPC-DI. For each SF you must configure the port to connect to the DI internal network as well as each of the service ports that you need for the SF. You can configure up to 12 service ports. This example creates a single SF that is used for an SAE gateway with four service ports. You must repeat a similar configuration for each SF required.

Define The Ports in the SF

```
# Create port for DI-Internal Network
  qvpc_sf_03_port_int:
    type: OS::Neutron::Port
    properties:
        network: {get_param: network_di_internal}
        allowed_address_pairs:
          - ip_address: "172.16.0.0/18"
```

qvpc_sf_#_port_int is the port that connects to the internal DI network. # is the number of the SF and can range from 3 to the maximum number of SFs allowed. The value of the network is extracted from the parameter **network_di_internal** which is retrieved from the ENV file.

```
# Create first service port (document as per your use)
  qvpc_sf_03_port_svc_01:
    type: OS::Neutron::Port
    properties:
```

```
      network: {get_param: network_service1}
```

qvpc_sf_#_port_svc_01 is the first service port. Ports are numbered consecutively from 1 to 12. The value of the network is extracted from the parameter **network_service1** which is retrieved from the ENV file.

```
# Create second service port (document as per your use)
  qvpc_sf_03_port_svc_02:
    type: OS::Neutron::Port
    properties:
      network: {get_param: network_service2}
      allowed_address_pairs:
        - ip_address: "192.168.10.0/24"
# Create third service port  (document as per your use)
  qvpc_sf_03_port_svc_03:
    type: OS::Neutron::Port
    properties:
      network: {get_param: network_service3}
# Create forth service port  (document as per your use)
  qvpc_sf_03_port_svc_04:
    type: OS::Neutron::Port
    properties:
      network: {get_param: network_service4}
```

The remaining three service ports are created - each retrieving the network information from the ENV file. Additional service ports can be created as required.

Create SF VM

```
qvpc_sf_03:
  type: OS::Nova::Server
  properties:
# Create VM name of format "<vnf_name>_sf_0<num>"
    name:
      str_replace:
          template: ${VF_NAME}_sf_03
          params:
            ${VF_NAME}: {get_param: vnf_name}
# Use SF image and SF Flavor
    image: { get_param: qvpc_image_sf }
    flavor: { get_param: flavor_sf }
    networks:
      - port: {get_resource: qvpc_sf_03_port_int}
      - port: {get_resource: qvpc_sf_03_port_svc_01}
      - port: {get_resource: qvpc_sf_03_port_svc_02}
      - port: {get_resource: qvpc_sf_03_port_svc_03}
      - port: {get_resource: qvpc_sf_03_port_svc_04}
    config_drive: True
```

The SF **qvpc_sf_#** is created with the name of the format 'vnf_name_sf_0#', where vnf_name is VNF name value retrieved from the ENV file and # is the slot of the SF. The values of the service ports are previously defined in the heat template. The image and flavor are also taken from the ENV file.

Each SF is defined similarly in the template.

Personality Configuration

Day 0 and Day 1 configurations are injected into the VNF using personality properties. The VPC-DI applies personality properties to the system, and expects this metadata from the HEAT template as shown here.

The personality defines the boot parameters file. Refer to Configuring Boot Parameters, on page 27 for more information on the boot parameters.

```
# Associate VM to unique slot (>2) and identify that its a SF
       config_drive: True
      personality:
        "staros_param.cfg":
            str_replace:
                template: |
                    CARDSLOT=$CARD_NUMBER
                    CARDTYPE=$CARD_TYPE
                    CPUID=$UUID
                    DI_INTERFACE_MTU=1500
                params:
                    $CARD_NUMBER: 3
                    $CARD_TYPE: "0x42070100"
                    $UUID: 0
        "user_key.pub": |
                    ssh-rsa
<public_key>
        "user_key": |
                    -----BEGIN RSA PRIVATE KEY-----
                    <private_key>
                    -----END RSA PRIVATE KEY-----
```

DI inter-VM communication is now only possible via authentication through externally supplied SSH keys. These keys are passed as part of the HEAT deployment. Public and private keys are required.

Generate the public and private SSH keys. Create a file called *user_key.pub* containing the public key. Create a file called *user_key* containing the private key. Ensure that both of these files are stored on the configuration drive. These files are referenced by HEAT as shown above.

VPC-DI HOT Outputs

The outputs section of the heat template defines the outputs from using the template. You can see the outputs by going to **Project>Orchestration>Stacks** and selecting the heat stack that you deployed. In the **Overview** tab you see any outputs from the heat stack.

You can also see output from the heat stack by running the **heat stack-show $[stack_name]** command at the command line.

Examples of types of output you might define for the VPC-DI are shown here:

```
qvpc_floating_ip:
  description: Floating IP of qvpc-di VIP
  value: { get_attr: [qvpc_vip_floating_ip, floating_ip_address]}
CF1_networks:
  description: The networks of the deployed CF-1
  value: { get_attr: [qvpc_cf_01, networks] }
CF2_networks_2:
  description: The networks of the deployed CF-2
  value: { get_attr: [qvpc_cf_02, networks] }
port_1_int:
  description: The port of the deployed server 1, di-internal
  value: { get_attr: [qvpc_cf_01_port_int, mac_address] }
port_1_mgmt:
  description: The port of the deployed server 1, cf-mgmt
  value: { get_attr: [qvpc_cf_01_port_mgmt, mac_address] }
port_2_int:
```

```
    description: The port of the deployed server 2, di-internal
    value: { get_attr: [qvpc_cf_02_port_int, mac_address] }
port_2_mgmt:
    description: The port of the deployed server 2, cf-mgmt
    value: { get_attr: [qvpc_cf_02_port_mgmt, mac_address] }
```

VMware Installation Notes

DI inter-VM communication is now only possible via authentication through externally supplied SSH keys. Public and private keys are required. These keys must be supplied prior to booting the VM as part of an ISO.

The keys must be generated on an external host and packaged in the ISO which must then be attached to the VM. The keys and the ISO files are generated as follows:

```
$ mkdir iso
$ ssh-keygen -t rsa -N "" -C "root@localhost" -f iso/user_key
$ genisoimage -o vpcdi_keys.iso iso
```

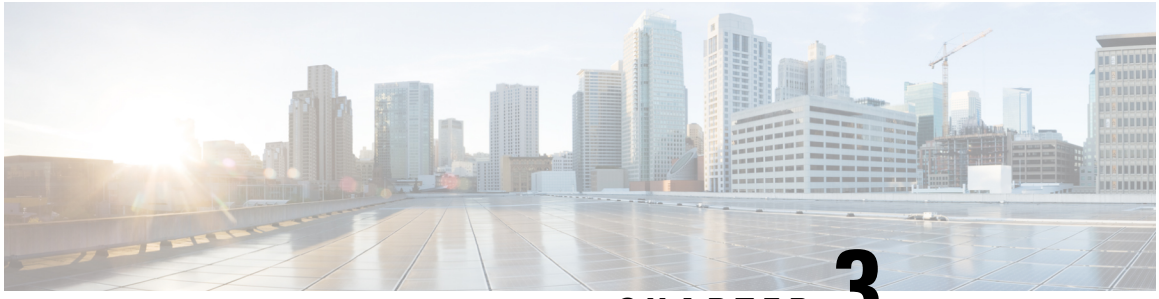
Once the ISO file is generated, power-up the VM and map to the CD-DVD ROM. From within vSphere, this is done by selecting the VM (CF or SF) from the list and clicking on the CD/DVD icon from the option bar near the top. Then select **Connect to ISO image on local disk** and choose the ISO. Repeat this for all of the VMs (CFs and SFs).

Once the keys are mapped, point the VPC-DI boot configuration to the image by setting the right boot priority and reload the VPC-DI.

Rules for VM Recovery

When you create a spare VM by cloning an old VM, you will encounter a mac address mismatch issue. The following are the VM Recovery rules for replacing to spare VM:

1. Do not use the VM suspend/resume.
2. You can shutdown/reboot the VM. It owns the slot as long as it is not deleted.
3. You can replace the faulty VM with the new one by recreating it only after deleting the faulty VM. The new VM can use the slot of the faulty VM.
4. If you want to bring back the faulty VM, you can recreate it and assign a slot that is not currently in use.



CHAPTER 3

System Operation and Configuration

Cisco VPC-DI provides subscriber management services for Mobile Packet Core networks.

Before you connect to the StarOS command line interface (CLI) and begin the configuration process, it is useful to understand how StarOS supports these services. This chapter provides terminology and background information to consider before you configure StarOS.

Terminology

This section defines important terms used throughout this guide.

Contexts

A context is a logical grouping or mapping of configuration parameters that pertain to various physical ports, logical IP interfaces, and services. A context can be thought of as a virtual private network (VPN).

The system supports the configuration of multiple contexts. Each context is configured and operates independently of the others. Once a context has been created, administrative users can configure services, logical IP interfaces, and subscribers for that context and then bind the logical interfaces to physical ports.

You can also assign a domain alias to a context; if a subscriber's domain name matches one of the configured alias names for a context, that context is used.

Logical Interfaces

You must associate a port with a StarOS virtual circuit or tunnel called a *logical interface* before the port can allow the flow of user data.Within StarOS, a logical interface is a named interface associated with a virtual router instance that provides higher-layer protocol transport, such as Layer 3 IP addressing. Interfaces are configured as part of VPN contexts and are independent from the physical ports that will be used to bridge the virtual interfaces to the network.

Logical interfaces are associated with ethernet+ppp+tunnel addresses and are bound to a specific port during the configuration process. Logical interfaces are also associated with services through bindings. Services are bound to an IP address that is configured for a particular logical interface. When associated, the interface takes on the characteristics of the functions enabled by the service.

There are several types of logical interfaces to configure to support Simple and Mobile IP data applications. These are briefly defined below.

Management Interface

This interface provides the point of attachment to the management network. The interface supports remote access to the StarOS command line interface (CLI). It also supports event notification via the Simple Network Management Protocol (SNMP).

Bindings

A binding is an association between elements within the system. There are two types of bindings: static and dynamic.

Static binding is accomplished through system configuration. Static bindings associate:

- A specific logical interface (configured within a particular context) to a physical port. Once the interface is bound, traffic can flow through the context as if it were any physically-defined circuit. Static bindings support any encapsulation method over any interface and port type.
- A service to an IP address assigned to a logical interface within the same context. This allows the interface to take on the characteristics (that is, support the protocols) required by the service.

Dynamic binding associates a subscriber to a specific egress context based on the configuration of their profile or system parameters. This provides a higher degree of deployment flexibility, as it allows a wireless carrier to support multiple services and facilitates seamless connections to multiple networks.

Management ports can only be bound in the local context. Traffic or subscriber ports can only be bound in a non-local context.

Services

Configure services within a context to enable certain functionality. The following are examples of services you can configure on the system, subject to licensing availability and platform type:

- Gateway GPRS Support Node (GGSN) services
- Serving GPRS Support Node (SGSN) Services
- Packet Data Serving Node (PDSN) services
- Home Agent (HA) services
- Layer 2 Tunneling Protocol Access Concentrator (LAC) services
- Dynamic Host Control Protocol (DHCP) services
- Mobility Management Entity (MME) Services
- PDN Gateway (P-GW) Services
- Serving Gateway (S-GW) Services

- Intelligent Policy Control Function (IPCF) Services (PCC-Service, PCC-Policy, PCC-AF)

AAA Servers

Authentication, Authorization and Accounting (AAA) servers store profiles, perform authentication, and maintain accounting records for each mobile data subscriber. The AAA servers communicate with the system over an AAA interface. The system supports the configuration of up to 128 interfaces to AAA servers.

It is important to note that for Mobile IP, there can be Foreign AAA (FAAA) and Home AAA (HAAA) servers. FAAA servers typically reside in the carrier's network. HAAA servers could be owned and controlled by either the carrier or the home network. If the HAAA server is owned and controlled by the home network, accounting data is transferred to the carrier via an AAA proxy server.

Important Mobile IP support depends on the availability and purchase of a license bundle that includes Home Agent (HA).

Subscribers

Subscribers are the end-users of the service; they gain access to the Internet, their home network, or a public network through the system.

There are three primary types of subscribers:

- **RADIUS-based Subscribers:** The most common type of subscriber, these users are identified by their International Mobile Subscriber Identity (IMSI) number, an Electronic Serial Number (ESN), or by their domain name or user name. They are configured on and authenticated by a RADIUS AAA server.

  Upon successful authentication, various attributes that are contained in the subscriber profile are returned. The attributes dictate such things as session parameter settings (for example, protocol settings and IP address assignment method), and what privileges the subscriber has.

  Important Attribute settings received by the system from a RADIUS AAA server take precedence over local-subscriber attributes and parameters configured on the system.

- **Local Subscribers**: These are subscribers, primarily used for testing purposes, that are configured and authenticated within a specific context. Unlike RADIUS-based subscribers, the local subscriber's user profile (containing attributes like those used by RADIUS-based subscribers) is configured within the context where they are created.

  When local subscriber profiles are first created, attributes for that subscriber are set to the system's default settings. The same default settings are applied to all subscriber profiles, including the subscriber named *default* which is created automatically by the system for each system context. When configuring local profile attributes, the changes are made on a subscriber-by-subscriber basis.

Important Attributes configured for local subscribers take precedence over context-level parameters. However, they *could* be over-ridden by attributes returned from a RADIUS AAA server.

- **Management Subscribers**: A management user is an authorized user who can monitor, control, and configure the system through the CLI. Management is performed either locally, through the system Console port, or remotely through the use of the Telnet or secure shell (SSH) protocols. Management users are typically configured as a local subscriber within the Local context, which is used exclusively for system management and administration. As with a local subscriber, a management subscriber's user profile is configured within the context where the subscriber was created (in this case, the Local context). However, management subscribers may also be authenticated remotely via RADIUS, if an AAA configuration exists within the local context, or TACACS+.

How the System Selects Contexts

This section describes the process that determines which context to use for context-level administrative users or subscriber sessions. Understanding this process allows you to better plan your configuration in terms of how many contexts and interfaces you need to configure.

Context Selection for Context-level Administrative User Sessions

The system comes configured with a context called *local* that you use specifically for management purposes. The context selection process for context-level administrative users (those configured within a context) is simplified because the management ports on the MIO are associated only with the Local context. Therefore, the source and destination contexts for a context-level administrative user responsible for managing the entire system should always be the local context.

A context-level administrative user can be created in a non-local context. These management accounts have privileges only in the context in which they are created. This type of management account can connect directly to a port in the context in which they belong, if local connectivity is enabled (SSHD, for example) in that context.

For all FTP or SFTP connections, you must connect through an MIO management interface. If you SFTP or FTP as a non-local context account, you must use the username syntax of *username@contextname*.

Important In release 20.0 and higher Trusted StarOS builds, FTP is not supported.

The context selection process becomes more involved if you are configuring the system to provide local authentication or work with a AAA server to authenticate the context-level administrative user.

The system gives you the flexibility to configure context-level administrative users locally (meaning that their profile will be configured and stored in its own memory), or remotely on an AAA server. If a locally-configured user attempts to log onto the system, the system performs the authentication. If you have configured the user profile on an AAA server, the system must determine how to contact the AAA server to perform authentication. It does this by determining the AAA context for the session.

The following table and flowchart describe the process that the system uses to select an AAA context for a context-level administrative user. Items in the table correspond to the circled numbers in the flowchart.

Figure 5: Context-level Administrative User AAA Context

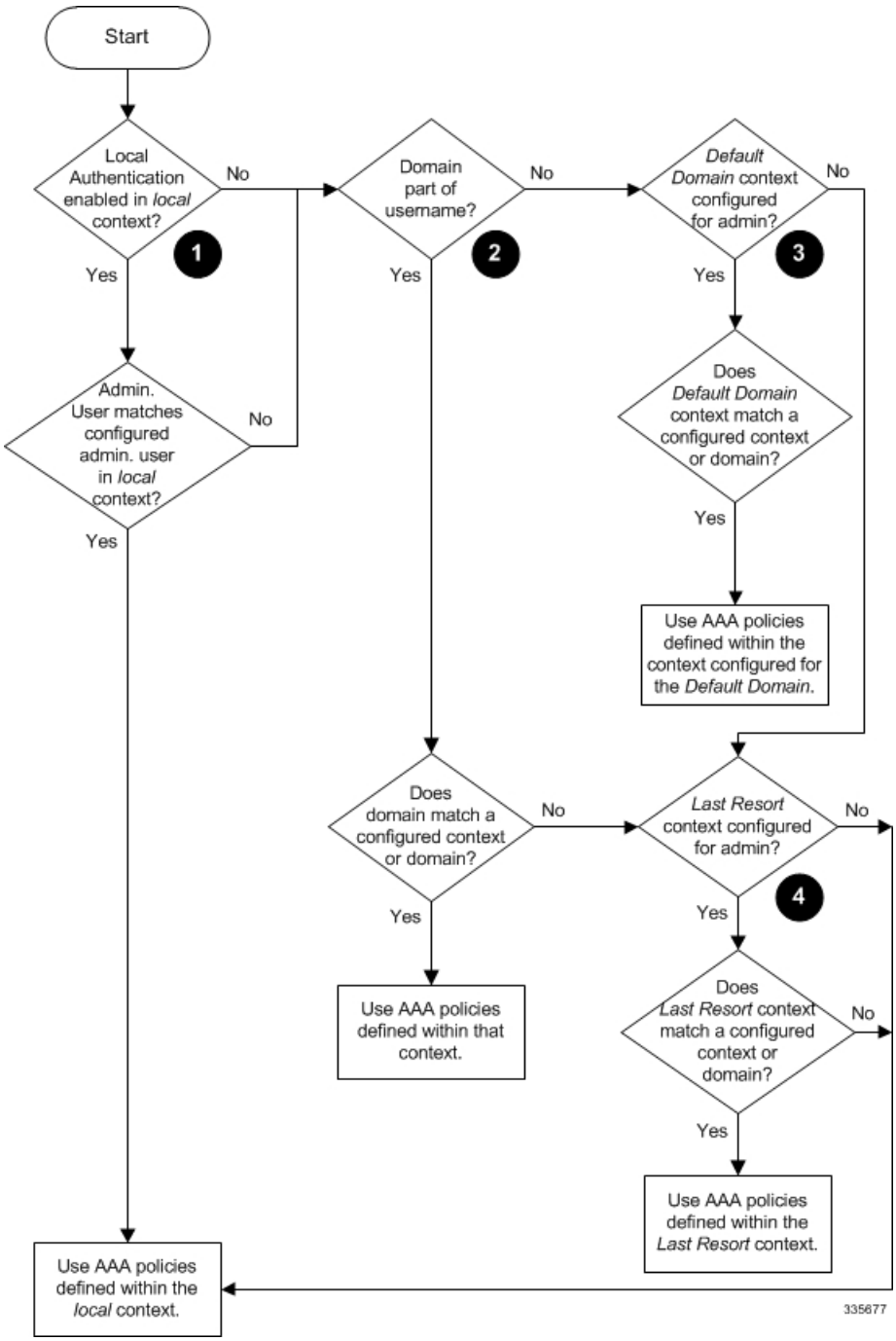


Table 8: Context-level Administrative User AAA Context Selection

Item	Description
1	During authentication, the system determines whether local authentication is enabled in the *local* context. If it is, the system attempts to authenticate the administrative user in the *local* context. If it is not, proceed to item 2 in this table. If the administrative user's username is configured, authentication is performed by using the AAA configuration within the *local* context. If not, proceed to item 2 in this table.
2	If local authentication is disabled on the system or if the administrative user's username is not configured in the *local* context, the system determines if a domain was received as part of the username. If there is a domain and it matches the name of a configured context or domain, the systems uses the AAA configuration within that context. If there is a domain and it does not match the name of a configured context or domain, Go to item 4 in this table. If there is no domain as part of the username, go to item 3 in this table.
3	If there was no domain specified in the username or the domain is not recognized, the system determines whether an *AAA Administrator Default Domain* is configured. If the default domain is configured and it matches a configured context, the AAA configuration within the *AAA Administrator Default Domain* context is used. If the default domain is not configured or does not match a configured context or domain, go to item 4 item below.
4	If a domain was specified as part of the username but it did not match a configured context, or if a domain was not specified as part of the username, the system determines if the *AAA Administrator Last Resort context parameter* is configured. If a last resort, context is configured and it matches a configured context, the AAA configuration within that context is used. If a last resort context is not configured or does not match a configured context or domain, the AAA configuration within the *local* context is used.

In Release 21.4 and higher (Trusted builds only):

- Users can only access the system through their respective context interface.

- If the user attempts to log in to their respective context through a different context interface, that user will be rejected.
- Irrespective of whether the users are configured in any context with 'authorized-keys' or 'allowusers', with this feature these users will be rejected if they attempt to log in via any other context interface other than their own context interface.
- Users configured in any non-local context are required to specify which context they are trying to log in to. For example:

  ssh username@ctx_name@ctx_ip_addrs

Context Selection for Subscriber Sessions

The context selection process for a subscriber session is more involved than that for the administrative users. Subscriber session context selection information for specific products is located in the *Administration Guide* for the individual product.

Understanding Configuration Files

The system supports the use of a file or script to modify configurable parameters. Using a file for offline system configuration reduces the time it takes to configure parameters on multiple systems.

A system configuration file is an ASCII text file that contains commands and configuration parameters. When you apply the configuration file, the system parses through the file line-by-line, testing the syntax and executing the command. If the syntax is incorrect, a message is displayed to the CLI and the system proceeds to the next command. Lines that begin with # are considered remarks and are ignored.

Important Pipes (|), used with the **grep** and **more** keywords, can potentially cause errors in configuration file processing. Therefore, the system automatically ignores keywords with pipes during processing.

Important Always save configuration files in UNIX format. Failure to do so can result in errors that prevent configuration file processing.

The commands and configuration data within the file are organized and formatted just as they would be if they were being entered at the CLI prompt. For example, if you wanted to create a context called *source* in the CLI, you would enter the following commands at their respective prompts:

```
[local]host_name# config
[local]host_name(config)# context source
[source]host_name(config-ctx)# end
```

To create a context called *source* using a configuration file, you would use a text editor to create a new file that consists of the following:

```
config
      context source
      end
```

There are several important things to consider when using configuration files:

- The system automatically applies a configuration file at the end of the boot process. After the system boots up for the first time, a configuration file that you have created and that is tailored to your network needs, can be applied. To make the system use your configuration file, modify the system's boot parameters according to the instructions located in *Software Management Operations*.
- In addition to being applied during the boot process, you can also apply configuration files manually at any time by executing the appropriate commands at the CLI prompt. Refer to the instructions in *Software Management Operations*.

Important When you apply a configuration file after the boot process, the file does not delete the configuration loaded as part of the boot process. Only those commands that are duplicated are overwritten.

- Configuration files can be stored in any of the following locations:
  - **USB Memory Stick:** Supported via a USB port on the active MIO (/usb1).
  - **Network Server:** Any workstation or server on the network that the system can access using the Secure File Transfer Protocol (SFTP). This is recommended for large network deployments in which multiple systems require the same configuration.
  - **/flash**: a solid-state device with limited storage.
  - **/hd-raid**: internal RAID storage.
- Each time you save configuration changes you made during a CLI session, you can save those settings to a file which you can use as a configuration file.

IP Address Notation

When configuring a port interface via the CLI you must enter an IP address. The CLI always accepts an IPv4 address, and in some cases accepts an IPv6 address as an alternative.

For some configuration commands, the CLI also accepts CIDR notation. Always view the online Help for the CLI command to verify acceptable forms of IP address notation.

IPv4 Dotted-Decimal Notation

An Internet Protocol Version 4 (IPv4) address consists of 32 bits divided into four octets. These four octets are written in decimal numbers, ranging from 0 to 255, and are concatenated as a character string with full stop delimiters (dots) between each number.

For example, the address of the loopback interface, usually assigned the host name localhost, is 127.0.0.1. It consists of the four binary octets 01111111, 00000000, 00000000, and 00000001, forming the full 32-bit address.

IPv4 allows 32 bits for an Internet Protocol address and can, therefore, support 2^{32} (4,294,967,296) addresses.

IPv6 Colon-Separated-Hexadecimal Notation

An Internet Protocol Version 6 (IPv6) address has two logical parts: a 64-bit network prefix, and a 64-bit host address part. An IPv6 address is represented by eight groups of 16-bit hexadecimal values separated by colons (:).

A typical example of a full IPv6 address is 2001:0db8:85a3:0000:0000:8a2e:0370:7334

The hexadecimal digits are case-insensitive.

The 128-bit IPv6 address can be abbreviated with the following rules:

- Leading zeroes within a 16-bit value may be omitted. For example, the address fe80:0000:0000:0000:0202:b3ff:fe1e:8329 may be written as fe80:0:0:0:202:b3ff:fe1e:8329
- One group of consecutive zeroes within an address may be replaced by a double colon. For example, fe80:0:0:0:202:b3ff:fe1e:8329 becomes fe80::202:b3ff:fe1e:8329

IPv6 allows 128 bits for an Internet Protocol address and can support 2^{128} (340,282,366,920,938,000,000,000,000,000,000,000,000) internet addresses.

CIDR Notation

Classless Inter-Domain Routing (CIDR) notation is a compact specification of an Internet Protocol address and its associated routing prefix. It is used for both IPv4 and IPv6 addressing in networking architectures.

CIDR is a bitwise, prefix-based standard for the interpretation of IP addresses. It facilitates routing by allowing blocks of addresses to be grouped into single routing table entries. These groups (CIDR blocks) share an initial sequence of bits in the binary representation of their IP addresses.

CIDR notation is constructed from the IP address and the prefix size, the latter being the number of leading 1 bits of the routing prefix. The IP address is expressed according to the standards of IPv4 or IPv6. It is followed by a separator character, the slash (/) character, and the prefix size expressed as a decimal number.

The address may denote a single, distinct, interface address or the beginning address of an entire network. In the latter case the CIDR notation specifies the address block allocation of the network. The maximum size of the network is given by the number of addresses that are possible with the remaining, least-significant bits below the prefix. This is often called the host identifier.

For example:

- the address specification 192.168.100.1/24 represents the given IPv4 address and its associated routing prefix 192.168.100.0, or equivalently, its subnet mask 255.255.255.0.
- the IPv4 block 192.168.0.0/22 represents the 1024 IPv4 addresses from 192.168.0.0 to 192.168.3.255.
- the IPv6 block 2001:DB8::/48 represents the IPv6 addresses from 2001:DB8:0:0:0:0:0:0 to 2001:DB8:0:FFFF:FFFF:FFFF:FFFF:FFFF.
- ::1/128 represents the IPv6 loopback address. Its prefix size is 128, the size of the address itself, indicating that this facility consists of only this one address.

The number of addresses of a subnet defined by the mask or prefix can be calculated as $2^{\text{address size - mask}}$, in which the address size for IPv4 is 32 and for IPv6 is 128. For example, in IPv4, a mask of /29 gives $2^{32-29} = 2^3 = 8$ addresses.

Alphanumeric Strings

Some CLI commands require the entry of an alphanumeric string to define a value. The string is a contiguous collection of alphanumeric characters with a defined minimum and maximum length (number of characters).

Character Set

The alphanumeric character set is a combination of alphabetic (Latin letters) and/or numeric (Arabic digits) characters. The set consists of the numbers 0 to 9, letters A to Z (uppercase) and a to z (lowercase). The underscore character (_) and dash/hyphen (-) are also considered to be members of the alphanumeric set of characters.

Blank spaces (whitespaces or SPACE characters) should mostly be avoided in alphanumeric strings, except in certain ruledef formats, such as time/date stamps.

Do not use any of the following "special" characters in an alphanumeric string except as noted below:

- **&** (ampersand)
- ' (apostrophe)
- < > (arrow brackets) [see exception below]
- ***** (asterisk) [see wildcard exception below]
- **{ }** (braces)
- **[ ]** (brackets)
- **$** (dollar sign) [see wildcard exception below]
- **!** (exclamation point) [see exception below]
- **( )** [parentheses]
- **%** (percent) [see exception below]
- **#** (pound sign) [see exception below]
- **?** (question mark)
- ' (quotation mark – single)
- " (quotation mark – double)
- **;** (semicolon)
- \ (slash – backward) [see exception below]
- / (slash – forward) [see exception below]
- ~ (tilde)
- | (vertical bar) [see exception below]

The following characters may appear in strings entered in ruledefs, APNs, license keys and other configuration/display parameters:

- < > (arrow brackets) [less than or greater than]
- ***** (asterisk) [wildcard]
- **:** (colon)
- **$** (dollar sign) [wildcard]
- **.** (dot)
- **=** (equals sign)
- **!** (exclamation point)
- **%** (percent)
- **/** (slash – forward)
- **|** (vertical bar)

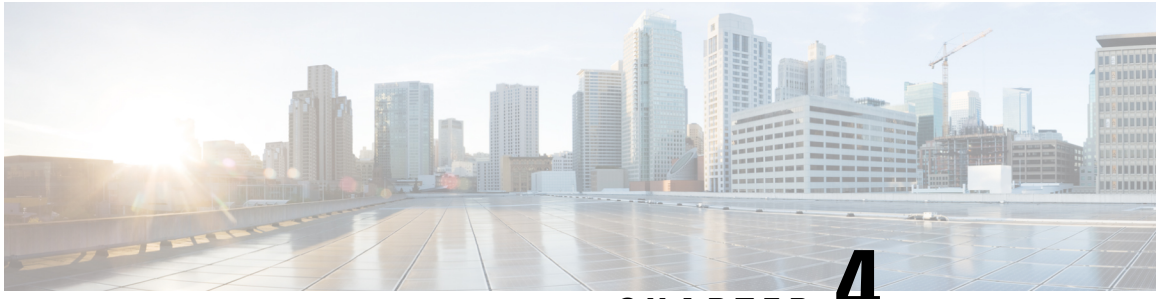
The following characters may be used to delimit the domain from the user name for global AAA functions:

- **@** (at sign)
- **-** (dash or hyphen)
- **#** (hash or pound sign)
- **%** [percent]
- \ (slash – backward) [must be entered as double slash "\\"]
- **/** (slash – forward)

Quoted Strings

If descriptive text requires the use of spaces between words, the string must be entered within double quotation marks (" "). For example:

```
interface "Rack 3 Chassis 1 port 5/2"
```



CHAPTER 4

Getting Started

Initial StarOS Configuration

Following successful installation of VPC-DI across all VMs, you must configure a set of StarOS parameters via the active Control Function (CF) VM. You then save these settings in a configuration file on the active CF that is accessed whenever a VM in the VPC-DI instance is rebooted. The standby CF and all Service Function (SF) VMs read this configuration file from the active CF.

For UGP with CUPS, you must login to the USP VM and configure a set of StarOS parameters. These settings are stored in a configuration file and sent to the CPF VNFC VMs whenever a VM is rebooted.

This chapter provides instructions for connecting to the active CF console port and creating the initial local context management configuration.

Using the StarOS CLI for Initial Configuration

The initial configuration consists of the following:

- Configuring a context-level security administrator and hostname
- Configuring the Ethernet interface on the vNIC
- Configuring the VPC-DI instance for remote CLI access via SSH

This section provides instructions for performing these tasks using the CLI.

Step 1 Log into the Console port of the active CF VM via the hypervisor.

Step 2 At the CLI prompt, enter:

```
[local]cf_host_name configure[local]cf_host_name(config)
```

Step 3 Enter the context configuration mode by entering the following command:

```
[local]cf_host_name(config) context local[local]cf_host_name(config-ctx)
```

The *local* context is the VPC-DI instance\'s management context. Contexts allow you to logically group services or interfaces. A single context can consist of multiple services and can be bound to multiple interfaces.

Step 4 Enter the following command to configure a context-level security administrator for the VPC-DI instance:

```
administrator user_name [ encrypted ] password password
| [ ecs ] [ expiry-date date_time ] [ ftp ] [ li-administration ] [ nocli ] [ noecs
 ]

  ]
```

Note You must configure a context-level security administrator during the initial configuration. After you complete the initial configuration process and end the CLI session, if you have not configured a security administrator, CLI access will be locked. See the *Context Configuration Mode Commands* chapter in the *Command Line Interface Reference* for complete information about this command.

Step 5 Enter the following command at the prompt to exit the context configuration mode:

```
[local]cf_host_name(config-ctx) exit
[local]cf_host_name(config)
```

Step 6 Enter the following command to configure a hostname by which the VPC-DI instance will be recognized on the network:

```
[local]cf_host_name(config) system hostname cf_host_name
```

cf_host_name is the name by which the VPC-DI instance will be recognized on the network. The hostname is an alphanumeric string of 1 through 63 characters that is case sensitive. The default hostname is "qvpc-di".

Step 7 Configure the network interfaces on the vNIC as follows:

a) Enter the context configuration mode by entering the following commands:

```
[local]cf_host_name(config) context local
[local]cf_host_name(config-ctx)
```

b) Enter the following command to specify a name for the interface:

```
[local]cf_host_name(config-ctx) interface interface_name
```

interface_name is the name of the interface expressed as an alphanumeric string of 1 through 79 characters that is case sensitive. The following prompt appears as StarOS enters the Ethernet Interface Configuration mode:

```
[local]cf_host_name(config-if-eth)
```

c) Configure an IP address for the interface configured in the previous step by entering the following command:

```
{ ip address | ipv6 address } ipaddress subnetmask
```

Note If you are executing this command to correct an address or subnet that was mis-configured with the Quick Setup Wizard, you must verify the default route and port binding configuration. Use *step 11* and *step 6* of this procedure. If there are issues, perform steps *7e* through *7k* to reconfigure the information.

d) Enter the following command to exit the Ethernet interface configuration mode:

```
[local]cf_host_name(config-if-eth) exit
[local]cf_host_name(config-ctx)
```

e) Configure a static route, if required, to point the VPC-DI instance to a default gateway. Entering the following command:

```
{ ip | ipv6 } route gw_address interface_name
```

f) Enter the following to exit from the context configuration mode:

```
[local]cf_host_name(config-ctx) exit
[local]cf_host_name(config)
```

g) Enter the Ethernet Port Configuration mode:

```
[local]cf_host_name(config) port ethernet slot/port
```

For VPC-DI, the *slot* corresponds to a CF or SF VM within the virtual chassis. The hypervisor assigns a unique slot number to each VM during initial configuration of the VPC-DI instance. Slots 1 and 2 are assigned to the CF VMs; slot numbers 3 through 32 are assigned to SF VMs. The CF only supports port 1. Each SF supports four vNICs numbered 1 through 4 with corresponding virtual ethernet ports numbered 10 through 14. SF port number 10 must be configured.

h) Bind the port to the interface that you created in step 7b. Binding associates the port and all of its settings to the interface. Enter the following command:

```
[local]cf_host_name(config-port-slot/port) bind interface interface_name local
[local]cf_host_name(config-port-slot/port) no shutdown
```

interface_name is the name of the interface that you configured in *step 7b*.

i) Exit the Ethernet Interface Configuration mode by entering the command:

```
[local]cf_host_name(config-port-slot/port) exit
[local]cf_host_name(config)
```

Note The management port also supports VLANs. For additional information, refer to the *VLANs* section of the *Interfaces and Ports* chapter.

Configuring System Administrative Users

This section describes some of the security features that allow security administrators to control user accounts.

Limiting the Number of Concurrent CLI Sessions

Security administrators can limit the number of concurrent interactive CLI sessions. Limiting the number of concurrent interactive sessions reduces the consumption of system-wide resources. It also prevents a user from potentially accessing sensitive user in formation which is already in use.

Most privileged accounts do not require multiple concurrent logins.

Note In 21.9 and later releases, multiple channels in a single CLI session is not supported.

Important Configuring the maximum number of sessions is recommended for all privileged accounts.

Security administrators can limit the number of concurrent interactive CLI sessions with three different ways depending on the authentication method which his used for that particular user account.

StarOS supports three login authentication methods:

- TACACS+ Server users
- Local-User users
- AAA Context users

For additional information on configuring the maximum number of sessions for TACACS+ Server users, see Operation. For additional information on configuring the maximum number of sessions for Local-User users and AAA context users, see Configuring Context-level Administrative Users.

Each authentication method must be configured separately because each of the three authentication methods can use the same user name.

Automatic Logout of CLI Sessions

Security administrators can configure an automatic logout of certain user accounts. Limiting the number of minutes that an interactive CLI session can be in use reduces the consumption of system-wide resources. It also prevents a user from potentially accessing a user account in a terminal window which is left idle. All authentication methods described in this section support both the idle session timeout technique and the absolute session timeout technique.

Most privileged accounts do not require an indefinite login timeout limit.

Important Configuring the session timeout is strongly recommended for all privileged accounts.

The idle timeout and session timeout fields in the **show tacacs summary** and **show tacacs session id** commands allow administrators to configure an automatic logout of certain accounts.

Session Timeout: allows a security administrator to specify the maximum amount of minutes that a user can be logged in to a session before the session is automatically disconnected.

Idle Timeout: allows a security administrator to specify the maximum amount of minutes that a session can remain in an idle state before the session is automatically disconnected.

Important The session timeout and idle timeout fields are not exclusive. If both are specified, then the idle timeout should always be lower than the session timeout since a lower session timeout will always be reached first.

For additional information on configuring the maximum number of minutes that an interactive CLI session can be in use, see the **idle-sessions threshold** command and the **clear tacacs sessions** CLI command in the

CLI Reference and the **show tacacs summary** and **show tacacs session id** in the *Statistics and Counter Reference*.

Configuring the System for Remote Access

Configure the system for remote access. An administrative user may access the instance from a remote location over the management network:

- Telnet
- Secure Shell (SSH)
- File Transfer Protocol (FTP) (secured or unsecured)
- Trivial File Transfer Protocol (TFTP)

Note If there are two simultaneous telnet sessions, and one administrator deletes the context into which the other administrator is logged, the administrator in the deleted context will not be automatically kicked into the *local* context. Although the deleted context will still appear in the CLI prompt, context specific commands will generate errors.

Note For maximum security, use SSH v2.

Step 1 Enter the context configuration mode by entering the following command:

```
[local]cf_host_name(config) context local
[local]cf_host_name(config-ctx)
```

Step 2 Configure the system to allow Telnet access, if desired:

```
[local]cf_host_name(config-ctx) server telnetd
```

Step 3 Configure the system to allow SSH access, if desired:

```
[local]cf_host_name(config-ctx) ssh generate key [ type v2-rsa ]
```

Note **v2-rsa** is the recommended key type.

Note In Release 4.0 and higher, the **v1-rsa** keyword has been removed and the **v2-dsa** keyword has been concealed within the Context Configuration mode **ssh generate** CLI command. A keyword that was supported in a previous release may be concealed in subsequent releases. The system continues to parse concealed keywords in existing scripts and configuration files created in a previous release. But the concealed keyword no longer appears in the command syntax for use in new scripts or configuration files. Entering a question mark (?) will not display a concealed keyword as part of the Help text. A removed keyword generates an error message when parsed.

```
[local]cf_host_name(config-ctx) server sshd
[local]cf_host_name(config-sshd) subsystem sftp
[local]cf_host_name(config-sshd) exit
```

Step 4 Configure the system to allow FTP access, if desired, by entering the following command:

```
[local]cf_host_name(config-ctx) server ftpd
```

Step 5 Exit the configuration mode by entering the following command:

```
[local]cf_host_name(config-ctx) end
[local]cf_host_name
```

Step 6 Verify the configuration by entering the following command:

```
[local]cf_host_name show configuration
```

The CLI output should be similar to the sample output:

```
context local
   interface interface_name
      ip address ipaddress subnetmask
      exit
   subscriber default
      exit
   administrator admin_name password admin_password
   server telnetd
   server ftpd
   ssh generate key
   server sshd
   subsystem sftp
   exit
port ethernet 1/1
   bind interface interface_name local
   exit
port ethernet 1/1
   no shutdown
   exit
 snmp engine-id local 800007e580ed826c191ded2d3d
end
```

Step 7 Verify the configuration of the IP routes by entering the following command:

```
[local]cf_host_name show ip route
```

The CLI output should be similar to the sample output:

```
"*" indicates the Best or Used route.
 Destination         Nexthop         Protocol  Prec Cost Interface
*0.0.0.0/0           ipaddress      static     1    0    vnic1
*network            0.0.0.0         connected  0    0    vnic1
```

Step 8 Verify the interface binding by entering the following command:

```
[local]cf_host_name show ip interface name interface_name
```

interface_name is the name of the interface that was configured in *step 7b*.The CLI output should be similar to the sample output:

```
Intf Name:       vnic1

Description:
IP State:        UP (Bound to 1/1 untagged, ifIndex 83951617)
IP Address:      ipaddress          Subnet Mask:     subnetmask
Bcast Address:   bcastaddress       MTU:             1500
Resoln Type:     ARP                ARP timeout:     3600 secsL3 monitor LC-port
switchover: DiasabledNumber of Secondary Addresses:  0
```

Step 9 Save your configuration as described in the *Verifying and Saving Your Configuration* chapter.

Configuring SSH Options

SSHv2 RSA is the only version of SSH supported under StarOS. Keywords previously supported for SSHv1 RSA and SSHv2 DSA have been removed from or concealed within the StarOS CLI.

Important A keyword that was supported in a previous release may be concealed in subsequent releases. StarOS continues to parse concealed keywords in existing scripts and configuration files created in a previous release. But the concealed keyword no longer appears in the command syntax for use in new scripts or configuration files. Entering a question mark (?) will not display a concealed keyword as part of the Help text. Removed keywords generate an error message when parsed.

Version 1 of the SSH protocol is now obsolete due to security vulnerabilities. The **v1-rsa** keyword has been removed for the Context Configuration mode **ssh** command. Running a script or configuration that uses the SSHv1-RSA key returns an error message and generates an event log. The output of the error message is shown below:

```
CLI print failure Failure: SSH V1 contains multiple structural vulnerabilities and is no
longer considered secure. Therefore we don't support v1-rsa SSH key any longer, please
generate a new v2-rsa key to replace this old one.
```

If the system boots from a configuration that contains the **v1-rsa** key, you can expect a boot failure when logging in through SSH. The workaround is to log in via the Console port, re-generate a new ssh v2-rsa key, and configure server sshd. It will then be possible to log in via ssh.

The **v2-dsa keyword** is now concealed for the Context Configuration mode **ssh** command

The **v1-rsa** keyword has been removed from the Exec mode **show ssh key** CLI command.

SSH Host Keys

SSH key-based authentication uses two keys, one "public" key that anyone is allowed to see, and another "private" key that only the owner is allowed to see. You create a key pair, securely store the private key on the device you want to log in from, and store the public key on the system (VPC-DI) that you wish to log into.

SSH host keys are generated within a specified StarOS context. The context is associated with a user interface.

You set or remove an administrative user name having authorized keys for access to the sshd server associated with context.

Setting SSH Key Size

The Global Configuration mode **ssh key-size** CLI command configures the key size for SSH key generation for all contexts (RSA host key only).

Step 1 Enter the Global Configuration mode.

```
[local]host_name# configure
[local]host_name(config)#
```

Step 2 Specify the bit size for SSH keys.

```
[local]host_name(config)# ssh key-size { 2048 | 3072 | 4096 | 5120 | 6144 | 7168 |
9216 }
```

The default bit size for SSH keys is 2048 bits.

Configuring SSH Key Generation Wait Time

SSH keys can only be generated after a configurable time interval has expired since the last key generation. The **ssh key-gen wait-time** command specifies this wait time in seconds. The default interval is 300 seconds (5 minutes).

Step 1 Enter the context configuration mode.

```
[local]host_name(config)# context context_name
[local]host_name(config-ctx)#
```

Step 2 Specify the wait time interval.

```
[local]host_name(config-ctx)# ssh key-gen wait-time seconds
[local]host_name(config-ctx)#
```

Notes:

- *seconds* is specified as an integer from 0 through 86400. Default = 300

Specifying SSH Encryption Ciphers

The SSH Configuration mode **ciphers** CLI command configures the cipher priority list in sshd for SSH symmetric encryption. It changes the cipher options for that context.

Step 1 Enter the SSH Configuration mode.

```
[local]host_name(config-ctx)# server sshd
```

Step 2 Specify the desired encryption algorithms.

```
[local]host_name(config-sshd)# ciphers algorithms
```

Notes:

- *algorithms* is a string of 1 through 511 alphanumeric characters that specifies the algorithm(s) to be used as a single string of comma-separated variables (no spaces) in priority order (left to right) from those shown below:
  - **blowfish-cbc** – symmetric-key block cipher, Cipher Block Chaining, (CBC)
  - **3des-cbc** – Triple Data Encryption Standard, CBC
  - **aes128-cbc** – Advanced Encryption Standard (AES), 128-bit key size, CBC

- **aes128-ctr** – AES, 128-bit key size, Counter-mode encryption (CTR)
- **aes192-ctr** – AES, 192-bit key size, CTR
- **aes256-ctr** – AES, 256-bit key size, CTR
- **aes128-gcm@openssh.com** – AES, 128-bit key size, Galois Counter Mode [GCM], OpenSSH
- **aes256-gcm@openssh.com** – AES, 256-bit key size, GCM, OpenSSH
- **chacha20-poly1305@openssh.com** – ChaCha20 symmetric cipher, Poly1305 cryptographic Message Authentication Code [MAC], OpenSSH

The default string for *algorithms* in a Normal build is:

```
blowfish-cbc,3des-cbc,aes128-cbc,aes128-ctr,aes192-ctr,aes256-ctr,aes128-gcm@openssh.com,aes256-gcm@openssh.com,
chacha20-poly1305@openssh.com
```

The default string for *algorithms* in a Trusted build is:

```
aes256-ctr,aes192-ctr,aes128-ctr
```

Step 3 Exit the SSH Configuration mode.

```
[local]host_name(config-sshd)# end
[local]host_name#
```

MAC Algorithm Configuration

Feature Summary and Revision History

Summary Data

Applicable Product(s) or Functional Area	All
Applicable Platform(s)	• ASR 5500 • VPC-DI • VPC-SI
Feature Default	Disabled - Configuration required
Related Changes in This Release	Not applicable
Related Documentation	• *ASR 5500 System Administration Guide* • *Command Line Interface Reference* • *VPC-DI System Administration Guide* • *VPC-SI System Administration Guide*

Revision History

Important Revision history details are not provided for features introduced before releases 21.2 and N5.1.

Revision Details	Release
First introduced.	21.13

Feature Description

The MAC Algorithm Configuration feature allows to configure or change the priority of MAC algorithms of internal SSHD servers.

A new CLI **MACs** CLI command is introduced in SSH Configuration Mode in support of this feature.

Configuring MAC Algorithms

This section describes how to configure the MAC alogrithims.

Use the following configuration to specify the priority of the MAC algorithms.

```
configure
  context context_name
    server sshd
      macs algorithms
      end

default macs
```

NOTES:

- *algorithms*: Refers to a string of 1 through 511 alphanumeric characters that specifies the algorithms to be used as a single string of comma-separated variables (no spaces) in priority order (left to right) from those listed as follows:
  - HMAC = hash-based message authentication code
  - SHA2 = Secure Hash Algorithm 2
  - SHA1 = Secure Hash Algorithm 1
  - ETM = Encrypt-Then-MAC
  - UMAC = message authentication code based on universal hashing
- The help string and list of algorithms in a Normal build are:

  `hmac-sha2-512-etm@openssh.com,hmac-sha2-256-etm@openssh.com,hmac-sha1-etm@openssh.com,hmac-sha2-512,hmac-sha2-256,hmac-sha1,umac-128-etm@openssh.com,umac-128@openssh.com,umac-64-etm@openssh.com,umac-64@openssh.com`
- The help string and list of algorithms in a Trusted build are:

  `hmac-sha2-512-etm@openssh.com,hmac-sha2-256-etm@openssh.com,hmac-sha1-etm@openssh.com,hmac-sha2-512,hmac-sha2-256,hmac-sha1`
- The default value string is:

```
hmac-sha2-512-etm@openssh.com,hmac-sha2-256-etm@openssh.com,hmac-sha1-etm@openssh.com,hmac-sha2-512,
hmac-sha2-256,hmac-sha1
```

Specifying MAC Algorithms

Use the following CLI commands to configure the priority of MAC algorithms. This command is configured in in SSH Configuration Mode.

```
configure
  context context_name
    server sshd
      macs algorithms
      end

default macs
```

NOTES:

- *algorithms*: Refers to a string of 1 through 511 alphanumeric characters that specifies the algorithms to be used as a single string of comma-separated variables (no spaces) in priority order (left to right) from those listed as follows:
  - HMAC = hash-based message authentication code
  - SHA2 = Secure Hash Algorithm 2
  - SHA1 = Secure Hash Algorithm 1
  - ETM = Encrypt-Then-MAC
  - UMAC = message authentication code based on universal hashing
- The help string and list of algorithms in a Normal build are:

  ```
  hmac-sha2-512-etm@openssh.com,hmac-sha2-256-etm@openssh.com,hmac-sha1-etm@openssh.com,hmac-sha2-512,
  hmac-sha2-256,hmac-sha1,umac-128-etm@openssh.com,umac-128@openssh.com,umac-64-etm@openssh.com,umac-64@openssh.com
  ```

- The help string and list of algorithms in a Trusted build are:

  ```
  hmac-sha2-512-etm@openssh.com,hmac-sha2-256-etm@openssh.com,hmac-sha1-etm@openssh.com,hmac-sha2-512,
  hmac-sha2-256,hmac-sha1
  ```

- The default value string is:

  ```
  hmac-sha2-512-etm@openssh.com,hmac-sha2-256-etm@openssh.com,hmac-sha1-etm@openssh.com,hmac-sha2-512,
  hmac-sha2-256,hmac-sha1
  ```

Generating SSH Keys

The **ssh generate** command generates a public/private key pair which is to be used by the SSH server. The **v1-rsa** keyword has been removed from and the **v2-dsa** keyword concealed within the **ssh generate** CLI command. The only keyword available for generating SSH keys is **v2-rsa**.

Important The generated key pair remains in use until the command is issued again.

Step 1 Enter the context configuration mode:

```
[local]host_name(config)# context context_name
[local]host_name(config-ctx)#
```

Step 2 Generate an SSH key pair.

```
[local]host_name(config-ctx)# ssh generate key type v2-rsa
[local]host_name(config-ctx)#
```

Setting SSH Key Pair

The **ssh key** command sets the public/private key pair to be used by the system. The **v2-dsa** keyword is concealed in the **ssh key** command.

Specify the SSH key pair parameters.

```
[local]host_name(config-ctx)# ssh key data length octets type v2-rsa
```

Notes:

- *data* is the encrypted key expressed as an alphanumeric string of 1 through 1023 characters
- **length** *octets* is the length of the encrypted key in octets expressed as an integer from 0 through 65535
- **type** specifies the key type; **v2-rsa** is the only supported type.

Important *For releases prior to 20.0*, StarOS supports a maximum of 64 configurable authorized SSH keys. *For release 20.0 and higher*, StarOS supports a maximum of 200 configurable authorized SSH keys.

Authorized SSH User Access

You must authorize users to access a StarOS context from a specific host with an SSH authentication-key pair.

Authorizing SSH User Access

The SSH Configuration mode **authorized-key** command grants user access to a context from a specified host.

Step 1 Go to the SSH Configuration mode.

```
[local]host_name(config-ctx)# server sshd
[local]host_name(config-sshd)#
```

Step 2 Specify administrative user access via the **authorized-key** command.

```
[local]host_name(config-sshd)# authorized-key username user_name host host_ip [ type {
v2-dsa | v2-rsa } ]
```

Notes:

- **username** *user_name* specifies an existing StarOS administrator user name as having authorized keys for access to the sshd server. The *user_name* is expressed as an alphanumeric string of 1 through 255 characters. User names should have been previously created via the Context Configuration mode **administrator** command using the **nopassword** option to prevent bypassing of the sshd keys. Refer to the *System Settings* chapter for additional information on creating administrators.
- **host** *host_ip* specifies the IP address of an SSH host having the authorization keys for this username. The IP address must be in IPv4 dotted-decimal or IPv6 colon-separated-hexadecimal notation.
- **type** specifies the key type; **v2-rsa** is the only supported type.

SSH User Login Restrictions

An administrator can restrict SSH access to the StarOS CLI to a "white list" of allowed users. Access to a service may be restricted to only those users having a legitimate need. Only explicitly allowed users will be able connect to a host via SSH. The user name may optionally include a specific source IP address.

The AllowUsers list consists of user name patterns, separated by space. If the pattern takes the form 'USER' then login is restricted for that user. If pattern is in the format 'USER@IP_ADDRESS' then USER and IP address are separately checked, restricting logins to those users from the specified IP address.

The default is to allow unrestricted access by any user.

Creating an Allowed Users List

The **allowusers add** command allows an administrator to create a list of users who may log into the StarOS CLI.

Step 1 Enter the context configuration mode.

```
[local]host_name(config)# context context_name
[local]host_name(config-ctx)#
```

Step 2 Go to the SSH Configuration mode.

```
[local]host_name(config-ctx)# server sshd
```

Step 3 Configure the SSH user list.

```
[local]host_name(config-sshd)# allowusers add user_list
```

user_list specifies a list of user name patterns, separated by spaces, as an alphanumeric string of 1 through 999 characters. If the pattern takes the form 'USER' then login is restricted for that user.

If the pattern is in the format 'USER@IP_ADDRESS' then user name and IP address are separately checked, restricting logins to those users from that particular IP address.

If the pattern is in the format 'USER@<context>@IP_ADDRESS' then user name, StarOS context and IP address are separately checked, restricting logins to those users associated with the specific context from that particular IP address.

The following limits apply to the *user_list*:

- The maximum length of this string is 3000 bytes including spaces.

- The maximum number of AllowUsers, which is counted by spaces, is 256, which is consistent with the limit from OpenSSH.

Important If you exceed either of the above limits, an error message is displayed. The message prompts you to use a regular expression pattern to shorten the string, or remove all the allowusers with **no allowusers add** or **default allowusers add** and re-configure.

For additional information, see the *SSH Configuration Mode Commands* chapter in the *Command Line Interface Reference*.

Step 4 Exit the SSH Configuration mode.

```
[local]host_name(config-sshd)# end
[local]host_name#
```

SSH User Login Authentication

StarOS authenticates SSH user login attempts via authorized-key/user-account pairings for the following scenarios:

- User tries to login with local context username through local context (VPN) interface with authorized-key configured on local context.
- User tries to login with non-local context username through non-local context interface with authorized-key configured on non-local context.
- User tries to login with local context username through non-local context interface with authorized-key configured on local context.
- User tries to login with non-local context username through local context interface with authorized-key configured on non-local context.

A failure to authenticate based on the current system configuration prevents the login and generates an error message.

StarOS does not permit users with different user IDs but having the same public SSH key to login to an unauthorized context. Authentication of the user takes into account the authorized-key/user-account pairing.

Important For StarOS release 21.0 onwards, a user cannot access the /flash directory if the user logs in from a non-local context.

Secure Session Logout

When StarOS is disconnected from an SSH client, the default behavior has sshd terminate the CLI or SFTP session in about 45 seconds (using default parameters). Two SSH Configuration mode CLI commands allow you to disable or modify this default sshd disconnect behavior.

Important For higher security, Cisco recommends at least a client-alive-countmax of 2 and client-alive-interval of 5. Smaller session logout values may lead to occasional ssh session logouts. Adjust values to balance security and user friendliness.

The **client-active-countmax** command sets the number of client-alive messages which may be sent without sshd receiving any messages back from the SSH client (default =3). If this threshold is reached while the client-alive messages are being sent, sshd disconnects the SSH client thus terminating the session.

The **client-alive-interval** command sets a timeout interval in seconds (default = 15) after which if no data has been received from the SSH client, sshd sends a message through the encrypted channel to request a response from the client. The number of times that the message is sent is determined by the client-alive-countmax parameter. The approximate amount of time before sshd disconnects an SSH client disconnect = client-alive-countmax X client-alive-interval.

The client-alive mechanism is valuable when the client or server depend on knowing when a connection has become inactive.

Important The client-alive messages are sent through the encrypted channel and, therefore, are not spoofable.

Important These parameter apply to SSH protocol version 2 only.

Changing Default sshd Secure Session Logout Parameters

The following command sequence modifies the default settings for the ClientAliveCountmax (default = 3) and ClientAliveInterval (default = 15 seconds) parameters.

Step 1 Enter the context configuration mode.

```
[local]host_name# configure
```

Step 2 Go to the SSH Configuration mode.

```
[local]host_name(config)# context context_name
```

Step 3 Set the ClientAliveCountmax parameter to 2.

```
[local]host_name(config-sshd)# client-alive-countmax 2
```

Step 4 Set the ClientAliveInterval parameter to 5 seconds.

```
[local]host_name(config-sshd)# client-alive-interval 5
```

Step 5 Exit the SSH Configuration mode.

```
[local]host_name(config-sshd)# end
[local]host_name#
```

SSH Client Login to External Servers

StarOS supports public key authentication for SSH/SFTP access from the StarOS gateway to external servers. You configure this feature by generating SSH client key pairs and pushing the client public key to external servers

Note By default StarOS only supports username-password authentication to external servers.

Setting SSH Client Ciphers

The SSH Client Configuration mode **ciphers** CLI command configures the cipher priority list when logging into an external server.

Step 1 Enter the SSH Client Configuration mode.

```
[local]host_name(config)# client ssh
```

Step 2 Specify the desired encryption algorithms.

```
[local]host_name(config-ssh)# ciphers algorithms
```

Notes:

- *algorithms* is a string of 1 through 511 alphanumeric characters that specifies the algorithm(s) to be used as a single string of comma-separated variables (no spaces) in priority order (left to right) from those shown below:
  - **blowfish-cbc** – symmetric-key block cipher, Cipher Block Chaining, (CBC)
  - **3des-cbc** – Triple Data Encryption Standard, CBC
  - **aes128-cbc** – Advanced Encryption Standard (AES), 128-bit key size, CBC
  - **aes128-ctr** – AES, 128-bit key size, Counter-mode encryption (CTR)
  - **aes192-ctr** – AES, 192-bit key size, CTR
  - **aes256-ctr** – AES, 256-bit key size, CTR
  - **aes128-gcm@openssh.com** – AES, 128-bit key size, Galois Counter Mode [GCM], OpenSSH
  - **aes256-gcm@openssh.com** – AES, 256-bit key size, GCM, OpenSSH
  - **chacha20-poly1305@openssh.com** – ChaCha20 symmetric cipher, Poly1305 cryptographic Message Authentication Code [MAC], OpenSSH

The default string for *algorithms* in a Normal build is:

```
aes256-ctr,aes192-ctr,aes128-ctr,aes256-gcm@openssh.com,aes128-gcm@openssh.com,chacha20-poly1305@openssh.com,
blowfish-cbc,3des-cbc,aes128-cbc
```

The default string for *algorithms* in a Trusted build is:

```
aes256-ctr,aes192-ctr,aes128-ctr
```

Step 3 Exit the SSH Client Configuration mode.

```
[local]host_name(config-ssh)# end
[local]host_name#
```

Setting Preferred Authentication Methods

The SSH Client Configuration mode **preferredauthentications** CLI command configures the preferred methods of authentication.

Step 1 Enter the SSH Client Configuration mode.

```
[local]host_name(config)# client ssh
```

Step 2 Specify the preferred authentication methods.

```
[local]host_name(config-ssh)# preferredauthentications methods
```

Notes:

- *methods* – specifies the preferred methods of authentication to be used as a single string of comma-separated variables (no spaces) in priority order (left to right) from those shown below:
  - **publickey** – authentication via SSH v2-RSA protocol.
  - **keyboard-interactive** – request for an arbitrary number of pieces of information. For each piece of information the server sends the label of the prompt.
  - **password** – simple request for a single password
- **default** – resets the value of methods to: publickey,password

Step 3 Exit the SSH Client Configuration mode.

```
[local]host_name(config-ssh)# exit
[local]host_name(config)#
```

Generating SSH Client Key Pair

You use commands in the SSH Client Configuration mode to specify a private key and generate the SSH client key pair.

Step 1 Enter the SSH client configuration mode.

```
[local]host_name(config)# client ssh
[local]host_name(config-ssh)#
```

Step 2 Enter SSH private key information and key type.

```
[local]host_name(config-ssh)# ssh key private_key_string length key_length [ type v2-rsa ]
[local]host_name(config-ssh)#
```

key *private_key_string* specifies a private key value as an alphanumeric string of 1 through 4499 characters.

length *key_length* specifies the length of the key in bytes as an integer from 0 through 65535.

type v2-rsa specifies the SSH client key type. The only supported SSH client key type is **v2-rsa**.

Step 3 Generate SSH client key pair.

```
[local]host_name(config-ssh)# ssh generate key [ type v2-rsa ]
[local]host_name(config-ssh)#
```

type v2-rsa specifies the SSH client key type. The only supported SSH client key type is **v2-rsa**.

Step 4 Verify that the SSH client key has been generated.

```
[local]host_name(config-ssh)# do show ssh client key
```

Step 5 Exit the SSH Client Configuration mode.

```
[local]host_name(config-ssh)# exit
[local]host_name(config)#
```

Pushing an SSH Client Public Key to an External Server

You must push the SSH client public key to an external server to support SSH/SFTP access to that server.

Step 1 From the Exec mode run the **push ssh-key** command.

```
[local]host_name# push ssh-key { host_name | host_ip_address } user username [ context
context_name ]
[local]host_name#
```

host_name specifies the remote server using its logical host name which must be resolved via DNS lookup. It is expressed as an alphanumeric string of 1 to 127 characters.

host_ip_address is expressed in IPv4 dotted-decimal or IPv6 colon-separated-hexadecimal notation.

user *username* specifies a valid username on the external server as an alphanumeric string of 1 to 79 characters.

context *context_name* specifies a valid context name. The context name is optional. If it is not provided the current context is used for processing.

Step 2 Repeat Step 1 to support SSH/SFTP access on other external servers.

Step 3 Test SSH client login to an external server.

```
local]host_name# ssh { hostname | ip_address } user username port port_number
```

Enabling NETCONF

An SSH key is a requirement before NETCONF protocol and the ConfD engine can be enabled in support of Cisco Network Service Orchestrator (NSO).

Refer to the *NETCONF and ConfD* appendix in this guide for detailed information on how to enable NETCONF.

Configuring the Management Interface with a Second IP Address

If necessary, you can configure a second IP address on the vNIC management interface.

Procedure

	Command or Action	Purpose
Step 1	Enter the configuration mode by entering the following command at the prompt:	`[local]host_name` **`configure`** `[local]host_name(config)`
Step 2	Enter the following to enter the context configuration mode:	`[local]host_name(config)` **`context local`** `[local]host-name(config-ctx)`
Step 3	Enter the interface slot number and port number via the following command:	`[local]host_name(config-ctx) 1/1` `[local]host_name(config-if-eth)`
Step 4	Enter the secondary IP address and subnet mask by entering the following command:	`[local]host_name(config-if-eth)` **`{ ip | ipv }`** **`address`** *`ipaddress subnet_mask`* `secondary`
Step 5	Exit the configuration mode by entering the following command:	`[local]host_name(config-if-eth)` **`end`**

Step 6	Confirm the interface ip addresses by entering the following command:	`[local]host_name` **`show config context local`**
		The CLI output should look similar to this example:
		`config`
		`  context local`
		`  interface interface_name`
		`  ip address ipaddress subnetmask`
		`  ip address ipaddress subnetmask secondary`
		`  exit`

Step 7	Continue with Verifying and Saving Your Interface and Port Configuration, on page 97.	

Upgrade and Migration of Open SSH to Cisco SSH

Feature Summary and Revision History

Summary Data

Applicable Product(s) or Functional Area	All

Applicable Platform(s)	• ASR 5500 • VPC-DI • VPC-SI
Feature Default	Enabled - Always-on
Related Changes in This Release	Not applicable
Related Documentation	• *ASR 5500 System Administration Guide* • *Command Line Interface Reference* • *VPC-DI System Administration Guide* • *VPC-SI System Administration Guide*

Revision History

Important Revision history details are not provided for features introduced before releases 21.2 and N5.1.

Revision Details	Release
With this release, the algorithm values of Ciphers and MACs are modified based on the upgrade and migration of OpenSSH to CiscoSSH.	21.16
First introduced.	Pre 21.2

Feature Changes

As a security measure for Cisco ASR 5500 and VPC products, the Ciphers and MACs algorithm values are modified to support the upgrade and migration of the Open SSH to Cisco SSH versions.

Previous Behavior: In releases earlier to 21.16, the **default** algorithm values of the **cipher** and **macs** commands were as follows:

- **Cipher**

  Release 20.x to 21.15 (Normal build only)

  Resets the value of *algorithm* in a Normal build to:

  blowfish-cbc,3des-cbc,aes128-cbc,aes128-ctr,aes192-ctr,aes256-ctr,aes128-gcm@openssh.com,aes256-gcm@openssh.com,chacha20-poly1305@openssh.com

- **MACs**

  Release 20.x to 21.15 (Trusted build only)

  Resets the value of *algorithm* in a Trusted build to:

  hmac-sha2-512-etm@openssh.com,hmac-sha2-256-etm@openssh.com,hmac-sha1-etm@openssh.com,hmac-sha2-512,hmac-sha2-256,hmac-sha1

- **KEX Algorithms**

  Release 20.x to 21.15

  Available Algorithms in Normal and Trusted Builds:

  `diffie-hellman-group1-sha1,diffie-hellman-group14-sha1`

New Behavior: In this release, the **default** algorithm values of the **cipher** and **macs** commands are as follows:

- **Cipher**

  Release 21.16 onwards: Post OpenSSH to CiscoSSH Upgrade and Migration

  Default Algorithms in a Normal Build:

  `aes256-ctr,aes192-ctr,aes128-ctr,aes256-gcm@openssh.com,aes128-gcm@openssh.com,chacha20-poly1305@openssh.com`

  Available Algorithms in a Normal Build:

  `aes256-ctr,aes192-ctr,aes128-ctr,aes256-gcm@openssh.com,aes128-gcm@openssh.com,chacha20-poly1305@openssh.com,aes128-cbc`

  Default and Available Algorithms in Trusted Builds:

  `aes256-ctr,aes192-ctr,aes128-ctr`

  Note There is no change in the default and configurable Ciphers for Trusted builds.

- **MACs**

  Release 21.16 onwards: Post OpenSSH to CiscoSSH Upgrade and Migration

  Default and Available Algorithms in Normal Builds:

  `hmac-sha2-512-etm@openssh.com,hmac-sha2-256-etm@openssh.com,hmac-sha1-etm@openssh.com,hmac-sha2-512,hmac-sha2-256,hmac-sha1`

  Default Algorithms in Trusted Builds:

  `hmac-sha2-512,hmac-sha2-256,hmac-sha1`

  Available Algorithms in Trusted Builds:

  `hmac-sha2-512,hmac-sha2-256,hmac-sha1`

  Note hmac-sha2-512-etm@openssh.com,hmac-sha2-256-etm@openssh.com,hmac-sha1-etm@openssh.com are removed from the Trusted builds.

- **KEX Algorithms**

  Release 21.16 onwards: Post OpenSSH to CiscoSSH Upgrade and Migration

  Available Algorithms in Normal and Trusted Builds:

  `diffie-hellman-group14-sha1`

Note KEX algorithms are not configurable in StarOS. Therefore, there are no CLI changes.

VM Hardware Verification

To prevent resource allocation issues, it is important that all VMs used for in the system have the same size CPU and the same size memory. To balance performance across all interfaces, make sure that the service ports and DI ports have the same throughput capabilities.

To verify the hardware configuration for all cards or a specific card, use the **show cloud hardware** [*card_number*} command. Sample output from this command on card 1 (CF) is shown here:

```
[local]s1# show cloud hardware 1

Card 1:
   CPU Nodes                : 1
   CPU Cores/Threads        : 8
   Memory                   : 16384M (qvpc-di-medium)
   Hugepage size            : 2048kB
   cpeth0                   :
     Driver                 : virtio_net
   loeth0                   :
     Driver                 : virtio_net
```

Sample output from this command on card 3 (SF) is shown here:

```
[local]s1# show cloud hardware 1

Card 3:
   CPU Nodes                : 1
   CPU Cores/Threads        : 8
   Memory                   : 16384M (qvpc-di-medium)
   Hugepage size            : 2048kB
   cpeth0                   :
     Driver                 : vmxnet3
   port3_10                 :
     Driver                 : vmxnet3
   port3_11                 :
     Driver                 : vmxnet3
```

To display the optimum configuration of the underlying VM hardware, use the **show hardware optimum**. To compare your current VM configuration to the optimum configuration, use the **show cloud hardware test** command. Any parameters not set to the optimum are flagged with an asterisk, as shown in this sample output. In this example, the CPU cores/threads and memory are not configured optimally.

```
[local]s1# show cloud hardware test 1

Card 1:
   CPU Nodes                : 1
*  CPU Cores/Threads        : 8                                Optimum value is 4
*  Memory                   : 8192M (qvpc-di-medium)           Optimum value is 16384
   Hugepage size            : 2048kB
   cpeth0                   :
     Driver                 : virtio_net
   loeth0                   :
```

```
     Driver                : virtio_net
```

To display the configuration file on the config disk or local flash, use the **show cloud configuration** *card_number* command. The location parameter file on flash memory is defined during the installation. And the config disk is usually created by the orchestrator and then attached to the card. Sample output from this command is shown here for card 1:

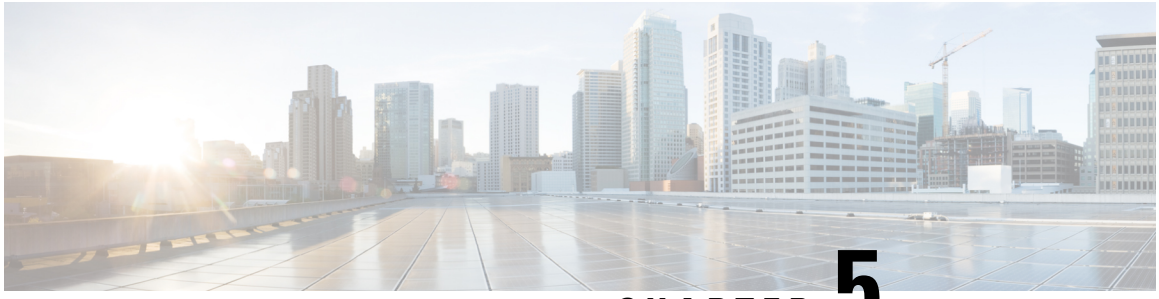
```
[local]s1# show cloud configuration 1

Card 1:
  Config Disk Params:
-------------------------
    No config disk available

  Local Params:
-------------------------
CARDSLOT=1
CARDTYPE=0x40010100
CPUID=0
```

To display the IFTASK configuration for all cards or a specific card, use the **show cloud hardware iftask** command. By default, the cores are configured to be used for both PMD and VNPU. Sample output from this command on card 4 is shown here:

```
[local]mySystem# show cloud hardware iftask 4
Card 4:
  Total number of cores on VM:      24
  Number of cores for PMD only:     0
  Number of cores for VNPU only:    0
  Number of cores for PMD and VNPU: 3
  Number of cores for MCDMA:        4
  Hugepage size:           2048 kB
  Total hugepages:     16480256 kB
  NPUSHM hugepages:           0 kB
  CPU flags: avx sse sse2 ssse3 sse4_1 sse4_2
  Poll CPU's: 1 2 3 4 5 6 7
  KNI reschedule interval: 5 us
```



CHAPTER 5

System Settings

This chapter provides instructions for configuring the following StarOS options.

It is assumed that the procedures to initially configure the system as described in *Getting Started* have been completed.

Important For VPC-DI deployments, you must log into the active Control Function (CF) VM to complete the procedures in this chapter.

Important The commands used in the configuration examples in this section are the most likely-used commands and/or keyword options. In many cases, other optional commands and/or keyword options are available. Refer to the *Command Line Interface Reference* for complete information.

Verifying and Saving Your Interface and Port Configuration

Verify that your interface configuration settings are correct by entering the following StarOS CLI command:

```
show ip interface
```

The output from this command should be similar to that shown below. In this example an interface named *mangement1* was configured in the local context.

```
Intf Name:       LOCAL1
Intf Type:       Broadcast
Description:     management1
VRF:             None
IP State:        UP (Bound to 1/1 untagged, ifIndex 16842753)
IP Address:      192.168.100.3        Subnet Mask:     255.255.255.0
Bcast Address:   192.168.100.255      MTU:             1500
Resoln Type:     ARP                  ARP timeout:     60 secs
L3 monitor LC-port switchover: Disabled
Number of Secondary Addresses: 0
```

Verify that the port configuration settings are correct by entering the following command:

show configuration port *slot/port*

For VPC-DI, the *slot* corresponds to a CF or SF VM. The hypervisor assigns a unique slot number to each VM during initial configuration of the VPC-DI instance. Slots 1 and 2 are assigned to the CF VMs slot numbers 3 through 32 are assigned to SF VMs. Each SF supports four vNICs numbered 1 through 12 with corresponding virtual ethernet ports numbered 10 through 21. SF port number 10 must be configured.

This previous command produces an output similar to the one shown below. It displays the configuration of port 1 in slot 1 (active CF).

```
config
  port ethernet 1/1
    no shutdown
    bind interface LOCAL1 local
```

Save your configuration as described in the *Verifying and Saving Your Configuration* chapter.

Configuring System Timing

The system is equipped with a clock that supplies the timestamp for statistical counters, accounting records, logging, and event notification. After the initial configuration of the system clock, you can configure the system to communicate with one or more Network Time Protocol (NTP) server(s) to ensure that the clock is always accurate.

All VPC instances must be aligned with the timing standard used by the IaaS datacenter in which the hosts are located.

In addition to configuring the timing source, you must configure the system's time zone.

Setting the System Clock and Time Zone

Use the following command example to configure the system clock and time zone:

clock set *date:time*
configure
  clock timezone *timezone* [**local**]
  end

Notes:

- Enter the date and time in the format YYYY:MM:DD:HH:mm or YYYY:MM:DD:HH:mm:ss.

- Refer to the online Help for the **clock timezone** command for a complete list of supported time zones.
- The optional **local** keyword indicates that the time zone specified is the local timezone.
- Daylight Savings Time is automatically adjusted for time zones supporting it.

Save your configuration as described in the *Verifying and Saving Your Configuration* chapter.

Verifying and Saving Your Clock and Time Zone Configuration

Enter the following command to verify that you configured the time and time zone correctly:

```
show clock
```

The output displays the date, time, and time zone that you configured.

Configuring Network Time Protocol Support

This section provides information and instructions for configuring the system to enable the use of the Network Time Protocol (NTP).

Important Configure the system clock and time zone prior to implementing NTP support. This greatly reduces the time period that must be corrected by the NTP server.

Note NTP should also be configured on all commercial off-the-shelf (COTS) servers running VPC VMs. The StarOS NTP configuration should match that of the COTS servers.

Many of the services offered by the StarOS require accurate timekeeping derived through NTP. If the time reference(s) used by StarOS are not accurate, the services may be unreliable. For this reason it should be assumed that normal system operation requires that NTP be configured.

The system uses NTP to synchronize its internal clock to external time sources (typically GPS NTP sources, or other Stratum 2 or 3 servers, switches or routers).

The system should have:

- NTP enabled.
- NTP configured for use in the local context only. Use of other contexts (which can be specified in the enable configurable) will cause issues.
- NTP configured for at least three external NTP servers. With three or more servers, outlyers and broken or misconfigured servers can be detected and excluded. Generally, the more servers the better (within reason).

Important Do not configure any external NTP servers using the **prefer** keyword. The NTP clock selection algorithms already have the built-in ability to pick the best server. Use of **prefer** usually results in a poorer choice than NTP can determine for itself.

Important Do not change the **maxpoll**, **minpoll**, or **version** keyword settings unless instructed to do so by Cisco TAC.

Use the following example to configure the necessary NTP association parameters:

```
configure
  ntp
    enable
    server ip_address1
    server ip_address2
    server ip_address3
    end
```

Notes:

- By default *context_name* is set to *local*. This is the recommended configuration.
- A number of options exist for the **server** command. Refer to the *NTP Configuration Mode Commands* chapter in the *Command Line Interface Reference* for more information.
- Enter the IP address of NTP servers using IPv4 dotted-decimal or IPv6 colon-separated-hexadecimal notation.

Important Configure the system with at least three (preferably four) NTP servers.

Save the configuration as described in the *Verifying and Saving Your Configuration* chapter.

Configuring NTP Servers with Local Sources

NTP can use network peers, local external clocks (such as GPS devices), or a local clock with no external source.

A local clock with no external source is usually a last-resort clock when no better clock is available. It is typically configured on a site's intermediate NTP server so that when a WAN network outage occurs, hosts within the site can continue to synchronize amongst themselves.

You can configure this in ntpd or on many commercially available NTP devices. This local clock should always have a high stratum number (8+) so that under normal conditions (when real sources are available) this local clock will not be used.

Using a Load Balancer

The NTP daemon and protocol assume that each configured server is running NTP. If a NTP client is configured to synchronize to a load balancer that relays and distributes packets to a set of real NTP servers, the load balancer may distribute those packets dynamically and confuse the NTP client. NTP packets are latency and jitter sensitive. Relaying them through a load balancer can confuse the NTP client and is not a supported practice.

Verifying the NTP Configuration

Verify the NTP configuration is correct. Enter the following command at the Exec mode prompt:

show ntp associations

The output displays information about all NTP servers. See the output below for an example deploying two NTP servers.

```
+----Peer Selection: ( ) - Rejected / No Response
|                    (x) - False Tick
|                    (.) - Excess
|                    (-) - Outlyer
|                    (+) - Candidate
|                    (#) - Selected
|                    (*) - System Peer
|                    (o) - PPS Peer
v
     remote           refid      st t when poll reach   delay   offset  jitter
==============================================================================
*10.81.254.202   .GPS.            1 u  160 1024  377   21.516    0.019   0.009
```

The following table describes the parameters output by the **show ntp associations** command.

Table 9: NTP Parameters

Column Title	Description
remote	List of the current NTP servers. One of these characters precedes each IP address to show the server's current condition: • ( ) Rejected/No response • X False tick • . Excess • - Outlyer • + Candidate • # Selected • * System peer • (o) PPS peer
refid	Last reported NTP reference to which the server is synchronizing.
st	NTP server stratum level.
t	Communication type: broadcast, multicast, etc.
when	Number of seconds since the last contact.
poll	Polling interval between the system and the NTP server.

Column Title	Description
reach	Octal value of the reachability shift register indicating which responses were received for the previous eight polls to this NTP server.
delay	Round-trip delay (in milliseconds) for messages exchanged between the system and the NTP server.
offset	Number of milliseconds by which the system clock must be adjusted to synchronize it with the NTP server.
jitter	Jitter in milliseconds between the system and the NTP server.

Configuring Software RSS

The Cisco Unified Computing System (USC) NIC supports hardware-based Receive Side Scaling (RSS); however RSS is only supported on IP traffic. For other network protocols, such as MPLS, GTP, L2TP, and GRE, all the traffic is routed into a single queue.

The VPC-DI provides a software RSS capability that distributes MPLS traffic to the available vCPU cores for processing. This increases resource utilization and provides improved throughput.

The software RSS capability can be supplemental to the Cisco UCS NIC hardware RSS support, meaning that it distributes some traffic not supported by the hardware NIC (MPLS traffic only in this release). The VPC-DI can also provide comprehensive RSS coverage, meaning that it distributes all traffic. This option is applicable when hardware that does not support RSS is used.

Configure the use of RSS with the **iftask sw-rss** command.

```
config
   iftask sw-rss {comprehensive | supplemental}
```

Use the **comprehensive** keyword to configure RSS for all incoming traffic. Use the **supplemental** keyword to configure RSS on protocols not supported by the hardware RSS functionality (MPLS traffic only in this release).

DI-Network RSS Encryption

Feature Summary and Revision History

Summary Data

Applicable Product(s) or Functional Area	All
Applicable Platform(s)	VPC-DI
Feature Default	Disabled - Configuration Required

Related Changes in This Release	Not applicable
Related Documentation	*VPC-DI System Administration Guide*

Revision History

Important Revision history details are not provided for features introduced before releases 21.2 and N5.1.

Revision Details	Release
The default setting for Distributed Instance Network (DI-network) RSS traffiic is now disabled and can be enabled with a new CLI command. In prior releases, this was functionality was automatically enabled and was not configurable.	21.8
First introduced.	Pre 21.2

Feature Changes

Previous Behavior: In Releases prior to 21.8, Receive Side Scaling (RSS) was enabled by default for all traffic on the internal Distributed Instance network (DI-network) for virtualized StarOS instances.

New Behavior: In Release 21.8 and later, RSS is disabled by default and can be enabled via a new CLI.

Command Changes

iftask di-net-encrypt-rss

This new CLI command has been added to control the enablement of RSS on encrypted traffic on the DI-network.

```
configure
   [no] iftask di-net-encrypt-rss
   end
```

Note The default setting is disabled.

Configuring SF Boot Configuration Pause

Under certain circumstances, within VPC-DI deployments, the CF applies the boot configuration before all SFs have completed their boot process.

The following Configuration Mode command, **wait cards active**, pauses configuration until all specified cards are operational or the timeout period expires (whichever criteria is met first). The pause occurs immediately following local management context creation and ntp/snmp configuration.

This command corrects a scenario where SFs come online late following chassis load or reload and the configuration pertaining to those SFs is not applied (and thereby lost).

```
configure
  [ no ] wait cards active { all | number } [ standby number ] timeout seconds

  end
```

Notes:

- **all**: Pause until all active mode cards attain operational status.
- *number* : Pause until the specified number of active mode cards attain operational status.*number* is 0 through the number of active mode cards.
- **standby** *number* : (Optional) Also wait for the specified number of non-active mode cards to attain operational status.

  number is 0 through the number of service slots not configured for active mode SFs.
- **timeout** *seconds*: Wait from 1 through 3600 *seconds* for the specified card set to attain operational status. The wait is terminated early when or if this condition is satisfied. Otherwise the wait is terminated when the timeout period expires.

The following example command instructs the system to wait up to 120 seconds for all active cards and 1 standby card to become active:

wait cards active all standby 1 timeout 120

Enabling CLI Timestamping

To display a timestamp (date and time) for every command that is executed on the CLI, enter the following command at the root prompt for the Exec mode:

```
timestamps
```

The date and time appear immediately after you execute the command.

Save the configuration as described in the *Verifying and Saving Your Configuration* chapter.

Configuring CLI Confirmation Prompts

A number of Exec mode and Global Configuration mode commands prompt users for a confirmation (Are you sure? [Yes|No]:) prior to executing the command.

This section describes configuration settings that:

- Automatically confirm commands for the current CLI session (Exec mode) or for all CLI sessions and users (Global Configuration mode).
- Requires confirmation prompting only for the Exec mode **configure** command and **autoconfirm** command.
- Selectively requires confirmation of Exec mode configuration commands.

Enabling Automatic Confirmation

You can use the **autoconfirm** command to disable confirmation prompting for configuration commands. The **autoconfirm** command is available in the Exec mode and Global Configuration mode. Enabling the autoconfirm feature automatically supplies a "Yes" response to configuration command prompts, including for critical commands such as **reload** and **shutdown**. By default autoconfirm is disabled.

In the Exec mode, autoconfirm applies only to the current interactive CLI session.

In the Global Configuration mode, autoconfirm applies to all CLI sessions for all CLI users:

```
configure
   autoconfirm
   end
```

To disable autoconfirm once it has been enabled, use the **no autoconfirm** command.

Important If commandguard is enabled, autoconfirm will disable commandguard.

Autoconfirm is intended as an "ease-of-use" feature. It presumes that the answer to "Are you sure? [Y/N]" prompts will be "Yes", and skips the prompt. Its use implies that the user is an expert who does not need these "safety-net" prompts.

Requiring Confirmation for autoconfirm and configure Commands

You can require confirmation prompting for the **autoconfirm** (Exec mode and Global Configuration mode) and **configure** (Exec mode) commands via the Global Configuration mode **commandguard** command.

Important If autoconfirm is enabled, commandguard will not take effect until autoconfirm is disabled in both Exec and Global Configuration modes.

The following command sequence enables the commandguard feature:

```
configure
   commandguard
   end
```

With commandguard enabled the confirmation prompt appears as shown in the example below:

```
[local]host_name# configure
Are you sure? [Yes|No]: yes
[local]host_name(config)#
```

To disable commandguard once it has been enabled, use the **no commandguard** command.

The status of **commandguard** is output in **show configuration** commands.

Requiring Confirmation for Specific Exec Mode Commands

A keyword for the **commandguard** command allows you to apply mandatory prompting for specified categories of Exec mode configuration commands, even when autoconfirm is enabled.

The command syntax is as follows:

```
configure
   commandguard exec-command exec_mode_category
   end
```

Notes:

- **exec-command** *exec_mode_category* specifies one of the following categories of Exec mode configuration commands.
  - card
  - clear
  - copy
  - debug
  - delete
  - filesystem
  - hd
  - reload
  - rename
  - shutdown
  - task
  - upgrade
- You can enter multiple **commandguard exec-command** *exec_mode_category* commands.
- All Exec mode commands beginning with the specified category word will prompt for confirmation, regardless if autoconfirm is enabled.
- You can turn off confirmation prompting for a specific category using **no commandguard exec-command** *exec_mode_category*.
- If autoconfirm is overridden by **commandguard exec-command** for an Exec mode command, StarOS displays an informational message indicating why autoconfirm is being overridden when you attempt to execute the command.
- Users may selectively override confirmation prompting for any Exec mode configuration command that supports the **-noconfirm** keyword.

For example, with **commandguard exec-command card** enabled, the confirmation prompt appears as shown below:

```
[local]host_name# card busy-out 1
Info: commandguard prevents autoconfirm of this command
Are you sure? [Yes|No]: yes
[local]host_name#
```

Configuring System Administrative Users

Getting Started describes how to configure a context-level security administrator for the system.

This section provides instructions for configuring additional administrative users having the following privileges:

- **Security Administrators**: have read-write privileges and can execute all CLI commands, including those available to Administrators, Operators, and Inspectors

- **Administrators**: have read-write privileges and can execute any command in the CLI except for a few security-related commands that can only be configured by Security Administrators. Administrators can configure or modify system settings and execute all system commands, including those available to the Operators and Inspectors.
- **Operators**: have read-only privileges to a larger subset of the Exec Mode commands. They can execute all commands that are part of the inspector mode, plus some system monitoring, statistic, and fault management functions. Operators do not have the ability to enter the Config Mode.
- **Inspectors**: are limited to a few read-only Exec Mode commands. The bulk of these are **show** commands for viewing a variety of statistics and conditions. An Inspector cannot execute **show configuration** commands and does not have the privilege to enter the Config Mode.

Configuration instructions are categorized according to the type of administrative user: context-level or local-user.

Important For information on the differences between these user privileges and types, refer to *Getting Started*.

User Name Character Restrictions

User names can only contain alphanumeric characters (a-z, A-Z, 0-9), hyphen, underscore, and period. The hyphen character cannot be the first character. This applies to AAA user names as well as local user names.

If you attempt to create a user name that does not adhere to these standards, you will receive the following message: "Invalid character; legal characters are "0123456789.-_abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ".

Configuring Context-level Administrative Users

This user type is configured at the context-level and relies on the AAA subsystems for validating user names and passwords during login. This is true for both administrative user accounts configured locally through a configuration file or on an external RADIUS or TACACS+ server. Passwords for these user types are assigned once and are accessible in the configuration file.

This section contains information and instructions for configuring context-level administrative user types.

It is possible to configure the maximum number of simulations CLI sessions on a per account or per authentication method basis. It will protect certain accounts that may have the ability to impact security configurations and attributes or could adversely affect the services, stability and performance of the system. The maximum number of simultaneous CLI sessions is configurable when attempting a new Local-User login and a new AAA context-based login. If the maximum number of sessions is set to 0, then the user is authenticated regardless of the login type. When the CLI task starts, a check is complete to identify the count. In this case, the CLI determines that the sessions for that user is 1 which is greater than 0 and it will display an error message in the output, it generate starCLIActiveCount and starCLIMaxCount SNMP MIB Objects and starGlobalCLISessionsLimit and starUserCLISessionsLimit SNMP MIB Alarms.

The **max-sessions** keyword for the **local-user username** *Global Configuration Mode* command configures the maximum number of simultaneous sessions available for a local user.

The **max-sessions** *Context Configuration Mode* command allows administrative users to configure the maximum simultaneous sessions allowed for corresponding users.

Refer to the *Command Line Interface Reference* for detailed information about these commands.

Configuring Context-level Security Administrators

Use the example below to configure additional security administrators:

```
configure
  context local
    administrator user_name { [ encrypted ] [ nopassword ] password password
 }
    end
```

Notes:

- Additional keyword options are available that identify active administrators or place time thresholds on the administrator. Refer to the *Command Line Interface Reference* for more information about the **administrator** command.
- The **nopassword** option allows you to create an administrator without an associated password. Enable this option when using ssh public keys (**authorized key** command in SSH Configuration mode) as a sole means of authentication. When enabled this option prevents someone from using an administrator password to gain access to the user account.

Save the configuration as described in the *Verifying and Saving Your Configuration* chapter.

Configuring Context-level Administrators

Use the example below to configure context-level configuration administrators:

```
configure
  context local
    config-administrator user_name { [ encrypted ] [ nopassword ] password
 password }
    end
```

Notes:

- Additional keyword options are available that identify active administrators or place time thresholds on the administrator. Refer to the *Command Line Interface Reference* for more information about the **config-administrator** command.
- The **nopassword** option allows you to create a config-administrator without an associated password. Enable this option when using ssh public keys (**authorized key** command in SSH Configuration mode) as a sole means of authentication. When enabled this option prevents someone from using a config-administrator password to gain access to the user account.

Save the configuration as described in the *Verifying and Saving Your Configuration* chapter.

Configuring Context-level Operators

Use the example below to configure context-level operators:

```
configure
  context local
    operator user_name { [ encrypted ] [ nopassword ] password password }
    end
```

Notes:

- Additional keyword options are available that identify active administrators or place time thresholds on the administrator. Refer to the *Command Line Interface Reference* for more information about the **operator** command.
- The **nopassword** option allows you to create an operator without an associated password. Enable this option when using ssh public keys (**authorized key** command in SSH Configuration mode) as a sole means of authentication. When enabled this option prevents someone from using an operator password to gain access to the user account.

Save the configuration as described in the *Verifying and Saving Your Configuration* chapter.

Configuring Context-level Inspectors

Use the example below to configure context-level inspectors:

```
configure
  context local
    inspector user_name { [ encrypted ] [ nopassword ] password password }
    end
```

Notes:

- Additional keyword options are available that identify active administrators or place time thresholds on the administrator. Refer to the *Command Line Interface Reference* for more information about the **inspector** command.
- The **nopassword** option allows you to create an inspector without an associated password. Enable this option when using ssh public keys (**authorized key** command in SSH Configuration mode) as a sole means of authentication. When enabled this option prevents someone from using an inspector password to gain access to the user account.

Save the configuration as described in the *Verifying and Saving Your Configuration* chapter.

Segregating System and LI Configurations

Lawful Intercept (LI) configuration includes sensitive information. By default in a Normal build, an administrator without li-administration privilege can view the LI configuration commands. However, display of the LI configuration commands can be restricted or segregated from the rest of the system configuration.

The Global Configuration mode **require segregated li-configuration** command permanently segregates display of System and Lawful Intercept CLI. The CLI commands with Lawful-Intercept keyword are encrypted and can only be viewed by an administrator with li-administration privilege.

Important In a Trusted build, LI segregation is turned on and cannot be disabled. The **require segregated li-configuration** command is invisible.

Segregating LI configuration from system configuration has the following impacts on StarOS:

- Only administrators with li-administration privilege can see Lawful Intercept CLI commands in the output of the **show configuration** command.
- Executing the **save configuration** command will automatically encrypt Lawful Intercept CLI configuration commands.

- When loading a saved configuration file via CLI command (for example, **configure** <*url*>), encrypted Lawful Intercept CLI commands will be decrypted and executed only for an administrator with LI privilege. For an administrator without LI privilege, encrypted Lawful Intercept CLI commands will not be decrypted and executed.
- During a system boot wherein the boot config is loaded, encrypted Lawful Intercept configuration will be decrypted and loaded silently, in other words Lawful Intercept CLI configuration will not be visible on the console port.
- The Exec mode **configure** command now supports a keyword that allows an LI administrator to load only encrypted Lawful Intercept configuration from a saved configuration file (for example, **configure encrypted** <*url*>). The **encrypted** keyword can only be executed by an LI Administrator.
- If you are running a system with encrypted Lawful Intercept configuration (segregated LI), the output of the **show boot initial-config** command contains a line indicating whether it needed to run the second pass or not during the initial boot. This line displays "encrypted li" if the encrypted Lawful Intercept configuration was processed. If the line reads "encrypted li errors" then the second pass was not successful, or gave some output which was not expected or informational in nature.
- A user with li-administration privileges can view the boot config output for the encrypted Lawful Intercept configuration with the **show logs encrypted-li** command.

For a detailed description of the Global Configuration mode **require segregated li-configuration** and associated commands, see the *Lawful Intercept CLI Commands* appendix in the *Lawful Intercept Configuration Guide*.

Note The *Lawful Intercept Configuration Guide* is not available on www.cisco.com. Contact your Cisco account representative to obtain a copy of this guide.

In Release 21.4 and higher (Trusted builds only):

- Users can only access the system through their respective context interface.
- If the user attempts to log in to their respective context through a different context interface, that user will be rejected.
- Irrespective of whether the users are configured in any context with 'authorized-keys' or 'allowusers', with this feature these users will be rejected if they attempt to log in via any other context interface other than their own context interface.
- Users configured in any non-local context are required to specify which context they are trying to log in to. For example:

  ssh username@ctx_name@ctx_ip_addrs

Verifying Context-level Administrative User Configuration

Verify that the configuration was successful by entering the following command:

```
show configuration context local
```

This command displays all of the configuration parameters you modified within the Local context during this session. The following displays sample output for this command. In this example, a security administrator named *testadmin* was configured.

```
config
  context local
    interface mgmt1
      ip address 192.168.1.10 255.255.255.0
    #exit
    subscriber default
    #exit
    administrator testadmin encrypted password fd01268373c5da85
    inspector testinspector encrypted password 148661a0bb12cd59
exit
  port ethernet 5/1
    bind interface mgmt1 local
  #exit
```

Configuring Local-User Administrative Users

The local user type supports ANSI T1.276-2003 password security protection. Local-user account information, such as passwords, password history, and lockout states, is maintained in /flash. This information is saved immediately in a separate local user database subject to AAA based authentication and is not used by the rest of the system. As such, configured local-user accounts are not visible with the rest of the system configuration.

Important In release 20.0 and higher Trusted StarOS builds, the local user database is disabled. The Global Configuration mode **local-user** commands, and Exec mode **show local-user** and **update local-user** commands are unavailable. For additional information on Trusted builds, see the *System Operation and Configuration* chapter.

Use the example below to configure local-user administrative users:

```
configure
  local-user username name
  end
```

Notes:

- Additional keyword options are available identify active administrators or place time thresholds on the administrator. Refer to the *Command Line Interface Reference* for more information about the **local-user username** command.

For additional information on the local-user database, see Updating and Downgrading the local-user Database, on page 112.

Verifying Local-User Configuration

Verify that the configuration was successful by entering the following command:

```
show local-user verbose
```

This command displays information on configured local-user administrative users. A sample output for this command appears below. In this example, a local-user named *SAUser* was configured.

```
Username:                SAUser
  Auth Level:            secadmin
  Last Login:            Never
  Login Failures:        0
  Password Expired:      Yes
  Locked:                No
  Suspended:             No
```

```
Lockout on Pw Aging:   Yes
Lockout on Login Fail: Yes
```

Updating Local-User Database

Update the local-user (administrative) configuration by running the following Exec mode command. This command should be run immediately after creating, removing or editing administrative users.

update local-user database

Updating and Downgrading the local-user Database

Prior to release 20.0, local-user passwords were hashed with the MD5 message digest-algorithm and saved in the local-user database. In release 20. 0, PBKDF2 (Password Based Key Derivation Function - Version 2) is now used to derive a key of given length, based on entered data, salt and number of iterations. Local-user account passwords are hashed using the PBKDF2 method with a randomly generated salt coupled with a large number of iterations to make password storage more secure.

When upgrading to release 20.0, existing user passwords in the local-user database are not automatically upgraded from MD5 to PBKDF2 hashing (only hashed password values are stored). Since hash functions are one-way, it is not possible to derive user passwords from the stored hash values. Thus it is not possible to convert existing hashed passwords to strongly hashed passwords automatically.

To update the database, a Security Administrator must run the Exec mode **update local-user database** CLI command. When this command is executed, StarOS reads the database from the /flash directory, reconstructs the database in the new format, and writes it back to the disk.

The database upgrade process does not automatically convert MD5 hashed passwords into the PBKDF2 format. StarOS continues to authenticate users using the old encryption algorithm. It flags the users using the old encryption algorithm with a "Weak Hash" flag. This flag appears in the output of the **show local-user [verbose]** Exec mode CLI command. When users re-login with their credentials, StarOS verifies the entered password using the MD5 algorithm, then creates a new hash using the PBKDF2 algorithm and then saves the result in the database. StarOS then clears the "Weak Hash" flag for that user.

Important Since hash functions are one-way, it is not possible to convert PBKDF2 hashed passwords to the MD5 format. The local-user database must be downgraded prior to reverting to StarOS releases prior to 20.0.

To downgrade the local-user database to use the MD5 hash algorithm, a Security Administrator must run the Exec mode **downgrade local-user database** command. StarOS prompts for confirmation and requests the Security Administrator to reenter a password. The entered password re-authenticates the user prior to executing the downgrade command. After verification, the password is hashed using the appropriate old/weak encryption algorithm and saved in the database to allow earlier versions of StarOS to authenticate the Security Administrator.

The downgrade process does not convert PBKDF2 hashed passwords to MD5 format. The downgrade process re-reads the database (from the /flash directory), reconstructs the database in the older format, and writes it back to the disk. Since the PBKDF2 hashed passwords cannot be converted to the MD5 hash algorithm, and earlier StarOS releases cannot parse the PBKDF2 encryption algorithm, StarOS suspends all those users encrypted via the PBKDF2 algorithm. Users encrypted via the MD5 algorithm ("Weak Hash" flag) can continue to login with their credentials. After the system comes up with the earlier StarOS release, suspended users can be identified in the output of the **show local-user [verbose]**command.

To reactivate suspended users a Security Administrator can:

- Set temporary passwords for suspended users, using the Exec mode **password change local-user** *username* command.
- Reset the suspend flag for users, using the Configuration mode **no suspend local-user** *username* command.

Restricting User Access to a Specified Root Directory

By default an admin user who has FTP/SFTP access can access and modify any files under the /mnt/user/ directory. Access is granted on an "all-or-nothing" basis to the following directories: /flash, /cdrom, /hd-raid, /records, /usb1 and /usb2.

An administrator or configuration administrator can create a list of SFTP subsystems with a file directory and access privilege. When a local user is created, the administrator assigns an SFTP subsystem. If the user's authorization level is not security admin or admin, the user can only access the subsystem with read-only privilege. This directory is used as the user's root directory. The information is set as environmental variables passed to the openssh sftp-server.

You must create the SFTP root directory before associating it with local users, administrators and config administrators. You can create multiple SFTP directories; each directory can be assigned to one or more users.

Configuring an SFTP root Directory

The **subsystem sftp** command allows the assignment of an SFTP root directory and associated access privilege level.

```
configure
   context local
      server sshd
         subsystem sftp  [ name sftp_name root-dir pathname mode { read-only
 | readwrite } ]
```

Notes:

- *sftp_name* is an alphanumeric string that uniquely identifies this subsystem.
- *pathname* specifies the root directory to which SFTP files can be transferred. Options include:
  - /hd-raid/records/cdr
  - /flash

Associating an SFTP root Directory with a Local User

The **local-user username** command allows an administrator to associate an SFTP root directory with a specified username.

```
configure
   local-user username user_name authorization-level level ftp sftp-server
sftp_name password password
   exit
```

Associating an SFTP root Directory with an Administrator

The **administrator** command allows an administrator to associate an SFTP root directory for a specified administrator.

```
configure
   context local
     administrator user_name password password ftp sftp-server sftp_name
     exit
```

Associating an SFTP root Directory with a Config Administrator

The config-administrator command allows an administrator to associate an SFTP root directory with a specified configuration administrator.

```
configure
   context local
     config-administrator user_name password password ftp sftp-server sftp_name
     exit
```

Configuring TACACS+ for System Administrative Users

This section describes TACACS+ (Terminal Access Controller Access Control System+) AAA (Authentication Authorization and Accounting) service functionality and configuration on the VPC-DI.

Operation

TACACS+ is a secure, encrypted protocol. By remotely accessing TACACS+ servers that are provisioned with the administrative user account database, the VPC-DI system can provide TACACS+ AAA services for system administrative users. TACACS+ is an enhanced version of the TACACS protocol that uses TCP instead of UDP.

The system serves as the TACACS+ Network Access Server (NAS). As the NAS the system requests TACACS+ AAA services on behalf of authorized system administrative users. For the authentication to succeed, the TACACS+ server must be in the same local context and network accessed by the system.

The system supports TACACS+ multiple-connection mode. In multiple-connection mode, a separate and private TCP connection to the TACACS+ server is opened and maintained for each session. When the TACACS+ session ends, the connection to the server is terminated.

TACACS+ is a system-wide function on the VPC-DI. TACACS+ AAA service configuration is performed in TACACS Configuration Mode. Enabling the TACACS+ function is performed in the Global Configuration Mode. The system supports the configuration of up to three TACACS+ servers.

Once configured and enabled on the system, TACACS+ authentication is attempted first. By default, if TACACS+ authentication fails, the system then attempts to authenticate the user using non-TACACS+ AAA services, such as RADIUS.

It is possible to configure the maximum number of simulations CLI sessions on a per account or per authentication method basis. It will protect certain accounts that may have the ability to impact security configurations and attributes or could adversely affect the services, stability and performance of the system. The maximum number of simultaneous CLI sessions is configurable when attempting a new TACACS+ user login. The recommendation is to use the max-sessions feature is through the TACACS+ server attribute option **maxsess**. The second way is though the StarOS CLI configuration mode TACACS+ mode using the **maxsess** keyword in the **user-id** command. If the maximum number of sessions is set to 0, then the user is authenticated regardless of the login type. When the CLI task starts, a check is complete to identify the count. In this case, the CLI determines that the sessions for that user is 1 which is greater than 0 and it will display an error

message in the output, it generate starCLIActiveCount and starCLIMaxCount SNMP MIB Objects and starGlobalCLISessionsLimit and starUserCLISessionsLimit SNMP MIB Alarms.

The **max-sessions** *TACACS+ Configuration Mode command* configures the maximum number of sessions available for TACACS+. Also the **default** option for the **user-id** *TACACS+ Configuration Mode command* configures the default attributes for a specific TACACS+ user identifier. Refer to the *Command Line Interface Reference* for detailed information about these commands.

Important The user can define the maximum number of simulations CLI sessions available in both the StarOS and TACACS+ server configuration. However, this option is extremely discouraged.

Important *For releases after 15.0 MR4,* TACACS+ accounting (CLI event logging) will not be generated for Lawful Intercept users with privilege level set to 15 and 13.

User Account Requirements

Before configuring TACACS+ AAA services, note the following TACACS+ server and StarOS user account provisioning requirements.

TACACS+ User Account Requirements

The TACACS+ server must be provisioned with the following TACACS+ user account information:

- A list of known administrative users.
- The plain-text or encrypted password for each user.
- The name of the group to which each user belongs.
- A list of user groups.
- TACACS+ privilege levels and commands that are allowed/denied for each group.

Important TACACS+ privilege levels are stored as Attribute Value Pairs (AVPs) in the network's TACACS+ server database. Users are restricted to the set of commands associated with their privilege level. A mapping of TACACS+ privilege levels to StarOS CLI administrative roles and responsibilities is provided in the table below.

To display the default mapping of TACACS+ privilege levels to CLI administrative roles, run the Exec mode **show tacacs priv-lvl** command. The default mapping varies based on the StarOS release and build type.

TACACS+ priv-levels can be reconfigured from their default StarOS authorization values via the TACACS+ Configuration mode **priv-lvl** and **user-id** commands. For additional information, see the *TACACS+ Configuration Mode Commands* chapter of the *Command Line Interface Reference*.

Important In release 20.0 and higher Trusted StarOS builds, FTP is not supported.

StarOS User Account Requirements

TACACS+ users who are allowed administrative access to the system must have the following user account information defined in StarOS:

- username
- password
- administrative role and privileges

Important For instructions on defining users and administrative privileges on the system, refer to *Configuring System Administrative Users*.

Configuring TACACS+ AAA Services

This section provides an example of how to configure TACACS+ AAA services for administrative users on the system.

Caution When configuring TACACS+ AAA services for the first time, the administrative user must use non-TACACS+ services to log into the StarOS. Failure to do so will result in the TACACS+ user being denied access to the system.

Log in to the system using non-TACACS+ services.

Use the example below to configure TACACS+ AAA services on the system:

```
configure
   tacacs mode
      server priority priority_number ip-address tacacs+srvr_ip_address
      end
```

Note:

- **server priority** *priority_number*: Must be an integer from 1 to 3 (*releases prior to 18.2*) or 1 through 4 (*releases 18.2+*), that specifies the order in which this TACACS+ server will be tried for TACACS+ authentication. 1 is the highest priority, and 3 or 4 is the lowest. The priority number corresponds to a configured TACACS+ server.
- **ip-address**: Must be the IPv4 address of a valid TACACS+ server that will be used for authenticating administrative users accessing this system via TACACS+ AAA services.
- By default, the TACACS+ configuration will provide authentication, authorization, and accounting services.

Enable TACACS+ on the StarOS:

```
configure
   aaa tacacs+
   end
```

For additional information, see Disable TACACS+ Authentication for Console, on page 118.

Save the configuration as described in the *Verifying and Saving Your Configuration* chapter.

Important For complete information on all TACACS+ Configuration Mode commands and options, refer to the *TACACS Configuration Mode Commands* chapter in the *Command Line Reference*.

Configuring TACACS+ for Non-local VPN Authentication

By default TACACS+ authentication is associated with login to the local context. TACACS+ authentication can also be configured for non-local context VPN logins. TACACS+ must configured and enabled with the option described below.

A **stop** keyword option is available for the TACACS+ Configuration mode **on-unknown-user** command. If TACACS+ is enabled with the command-keyword option, the VPN context name into which the user is attempting a login must match the VPN name specified in the username string. If the context name does not match, the login fails and exits out.

Without this option the login sequence will attempt to authenticate in another context via an alternative login method. For example, without the **on-unknown-user stop** configuration, an admin account could log into the local context via the non-local VPN context. However, with the **on-unknown-user stop** configuration, the local context login would not be attempted and the admin account login authentication would fail.

```
configure
   tacacs mode
      on-unkown-user stop &quest;
      end
```

Verifying the TACACS+ Configuration

This section describes how to verify the TACACS+ configuration.

Log out of the system CLI, then log back in using TACACS+ services.

Important Once TACACS+ AAA services are configured and enabled on the StarOS, the system first will try to authenticate the administrative user via TACACS+ AAA services. By default, if TACACS+ authentication fails, the system then continues with authentication using non-TACACS+ AAA services.

At the Exec Mode prompt, enter the following command:

```
show tacacs [ client | priv-lvl | session | summary ]
```

The output of the **show tacacs** commands provides summary information for each active TACACS+ session such as username, login time, login status, current session state and privilege level. Optional filter keywords provide additional information.

An example of this command's output is provided below. In this example, a system administrative user named *asradmin* has successfully logged in to the system via TACACS+ AAA services.

```
active session #1:
   login username               : asradmin
   login tty                    : /dev/pts/1
   time of login                : Fri Oct 22 13:19:11 2011
   login server priority        : 1
   current login status         : pass
   current session state        : user login complete
   current privilege level      : 15
   remote client application    : ssh
   remote client ip address     : 111.11.11.11
   last server reply status     : -1
total TACACS+ sessions          : 1
```

Important For details on all TACACS+ maintenance commands, refer to the *Command Line Interface Reference*.

IPv6 Address Support for TACACS+ Server

Separating Authentication Methods

You can configure separate authentication methods for accessing the Console port and establishing SSH/telnet sessions (vty lines).

If you configure TACACS+ globally, access to the Console and vty lines are both authenticated using that method.

Since the Console port is a last resort access to StarOS, you can configure local authentication for the Console and employ TACACS+ for the vty lines.

Important This feature extends to AAA (Authentication, Authorization and Accounting) service as well as local users. For example, local-users may have only Console access and AAA (VPN context) users with access only via vty lines.

Separating authentication methods (Console versus vty lines) requires disabling Console access for users based on the type of authentication.

Disable TACACS+ Authentication for Console

A **noconsole** keyword for the Global Configuration mode **aaa tacacs+** command disables TACACS+ authentication on the Console line.

```
configure
  aaa tacacs+ noconsole
  exit
```

By default, TACACS+ server authentication is performed for login from a Console or vty line. With **noconsole** enabled, TACACS+ authentication is bypassed in favor of local database authentication for a console line; on vty lines, TACACS+ remains enabled.

Important When **aaa tacacs+ noconsole** is configured, a local user with valid credentials can log into a Console port even if **on-authen-fail stop** and **on-unknown-user stop** are enabled via the TACACS+ Configuration mode. If the user is not a TACACS+ user, he/she cannot login on a vty line.

Disable AAA-based Authentication for Console

A **noconsole** keyword for the Global Configuration mode **local-user allow-aaa-authentication** command disables AAA-based authentication on the Console line.

```
configure
  local-user allow-aaa-authentication noconsole
  exit
```

Since local-user authentication is always performed before AAA-based authentication and **local-user allow-aaa-authentication noconsole** is enabled, the behavior is the same as if **no local-user allow-aaa-authentication** is configured. There is no impact on vty lines.

Important This command does not apply for a Trusted build because the local-used database is unavailable.

Disable TACACS+ Authentication at the Context Level

When you enable **aaa tacacs+** in the Global Configuration mode, TACACS+ authentication is automatically applied to all contexts (local and non-local). In some network deployments you may wish to disable TACACS+ services for a specific context(s).

You can use the **no aaa tacacs+** Context Configuration command to disable TACACS+ services within a context.

```
configure
  context ctx_name
    no aaa tacacs+
```

Use the **aaa tacacs+** Context Configuration command to enable TACACS+ services within a context where it has been previously disabled.

Important AAA TACACS+ services must be enabled in the Global Configuration mode (all contexts) before you can selectively disable the services at the context level. You cannot selectively enable TACACS+ services at the context level when it has not been enabled globally.

Limit local-user Login on Console/vty Lines

As a security administrator when you create a StarOS user you can specify whether that user can login through the Console or vty line. The **[noconsole | novty]** keywords for the Global Configuration mode **local-user username** command support these options.

```
configure
  local-user username <username> [ noconsole | novty ]
  exit
```

The **noconsole** keyword prevents the user from logging into the Console port. The **novty** keyword prevents the user from logging in via an SSH or telnet session. If neither keyword is specified access to both Console and vty lines is allowed.

Important Use of the **noconsole** or **novty** keywords is only supported on the new local-user database format. If you have not run **update local-user database**, you should do so before enabling these keywords. Otherwise, **noconsole** and **novty** keywords will not be saved in the local-user database. After a system reboot, all users will still be able to access the Console and vty lines. For additional information, see the Updating and Downgrading the local-user Database, on page 112.

Important This command does not apply for a Trusted build because the local-used database is unavailable.

Limit Console Access for AAA-based Users

AAA-based users normally login through on a vty line. However, you may want to limit a few users to accessing just the Console line. If you do not use the local-user database (or you are running a Trusted build), this needs to be done by limiting access to the Console line for other AAA-based users. Enable the **noconsole** keyword for all levels of admin users that will not have access to the Console line.

The **noconsole** keyword is available for the Context Configuration mode commands shown below.

```
configure
  context <ctx_name>
    administrator <username> { encrypted | nopassword | password } noconsole

    config-administrator <username> { encrypted | nopassword | password }
noconsole
    inspector <username> { encrypted | nopassword | password } noconsole
    operator <username> { encrypted | nopassword | password } noconsole
    exit
```

The **noconsole** keyword disables user access to the Console line. By default **noconsole** is not enabled, thus all AAA-based users can access the Console line.

Important The **local-user allow-aaa-authentication noconsole** command takes precedence. In that case, all AAA-based users cannot access the Console line.

Verify Configuration Changes

You can verify changes made related to the separation of authentication methods via the Exec mode **show configuration** command. After saving the configuration changes, run **show configuration |grep noconsole** and **show configuration |grep novty**. The output of these commands will indicate any changes you have made.

Configuring a Chassis Key

A chassis key should be configured for each system. This key is used to decrypt encrypted passwords found in configuration files.

Overview

The chassis key is used to encrypt and decrypt encrypted passwords in the configuration file. If two or more chassis are configured with the same chassis key value, the encrypted passwords can be decrypted by any of the chassis sharing the same chassis key value. As a corollary to this, a given chassis key value will not be able to decrypt passwords that were encrypted with a different chassis key value.

The chassis key is used to generate the chassis ID which is stored in a file and used as the master key for protecting sensitive data (such as passwords and secrets) in configuration files

For release 15.0 and higher, the chassis ID is an SHA256 hash of the chassis key. The chassis key can be set by users through a CLI command or via the Quick Setup Wizard. If the chassis ID does not exist, a local MAC address is used to generate the chassis ID.

For release 19.2 and higher, the user must explicitly set the chassis key through the Quick Setup Wizard or CLI command. If it is not set, a default chassis ID using the local MAC address will not be generated. In the absence of a chassis key (and hence the chassis ID), sensitive data will not appear in a saved configuration file. The chassis ID is the SHA256 hash (encoded in base36 format) of the user entered chassis key plus a 32-byte secure random number. This assures that the chassis key and chassis ID have 32-byte entropy for key security.

If a chassis ID is not available encryption and decryption for sensitive data in configuration files will not work.

Configuring a New Chassis Key Value

CLI Commands

Important Only a user with Security Administrator privilege can execute the **chassis key value** and **chassis keycheck** commands.

Use the Exec mode **chassis key value** *key_string* command to enter a new chassis key.

The *key_string* is an alphanumeric string of 1 through 16 characters. The chassis key is stored as a one-way encrypted value, much like a password. For this reason, the chassis key value is never displayed in plain-text form.

The Exec mode **chassis keycheck** *key_string* command generates a one-way encrypted key value based on the entered *key_string*. The generated encrypted key value is compared against the encrypted key value of the previously entered chassis key value. If the encrypted values match, the command succeeds and keycheck passes. If the comparison fails, a message is displayed indicating that the key check has failed. If the default chassis key (MAC address) is currently being used, this key check will always fail since there will be no chassis key value to compare against.

Use the **chassis keycheck** command to verify whether multiple chassis share the same chassis key value.

Important For release 19.2 and higher, in the absence of an existing chassis ID file the **chassis keycheck** command is hidden.

For additional information, refer to the *Exec Mode Commands* chapter in the *Command Line Interface Reference*.

Beginning with Release 15.0, the chassis ID will be generated from the chassis key using a more secure algorithm. The resulting 44-character chassis ID will be stored in the same file.

Release 14 and Release 15 chassis IDs will be in different formats. Release 15 will recognize a Release 14 chassis ID and consider it as valid. Upgrading from 14.x to 15.0 will not require changing the chassis ID or configuration file.

However, if the chassis key is reset in Release 15 through the Quick Setup Wizard or CLI command, a new chassis ID will be generated in Release 15 format (44 instead of 16 characters). Release14 builds will not recognize the 44-character chassis ID. If the chassis is subsequently downgraded to Release 14, a new 16-character chassis ID will be generated. To accommodate the old key format, you must save the configuration file in pre-v12.2 format before the downgrade. If you attempt to load a v15 configuration file on the downgraded chassis, StarOS will not be able to decrypt the password/secrets stored in the configuration file.

For release 19.2 and higher, in a chassis where the chassis ID file already exists nothing is changed. However, if the chassis ID file is lost in both management cards, all existing configuration files become invalid. Entering a new chassis key that is the same as the original value will not resolve the issue because of the new method used to generate the chassis ID.

Caution After setting a new chassis key, you must save the configuration before initiating a reload. See the *Verifying and Saving Your Configuration* chapter.

Quick Setup Wizard

The Quick Setup Wizard prompts the user to enter a chassis key value. If a chassis key value is not entered a default chassis is generated using the chassis' MAC address (releases prior to 20.0).

For releases 20.0 and higher, if the chassis ID file does not exist, the Quick Setup Wizard prompts the user to enter a chassis key. A default chassis ID is not generated if a chassis key is not entered.

To run the Quick Setup Wizard, execute the Exec mode **setup** command.

```
[local]host_name# setup
1. Do you wish to continue with the Quick Setup Wizard[yes/no]: y
2. Enable basic configuration[yes/no]: y
3. Change chassis key value[yes/no]: y
4. New chassis key value: key_string
```

Enabling Automatic Reset of FSC Fabric

By default if an excessive number of discarded fabric egress packets occurred in the switch fabric, a manual reset of the Fabric Storage Card(s) is required for fabric recovery.

You can optionally enable automatic resets of FSCs if an excessive number of discarded fabric egress packets is detected.

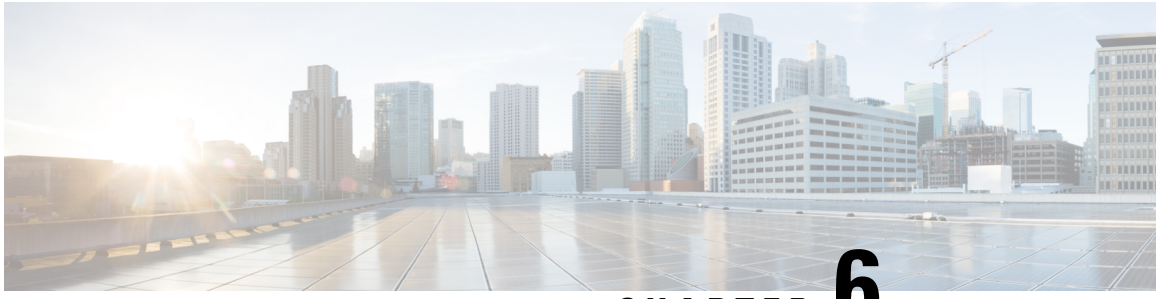
A Global Configuration mode **fabric fsc-auto-recover** command enables or disables automatic FSC resets upon detection of an excessive number of discarded fabric egress packets.

The following command sequence enables this feature:

```
configure
   fabric fsc-auto-recovery { disable | enable } [ max-attempts [
number_attempts | unlimited ] ]
   end
```

max-attempts [*number_attempts* **| unlimited]** specifies how many times StarOS will attempt to reset each FSC as an integer from 1 to 99 or unlimited (will not stop until FSC is reset). The default setting is 1.

Important To enable this feature, you must first configure the Fabric Egress Drop Threshold via the Global Configuration mode **fabric egress drop-threshold** command.



CHAPTER 6

Config Mode Lock Mechanisms

This chapter describes how administrative lock mechanisms operate within StarOS configuration mode.

It contains the following sections:

Overview of Config Mode Locking

You enter the Global Configuration (config) mode via the Exec mode **configure** command. By default all administrative users share config mode. Multiple administrative users can share access to config mode simultaneously. This is called a shared-lock.

The primary indication for the existence of a shared-lock is a message displayed when entering config mode.

```
Warning: One or more other administrators may be configuring this system
```

Note There are no default restrictive behavior changes when entering config mode under a shared-lock.

When multiple administrators edit or save the running config, concurrent changes may result in conflicting, inconsistent, or missing configuration commands. A similar problem can occur when saving the configuration if someone is attempting to restart the system.

An optional **lock [force | warn]** keyword for the **configure** command allows an administrator to request a mutually exclusive lock of the config mode to assure that no other user is simultaneously modifying the configuration. This is called an exclusive-lock. Once an exclusive-lock is granted to an administrator, no one else can access config mode for the duration of the session while the lock is held. The exclusive-lock is terminated only when the user holding the lock exits to Exec mode.

A shutdown-lock is enabled during a save configuration operation to prevent other users from reloading or shutting down the system while the configuration is being saved.

Config mode locking mechanisms such as shared-lock, exclusive-lock and shutdown-lock mitigate the possibility of conflicting commands, file corruption and reboot issues.

Requesting an Exclusive-Lock

Important To avoid complications resulting from the failure of an administrator holding an exclusive lock to exit config mode, it is a best practice to configure all administrator accounts with CLI session absolute timeouts and/or idle timeouts. For additional information on setting these timeouts, see the *Using the CLI for Initial Configuration* section of the *Getting Started* chapter in this guide.

You can request an exclusive-lock on config mode by executing the Exec mode **configure lock** command.

```
[local]host_name# configure [ <url> ] lock [ force | warn ]
```

If you specify a URL, the exclusive lock is associated with the pre-loaded configuration file. If you do not specify a URL, the exclusive lock is granted for the running configuration. For additional information see Effect of Config Lock on URL Scripts, on page 127.

The **force** option forces all other administrators to exit out of configuration mode, including anyone currently holding the exclusive-lock.

The **warn** option warns all other administrators to exit out of configuration mode. This administrator will be taking the exclusive-lock soon. You may want to use this option before actually forcing administrators out of configuration mode,

If there are no other administrators in config mode, entering **configure lock** immediately grants you an exclusive-lock.

```
[local]host_name# configure lock
Info: No one else can access config mode while you have the exclusive lock
[local]host_name#
```

When the exclusive lock is granted, no other administrators are allowed to enter into config mode or load a config file. Any other administrators attempting to enter into config mode or load a config file will see the following message:

```
Failure: User <username> has the exclusive lock
 - please enter 'show administrators' for more information
```

If another administrator attempts to enter config mode with the exclusive-lock when it is already enabled, the following message appears:

```
Failure: Another administrator is still in configuration mode
 - please enter 'show administrators' for more information
```

If you do not obtain an exclusive lock initially, you can use **configure lock force**.

If **configure lock force** is successful, all users who have been forced to exit to Exec mode will see a warning message indicating that they were forced to exit from config mode:

```
[local]host_name(config)#
 Warning: Administrator <username> has forced you to exit from configuration mode
[local]host_name#
```

A **configure lock force** command may not be successful because there is a very small chance that another administrator may be in the middle of entering a password or performing a critical system operation that cannot be interrupted. In this case a failure message will appear:

```
[local]host_name# configure lock force
 Failure: Another administrator could not release the configuration mode lock
 - please enter 'show administrators' for more information
```

The **configure lock warn** command sends a warning message to all config mode users (if any) and then waits up to 10 seconds to try and acquire the exclusive-lock. If any users are still in config mode, the config mode remains in a shared-lock state.

```
[local]host_name# configure lock warn
 please wait for this message to be sent to the other administrators......
 [local]host_name(config)#
```

The other administrators would eventually see this message in their session output:

```
[local]host_name(config)#
 Administrator <username> requires exclusive access to configuration mode
 >>> You need to exit from configuration mode as soon as possible <<<
 [local]host_name#
```

The **configure lock warn** command does not usually result in the exclusive-lock being acquired since the other administrators would typically not anticipate seeing the message in their session output.

Important StarOS logs all major config mode lock interactions to the event log and syslog facility (if configured). You can access a record of what interactions transpired at any time.

Effect of Config Lock on URL Scripts

When attempting to load a config script file using the **configure** <*url*> command, you must acquire either the shared-lock (default) or the exclusive-lock. Since the config script file typically contains the **config** command, the lock is actually held before and after the **config** command is parsed and executed.

The lock is held throughout the execution of the entire config file. Since the same shared-lock is used as the interactive config mode lock, a warning message is displayed followed by a confirmation prompt (if **-noconfirm** is not enabled) as shown in the example below.

```
 [local]host_name# config /flash/myconfig.cfg
 Warning: One or more other administrators may be configuring this system
 Are you sure? [Y/N]:
```

With **-noconfirm** enabled, since all the commands are also echoed to the screen, the warning message will likely scroll off the screen and may not be noticed.

Important When StarOS first starts up, the Initial Boot Config File is always exclusively locked while loading.

Saving a Configuration File

Saving a partial or incomplete configuration file can cause StarOS to become unstable when the saved configuration is loaded at a later time. StarOS inhibits the user from saving a configuration which is in the process of being modified.

With a shared-lock in-effect for the duration of the save operation, you are prompted to confirm the save operation.

```
[local]host_name# save configuration /flash/config.cfg
 Warning: One or more other administrators may be configuring this system
 Are you sure? [Y/N]:
```

If an exclusive-lock is being held by a user, the save operation will fail.

```
[local]host_name# save configuration /flash/config.cfg
Failure: Configuration mode is currently locked, use ignore-lock to ignore lock
```

You can use the **ignore-locks** keyword with the **save configuration** command to override an existing exclusive-lock.

```
[local]host_name# save configuration /flash/config.cfg ignore-locks
 Warning: Ignoring the configuration mode lock held by another administrator
```

Important The **save configuration** command also enables a shutdown-lock that prevents any other users from reloading or shutting down the system while the configuration is being saved. For additional information, refer to Reload and Shutdown Commands, on page 128.

Reload and Shutdown Commands

The Exec mode **reload** and **shutdown** commands can result in a corrupted or partial configuration file when either of these commands are executed while a **save configuration** command is still in progress.

To prevent this problem from occurring, the **reload** and **shutdown** commands share a CLI shutdown-lock with all **save configuration** commands executed across StarOS. This means while any **save configuration** command is executing, StarOS cannot execute a **reload** or **shutdown** command. These commands are queued indefinitely until all save configuration operations are complete.

To prevent the user from being “hung” indefinitely in the wait queue, the user may press Control+C to exit the wait as shown in the example below.

```
[local]host_name# reload
Are you sure? [Yes|No]: yes
Waiting for other administrators to finish saving configuration
(ctrl-c to abort) .......^C
Action aborted by ctrl-c
[local]host_name#
```

On those rare occasions when you must reboot StarOS immediately regardless of the risk of corrupting any file(s) in the process of being saved, you can use the **ignore-locks** keyword in combination with the **reload** or **shutdown** command. With this option StarOS displays the appropriate warning message, but does not wait for save configuration operations to complete before initiating the reboot.

```
[local]host_name# reload ignore-locks -noconfirm
Warning: One or more other administrators are saving configuration
Starting software 21.0...
Broadcast message from root (pts/2) Wed May 11 16:08:16 2016...
The system is going down for reboot NOW !!
```

Caution Employing the **ignore-locks** keyword when rebooting the system may corrupt the configuration file.

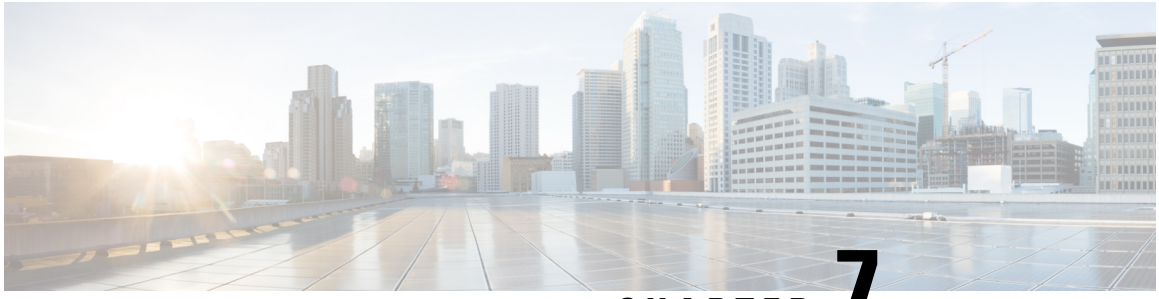
show administrators Command

The Exec mode **show administrators** command has a single-character "M" column that indicates the current lock mode for the administrator's session. The M-mode characters are defined as follows:

- [blank] – Administrator is in Exec mode
- **c** – Administrator session is currently in Config Mode (shared-lock)
- **s** – Administrator session is currently saving the config
- **f** – Administrator session is currently loading the config file
- **L** – Administrator session is currently in Config Mode with the exclusive-lock

The following is sample output of the **show administrators** command indicating current lock mode:

```
[local]asr5500# show administrators
Administrator/Operator Name    M Type    TTY            Start Time
------------------------------ - ------- -------------- ------------------------
Bob                              admin   /dev/pts/2     Tue Mar 29 11:51:15 2016
Alice                          c admin   /dev/pts/1     Mon Mar 28 14:41:15 2016
Carol                            admin   /dev/pts/0     Mon Mar 28 14:40:52 2016
```



CHAPTER 7

Management Settings

This chapter provides instructions for configuring Object Request Broker Element Management (ORBEM) and Simple Network Management Protocol (SNMP) options.

This chapter includes the following sections:

SNMP MIB Browser

This section provides instructions to access the latest Cisco Starent MIB files using a MIB Browser. An updated MIB file accompanies every StarOS release. For assistance to set up an account and access files, please contact your Cisco sales or service representative for additional information.

A MIB Browser allows the user to pull out data from SNMP enabled devices. You can load standard and propriety MIBs. The tool allows the user to see the MIB data in a readable format and also offers the ability to search for a specific OID. The Browser displays all of the MIBs in a MIB tree which makes it easy to find and identify all Objects, Traps or Conformances.

Use the following procedure to view the SNMP MIBs for a specific StarOS build :

Step 1 Contact Cisco sales or a service representative, to obtain access to the MIB files for a specific StarOS release.

Step 2 Download the compressed companion file to a folder on your desktop. The file name follows the convention: **companion_**xx.x.x**.tgz**

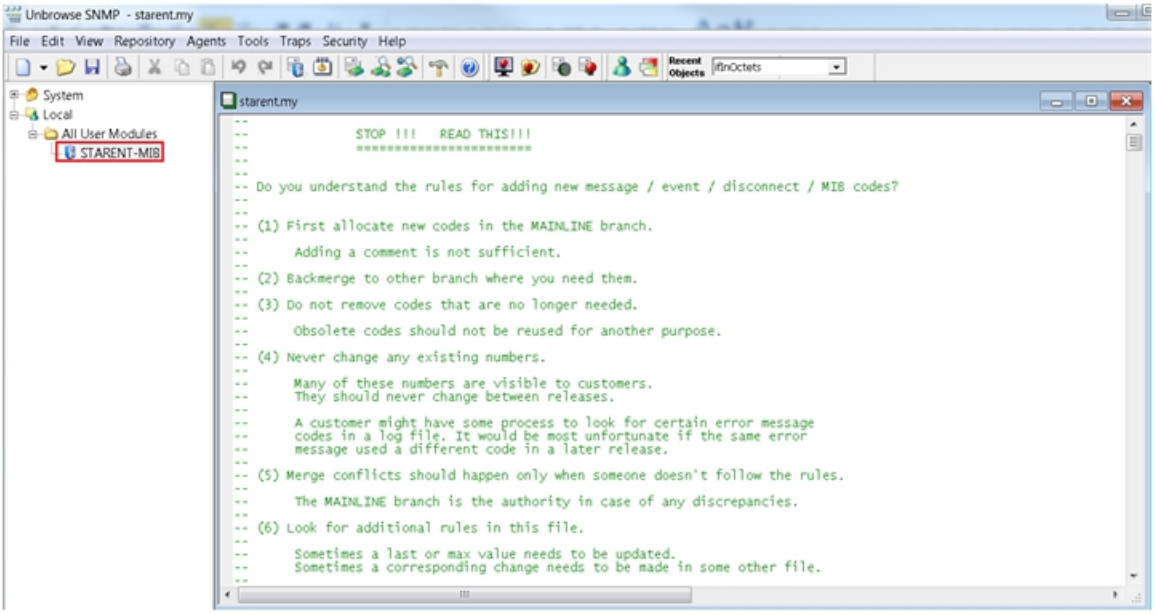
Step 3 Open the companion file, unzip it and extract it to the same folder.

Step 4 Double click on the new **companion-**xx.x.x.xxxxx file folder.

Step 5 Unzip and extract the **companion-**xx.x.x.xxxxx**.tar** file.

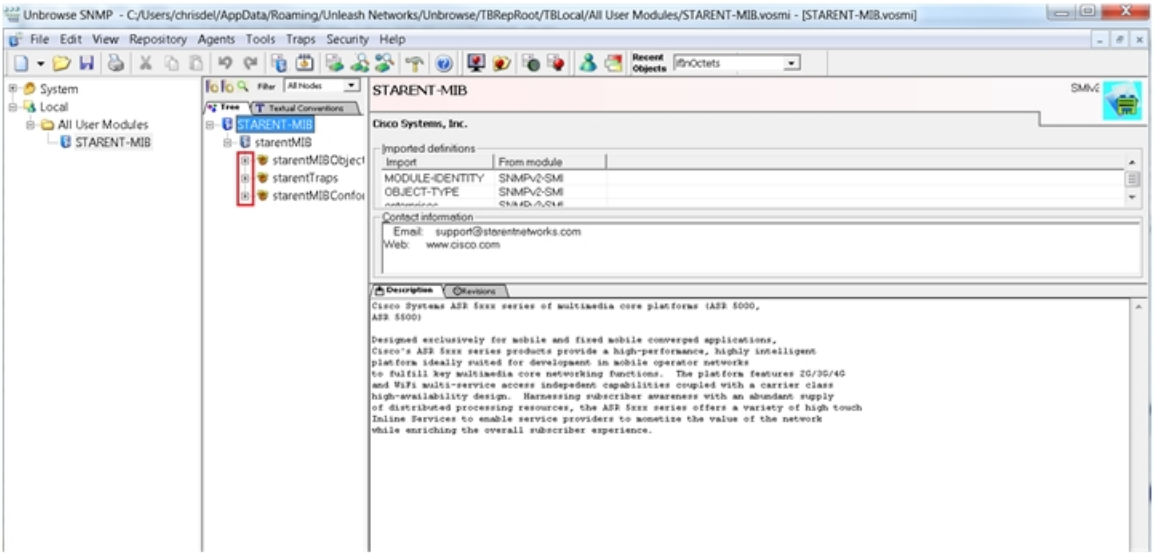
Step 6 From your MIB browser, search for and open the **starent.my** file within the .tar file. You can use any SNMP MIB Browser that allows you to compile a MIB **.my** file before viewing it.

Step 7 To compile the MIB file, click on the STARENT-MIB file and select **File** > **Open**.



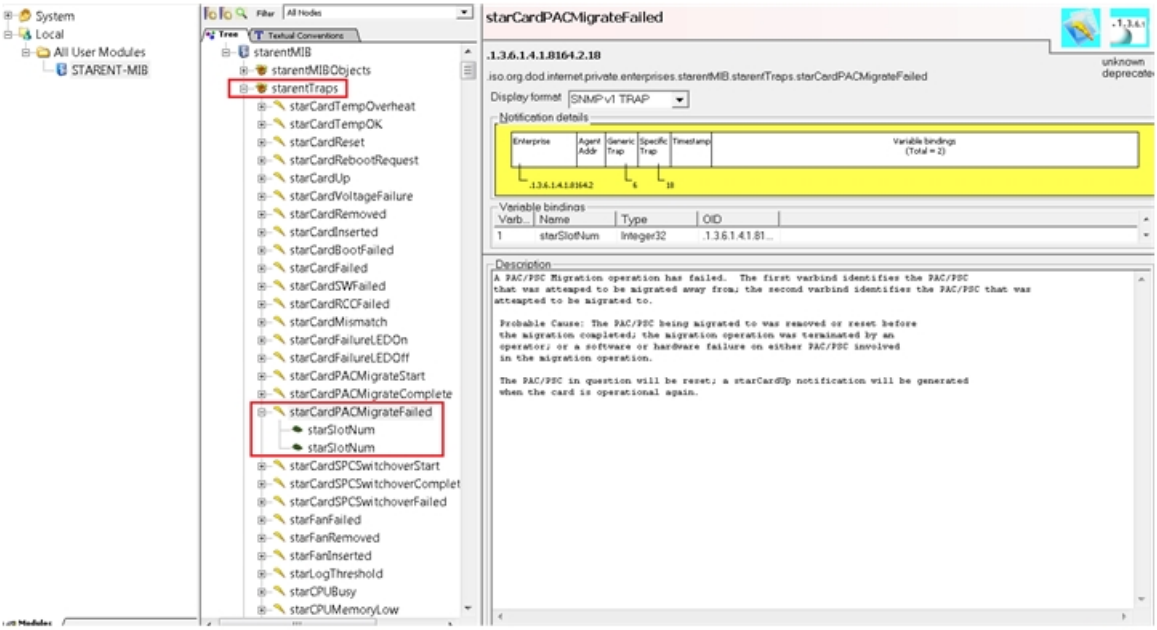
335998

The STARENT-MIB.vosmi file opens.



335999

In the example below the MIB Browser presents a tree diagram that allows you to display details for each Object, Trap and Conformance. The example below includes the OID number and trap details for the starCardPACMigrateFailed trap.



The SNMP MIB browser allows you to search for specific MIBs. You can search for a specific OID (object identifier) to find a specific MIB entry.

Important For information on SNMP MIBs changes for a specific release, refer to the *SNMP MIB Changes in Release xx* chapter of the appropriate version of the to the *Release Change Reference*.

SNMP Support

The system uses the SNMP to send traps or events to the EMS server or an alarm server on the network. You must configure SNMP settings to communicate with those devices.

Important Commands used in the configuration samples in this section provide base functionality. The most common commands and keyword options are presented. In many cases, other optional commands and keyword options are available. Refer to the *Command Line Interface Reference* for complete information.

The *SNMP MIB Reference* describes the MIBs and SNMP traps supported by the StarOS.

To configure the system to communicate with the EMS server or an alarm server:

Step 1 Set SNMP parameters such as UDP port, and alarm server target by applying the example configuration in Configuring SNMP and Alarm Server Parameters, on page 134

Step 2 To view your new SNMP configuration, follow the steps in Verifying SNMP Parameters, on page 135

Step 3 Save the configuration as described in *Verifying and Saving Your Configuration*.

Configuring SNMP and Alarm Server Parameters

Use the following example to set SNMP and alarm server parameters:

```
configure
  system contact contact_name
  system location location_name
  snmp authentication-failure-trap
  snmp community community_string
  snmp server port port_number
  snmp target name ip_address
  snmp engine-id local id_string
  snmp notif-threshold value low low_value period time_period
  snmp user user_name
  snmp mib mib_name
  snmp runtime-debug [ debug-tokens token_id token_id token_id...token_id
  end
```

Notes:

- The **system contact** is the name of the person to contact when traps are generated that indicate an error condition.
- An **snmp community** string is a password that allows access to system management information bases (MIBs).
- The system can send SNMPv1, SNMPv2c, or SNMPv3 traps to numerous target devices. However, an EMS may only process SNMP version 1 (SNMPv1) and SNMP version 2c (SNMPv2c) traps. If the SNMP target you are configuring is the EMS application, use the **snmp target** command to configure use of version 1 or version 2c. Issue this command as many times as you need to configure multiple targets. If you configure multiple targets, generated alarms are sent to every configured target.
- The **snmp notif-threshold** command configures the number of SNMP notifications that need to be generated for a given event and the number of seconds in the monitoring window size (default = 300), before the notification is propagated to the SNMP users (default = 300).
- The **snmp engine-id local** command is optional. It is only required if your network requires SNMP v3 support. The engine ID uniquely identifies the SNMP engine and associated SNMP entities, thus providing a security association between the two for the sending and receiving of data.
- The **snmp user** name is for SNMP v3 and is optional. There are numerous keyword options associated with this command.
- Use the **snmp mib** command to enable other industry standard and Cisco MIBs. By default only the STARENT-MIB is enabled.
- By default SNMP runtime debugging always runs and consumes CPU cycles for event logging. To control CPU usage you can set **no snmp runtime-debug** to disable runtime debugging. An option to this command allows you to specify SNMP token values that will locate and parse specified MIBs.

Important SNMPv3 traps may not be supported by some EMS applications.

Verifying SNMP Parameters

Step 1 Run the **show snmp server** command to verify that the SNMP server information is correctly configured. The following displays a sample output of this command.

```
SNMP Server Configuration:
    Server State             : enabled
    SNMP Port                : 161
    sysLocation              : chicago
    sysContact               : admin
    authenticationFail traps : Enabled
    EngineID                 : 123456789
    Alert Threshold          : 100 alerts in 300 seconds
    Alert Low Threshold      : 20 alerts in 300 seconds
SNMP Agent Mib Configuration:
                           STARENT-MIB : Enabled
                                IF-MIB : Disabled
                            ENTITY-MIB : Disabled
                      ENTITY-STATE-MIB : Disabled
                    ENTITY-SENSORE-MIB : Disabled
                    HOST-RESOURCES-MIB : Disabled
     CISCO-MOBILE-WIRELESS-SERVICE-MIB : Disabled
              CISCO-ENTITY-DISPLAY-MIB : Disabled
                     CISCO-PROCESS-MIB : Disabled
          CISCO-ENTITY-FRU-CONTROL-MIB : Disabled
```

Step 2 Verify that the SNMP community(ies) were configured properly by entering the following command:

show snmp communities

The output of this command lists the configured SNMP communities and their corresponding access levels.

Step 3 Verify that the SNMP transports are configured properly by entering the following command:

show snmp transports

The following displays a sample output:

```
Target Name:   rms1
IP Address:    192.168.1.200
Port:          162
Default:       Default
Security Name: public
Version:       1
Security:
View:
Notif Type:    traps
```

Controlling SNMP Trap Generation

The system uses SNMP traps (notifications) to indicate that certain events have occurred. By default, the system enables the generation of all traps. However, you can disable individual traps to allow only traps of a certain type or alarm level to be generated. This section provides instructions for disabling/enabling SNMP traps.

Important Commands used in the configuration samples in this section provide base functionality. The most common commands and keyword options are presented. In many cases, other optional commands and keyword options are available. Refer to the *Command Line Interface Reference* for complete information regarding all commands.

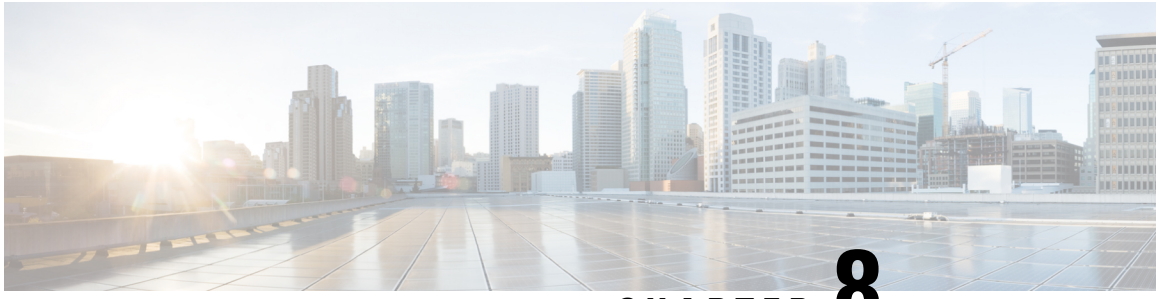
To configure SNMP trap generation:

Step 1 Set parameters by applying the following example configuration:

```
configure
  snmp trap suppress
  snmp trap suppress trap_name1 trap_name2 ... trap_nameN
```

If at a later time you wish to re-enable a trap that was previously suppressed, use the **snmp trap enable** command.

Step 2 Save the configuration as described in *Verifying and Saving Your Configuration*.



CHAPTER 8

Verifying and Saving Your Configuration

This chapter describes how to save your system configuration.

Verifying the Configuration

You can use a number of commands to verify the configuration of your feature, service, or system. Many are hierarchical in their implementation and some are specific to portions of or specific lines in the configuration file.

Feature Configuration

In many configurations, you have to set and verify specific features. An example includes IP address pool configuration. Using the example below, enter the listed commands to verify proper feature configuration.

Enter the **show ip pool** command to display the IP address pool configuration. The output from this command should look similar to the sample shown below. In this example, all IP pools were configured in the *isp1* context.

```
context : isp1:
+-----Type:        (P) - Public        (R) - Private
|                  (S) - Static        (E) - Resource
|
|+----State:       (G) - Good          (D) - Pending Delete      (R)-Resizing
||
||++--Priority:    0..10 (Highest      (0) .. Lowest (10))
||||
||||+-Busyout:     (B) - Busyout configured
||||||
vvvvvv  Pool Name           Start Address  Mask/End Address Used      Avail
----- --------------------- -------------- ---------------  ----------------
PG00    ipsec               12.12.12.0       255.255.255.0   0         254
PG00    pool1               10.10.0.0        255.255.0.0     0         65534
SG00    vpnpool             192.168.1.250    92.168.1.254    0         5

Total Pool Count: 5
```

Important To configure features on the system, use the **show** commands specifically for these features. Refer to the *Exec Mode show Commands* chapter in the *Command Line Interface Reference* for complete information.

Service Configuration

Verify that your service was created and configured properly by entering the following command:

show *service_type service_name*

The output is a concise listing of the service parameter settings similar to the sample displayed below. In this example, a P-GW service called *pgw* is configured.

```
Service name                        : pgw1
Service-Id                                : 1
Context                                   : test1
Status                                    : STARTED
Restart Counter                           : 8
EGTP Service                              : egtp1
LMA Service                               : Not defined
Session-Delete-Delay Timer                : Enabled
Session-Delete-Delay timeout              : 10000(msecs)
PLMN ID List                              : MCC: 100, MNC: 99
Newcall Policy                            : None
```

Context Configuration

Verify that your context was created and configured properly by entering the **show context name** *name* command.

The output shows the active context. Its ID is similar to the sample displayed below. In this example, a context named *test1* is configured.

```
Context Name     ContextID          State
------------       ---------          -----
test1              2                  Active
```

System Configuration

Verify that your entire configuration file was created and configured properly by entering the **show configuration** command.

This command displays the entire configuration including the context and service configurations defined above.

Finding Configuration Errors

Identify errors in your configuration file by entering the **show configuration errors** command.

This command displays errors it finds within the configuration. For example, if you have created a service named "service1", but entered it as "srv1" in another part of the configuration, the system displays this error.

You must refine this command to specify particular sections of the configuration. Add the **section** keyword and choose a section from the help menu as shown in the examples below.

show configuration errors section ggsn-service

or

show configuration errors section aaa-config

If the configuration contains no errors, an output similar to the following is displayed:

```
##########################################################################
Displaying Global
AAA-configuration errors
##########################################################################
Total 0 error(s) in this section !
```

Synchronizing File Systems

Whenever changes are made to a configuration or StarOS version boot order on the active CF, the file systems must be synchronized with the standby CF. This assures that the changes are identically maintained across the management cards.

Enter the following Exec mode command to synchronize the local file systems:

[local]host_name# **filesystem synchronize all**

The **filesystem** command supports multiple keywords that allow you to check for and repair file system corruption, as well as synchronize a file system with a specific storage device. For additional information, see the *Exec Mode Commands* chapter in the *Command Line Interface Reference*.

Saving the Configuration

These instructions assume that you are at the root prompt for the Exec mode:

[local]*host_name*#

To save your current configuration, enter the following command:

save configuration *url* **[obsolete-encryption | showsecrets | verbose] [-redundant] [-noconfirm]**

url specifies the location in which to store the configuration file. It may refer to a local or a remote file.

Important Do not use the "/" (forward slash), ":" (colon) or "@" (at sign) characters when entering a string for the following URL fields: directory, filename, username, password, host or port#.

Important The **-redundant** keyword saves a configuration file to the standby CF virtual machine. This command does not synchronize the local file system. If you have added, modified, or deleted other files or directories to or from a local device for the active CF VM, you must synchronize the local file system on both CF VMs. See Synchronizing File Systems, on page 139.

Important The **obsolete-encryption** and **showsecrets** keywords have been removed from the **save configuration** command in StarOS 19.2 and higher. If you run a script or configuration that contains the removed keyword, a warning message is generated.

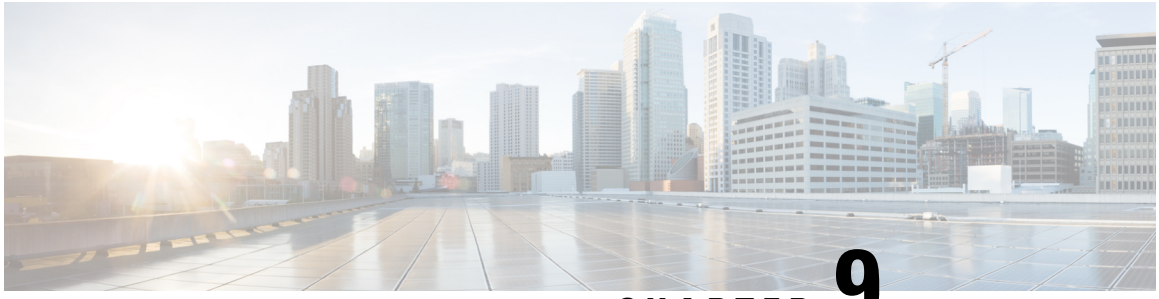
Note Although usb1 and usb2 keyword options are available in this command, this options are only available if the devices have been configured for the server via the hypervisor. This involves creating a virtual controller and specifying the available devices.

The recommended procedure is to save VPC configurations to an external network device.

For complete information about the above command, see the *Exec Mode Commands* chapter of the *Command Line Interface Reference*.

To save a configuration file called *system.cfg* to a directory that was previously created called *cfgfiles*, enter the following command:

```
save configuration /flash/cfgfiles/system.cfg
```



CHAPTER 9

System Interfaces and Ports

This chapter describes how to create a context and configure system interfaces and ports within the context. Before beginning these procedures, refer to your product-specific administration guide for configuration information for your product.

Contexts

Even though multiple contexts can be configured to perform specific functions, they are all created using the same procedure.

Creating Contexts

Important Commands used in the configuration examples in this section represent the most common or likely commands and/or keyword options. In many cases, other commands and/or keyword options are available. Refer to the *Command Line Interface Reference* for complete information regarding all commands.

To create a context, apply the following example configuration:

```
configure
  context name
    end
```

Repeat to configure additional contexts.

Viewing and Verifying Contexts

Step 1 Verify that your contexts were successfully created by entering the following command:

```
[local]host_name# show context all
```

The output is a two-column table similar to the example below. This example shows that two contexts were created: one named *source* and one named *destination*.

```
Context Name    ContextID    State
------------    ---------    -----
local           1            Active
source          2            Active
destination     3            Active
```

The left column lists the contexts that are currently configured. The center column lists the corresponding context ID for each of the configured contexts. The third column lists the current state of the context.

Step 2 Save your configuration as described in the *Verifying and Saving Your Configuration* chapter.

Step 3 Now that the context has been created, interfaces and specific functionality can be configured within the context. Proceed to other sections for instructions on configuring specific services and options.

Ethernet Interfaces and Ports

Regardless of the type of application interface, the procedure to create and configure it consists of the following:

Step 1 Create an interface and assign an IP address and subnet mask to it by applying the example configuration in Creating an Interface, on page 142.

Step 2 Assign a physical port for use by the interface and bind the port to the interface by applying the example configuration in Configuring a Port and Binding It to an Interface, on page 143.

Step 3 Optionally configure a static route for the interface by applying the example configuration in Configuring a Static Route for an Interface, on page 143.

Step 4 Repeat the above steps for each interface to be configured.

This section provides the minimum instructions for configuring interfaces and ports to allow the system to communicate on the network. Commands that configure additional interface or port properties are described in the *Ethernet Port Configuration Mode Commands* and *Ethernet Interface Configuration Mode Commands* chapters of the *Command Line Interface Reference.*

To ensure that system line card and port-level redundancy mechanisms function properly, the Spanning Tree protocol must be disabled on devices connected directly to any system port. Failure to turn off the Spanning Tree protocol may result in failures in the redundancy mechanisms or service outage.

Creating an Interface

Use the following example to create a new interface in a context:

```
configure
   context name
      interface name
         { ip | ipv6 } address address subnetmask [ secondary ]
         end
```

Notes:

- *Optional:* Add the **loopback** keyword option to the **interface** *name* command, to set the interface type as "loopback" which is always UP and not bound to any physical port.
- *Optional:* Add the **secondary** keyword to the **{ ip | ipv6 } address** command, to assign multiple IP addresses to the interface. IP addresses can be entered using IPv4 dotted-decimal or IPv6 colon-separated-hexadecimal notation.
- *Optional:* In the interface config mode, add the **port-switch-on-L3-fail address** command, to configure the interface for switchover to the port on the redundant line card if connectivity to a specified IP address is lost. This IP address can be entered using IPv4 dotted-decimal or IPv6 colon-separated-hexadecimal notation.

Configuring a Port and Binding It to an Interface

Use the following example configuration to configure and assign a port to an interface:

```
configure
  port ethernet slot#/port#
    description description
    no shutdown
    bind interface interface_name  context_name
    end
```

Notes:

- For **port ethernet** *slot#*, the slot corresponds to a CF or SV VM within the virtual chassis. The hypervisor assigns a unique slot number to each VM during initial configuration of the VPC-DI instance. Slots 1 and 2 are assigned to the CFs; slot number 3 through 32 are assigned to SFs.
- For **port ethernet** *port#*, the CF only supports port 1; each SF supports 12 vNICs number 1 through 12 with corresponding virtual Ethernet ports numbered 10 through 21. SF port number 10 must be configured.

  Note StarOS supports up to 12 service ports, but the actual number of ports may be limited by the hypervisor.

- *Optional:* In the Ethernet Port configuration mode, add the preferred **slot** *slot#* command if you want to specify a port preference.
- Binding associates the port and all of its settings to the named interface.

Configuring a Static Route for an Interface

Use the following example to configure a static route for an interface:

```
configure
  context name
      { ip | ipv6 } route ip_address netmask next-hop gw_address interface_name
    end
```

Notes:

- *ip_address* and *netmask* are the IP address and subnet mask of the target network. This IP address can be entered using IPv4 dotted-decimal or IPv6 colon-separated-hexadecimal notation.
- *gw_address* is the IP address of the default gateway or next-hop route. This IP address can be entered using IPv4 dotted-decimal or IPv6 colon-separated-hexadecimal notation.
- To configure a route to the gateway router, use 0.0.0.0 for the network and mask variables.
- Repeat as needed. Multiple static routes can be configured to the same destination to provide an alternative means of communication in case the preferred route fails.

Viewing and Verifying Port Configuration

Step 1 Verify that your interface configuration settings are correct by entering the following commands:

```
[local]host_name# context context_name
[context_name]host_name# show { ip | ipv6 } interface
```

context_name represents the name of the context in which the interface was created. The output from these commands should be similar to the following example.

In this example an interface named *mgmt1* was configured in the local context.

Example:

In this example an interface named *mgmt1* was configured in the local context.

```
Intf Name:       mgmt1
Intf Type:       Broadcast
IP State:        UP (Bound to 10/11 untagged, ifIndex 285278209)
IP Address:      192.168.100.3        Subnet Mask:     255.255.255.0
Bcast Address:   192.168.100.255      MTU:          1500
Resoln Type:     ARP                  ARP timeout:     3600 secs
Number of Secondary Addresses:  0
Total interface count:         1
```

Step 2 Verify that your port configuration settings are correct by entering the following command:

```
[context_name]host_name# show configuration port slot#/port#
```

slot is the chassis slot number. The slot corresponds to a CF or SF VM within the virtual chassis. The hypervisor assigns a unique slot number to each VM during initial configuration of the VPC-DI instance. Slots 1 and 2 are assigned to the CF VMs; slot numbers 3 through 32 are assigned to SF VMs. CF only supports port 1. Each SF supports four vNICs numbered 1 through 4 with corresponding virtual ethernet ports numbered 10 through 14. SF port number 10 must be configured.

Example:

This command produces an output similar to that displayed in the following example that shows the configuration for port 10 for the SF in slot 3. In this example, the port is bound to an interface called *sf3* configured in a context called *source*.

```
config
   port ethernet 3/10
      description 10_SF3
      no shutdown
      bind interface sf3 source
```

```
end
```

slot is the chassis slot number, aways slot 1.

Step 3 Verify that your static route(s) was configured properly by entering the following command:

```
[context_name]host_name# show ip static-route
```

Example:

This command produces an output similar to that displayed in the following example that shows a static route to a gateway with an IP address of 192.168.250.1.

```
Destination      Nexthop         Protocol  Prec  Cost  Interface
0.0.0.0/0        192.168.250.1    Static      0    0     vNIC1
0.0.0.0/0        192.168.250.1    Static      0    0     sf3 source
```

Step 4 Save the configuration as described in the *Verifying and Saving Your Configuration* chapter.

VLANs

Virtual LANs (VLANs) allow two logically separated networks to use the same physical medium. VLAN segmentation, also called 802.1q tagging, works by appending a tag identifying the VLAN ID to each Ethernet frame.

For information on how to create VLANs to handle specific packet types, see the *VLANs* chapter.

Hypervisors

VLAN usage under KVM is an extension to bridge interface sharing. The difference lies in which interface participates in the bridge set. The physical interfaces (such as eth0, eth1) are bound to the bridge, which is used by each guest. These interfaces carry unmodified packets coming externally or being generated internally, with or without a VLAN ID tag.

VMware supports the use of virtual switches that allow virtual machines on one vSphere host to communicate with each other using the same protocols as physical switches. The vSwitch emulates a traditional physical Ethernet network switch by forwarding frames at the data-link layer. A vSphere host can have numerous virtual switches, each with more than 1,000 internal virtual ports for virtual machines. The vSphere platform supports the vSphere Standard Switch virtual switch configuration at the host level and the vSphere Distributed Switch, a single virtual switch that spans multiple associated hosts.

VLANs and Management Ports

The management interface supports VLAN configuration. This support extends to the local context.

Bulkstats can be sent out an interface other than the normal management interface. This interface also supports VLANs.

You can also configure other OA&M services on separate VLANs.

Yo can assign separate source IP addresses for the OA&M services. OA&M services should not be bound to the same VLAN as service VLANs. Other services include SGi, Gi, Pi, eGTP or other packet core-specific interfaces and services.

This feature is implemented by adding support for the **vlan** command to the management port in the local context. See the example command sequence below.

```
configure
   port ethernet 1/1
      vlan 184
         no shutdown
         bind interface 19/3-UHA foo
```

CHAPTER 10

System Security

This chapter describes the StarOS security features.

This chapter explores the following topics:

Protection of Passwords

Users with privilege levels of Inspector and Operator cannot display decrypted passwords in the configuration file via the command line interface (CLI).

Secure Password Encryption

By default for StarOS releases prior to 21.0 the system encrypts passwords using an MD5-based cipher (option A). These passwords also have a random 64-bit (8-byte) salt added to the password. The chassis key is used as the encryption key.

Setting a chassis key supports an encryption method where the decryption requires the knowledge of a "shared secret". Only a chassis with knowledge of this shared secret can access the passwords. To decipher passwords, a hacker who knew the chassis key would still need to identify the location of the 64-bit random salt value within the encryption.

Passwords encrypted with MD-5 will have "+A" prefixes in the configuration file to identify the methodology used for encrypting.

Important For release 21.0 and higher, the default is Algorithm B.

For release 15.0 and higher, another type of encryption algorithm can be specified. The Global Configuration mode **cli-encrypt-algorithm** command allows an operator to configure the password/secret encryption algorithm. The default encryption/password algorithm for releases prior to 21.0 is MD-5 as described above (option A). A second password encryption algorithm (option B) uses AES-CTR-128 for encryption and HMAC-SHA1 for authentication. The encryption key protects the confidentiality of passwords, while the authentication key protects their integrity. For release 21.0 and higher Algorithm B is the default. Passwords encrypted with this key will have "+B" prefixes in the configuration file.

For release 19.2 and higher, a third type of encryption algorithm can be specified (option C). This algorithm specifies the use of the HMAC-SHA512 cipher algorithm for encryption and authentication. Passwords encrypted with this key will have "+C" prefixes in the configuration file.

Also for release 19.2 and higher, the encryption key is hashed from the chassis ID and a 16-byte Initialization Vector (IV) obtained from an internal random number generator. No two passwords are encrypted using the same encryption key/IV pair. The Security Administrator must set a chassis key in order to generate the chassis ID and resulting encryption key. A default chassis key based on a local MAC address is no longer supported.

The syntax for the **cli-encrypt-algorithm** command is:

```
config
   cli-encrypt-algorithm { A | B | C }
```

Support for Non-Current Encryptions and Decryptions

The system supports previously formatted encrypted passwords. The syntax of the encrypted passwords indicates which methodology was used for encryption. If the system does not see a prefix before the encrypted password, the earlier encryption method using a fixed key will be used. If the encrypted password includes the "+A" prefix, the decryption method uses the chassis key and random salt.

If the user saves a new configuration, the generated file will always contain passwords encrypted by the most recent method. The user cannot generate the earlier DES-based encryption values. However, all future StarOS releases will continue to support plain-text password entry for all two-way encryptable passwords

The recommended process for changing the chassis key without causing a "lock-out" state is as follows:

- Load the configuration file of the last good configuration using the previous chassis key.
- Change the chassis key to the new desired value.
- Save the configuration with this new chassis key.

Refer to *Configuring a Chassis Key* in *System Settings* for additional information.

Support for ICSR Configurations

Inter-Chassis Session Recovery (ICSR) is a redundancy configuration that employs two identically configured VPC-DI chassis/instances as a redundant pair.

ICSR pairs share the same chassis key. If the ISCR detects that the two chassis/instances have incompatible chassis keys, an error message is logged but the ICSR system will continue to run. Without the matching chassis key, the standby ICSR peer can recover services if the active peer goes out of service; the standby peer will still have access to the passwords in their decrypted form.

ICSR peers use Service Redundancy Protocol (SRP) to periodically check to see if the redundancy configuration matches with either decrypted passwords or DES-based two-way encryption strings. Since the configuration is generated internally to the software, users are not able to access the configuration used to check ICSR compatibility.

Encrypted SNMP Community Strings

Simple Network Management Protocol (SNMP) uses community strings as passwords for network elements. Although these community strings are sent in clear-text in the SNMP PDUs, the values can be encrypted in the configuration file.

The **snmp community encrypted name** command enables the encryption of SNMP community strings. For additional information, see the *Global Configuration Mode Commands* chapter in the *Command Line Interface Reference*.

Enhanced Password Security

Lawful Intercept Restrictions

This section describes some of the security features associated with the provisioning of Lawful Intercept (LI).

LI Server Addresses

An external authenticating agent (such as RADIUS or Diameter) sends a list of LI server addresses as part of access-accept. For any intercept that was already installed or will be installed for that subscriber, a security check is performed to match the LI server address with any of the LI-addresses that were received from the authenticating agent. Only those addresses that pass this criteria will get the intercepted information for that subscriber.

While configuring a campon trigger, the user will not be required to enter the destination LI server addresses. When a matching call for that campon trigger is detected, a security check is done with the list received from the authentication agent. The LI-related information is only forwarded if a matching address is found.

When an active-only intercept is configured, if a matching call is found, a security check is made for the LI address received from the authentication agent and the intercept configuration will be rejected.

If no information related to LI server addresses is received for that subscriber, LI server addresses will not be restricted.

Important A maximum of five LI server addresses are supported via an authenticating agent.

Important The ability to restrict destination addresses for LI content and event delivery using RADIUS attributes is supported only for PDSN and HA gateways.

Modifying Intercepts

One LI administrator can access and/or modify the intercepts created by another LI administrator. Whenever an intercept is added, removed or modified, an event log is displayed across LI administrators about the change. An SNMP trap is also generated.

Adding, Modifying and Removing Users

It is considered uncommon for a user to be added or removed from the system. Likewise, it is considered uncommon for a user's privileges to modified. However, if the system is compromised, it is common for attackers to add or remove a privileged user, raise their privileges or lower the privileges of others.

As a general rule, lower privileged users should not be allowed to increase their privileges or gain access to sensitive data, such as passwords, which were entered by higher privileged users.

Important The system can only detect changes in users and user attributes, such as privilege level, when these users are configured through the system.

Notification of Users Being Added or Deleted

Users with low level authorization should not be able to create users with high level authorization. However, if a malicious actor were to be able to create a high level authorized user, they could then delete the other high level authorized users, thereby locking them out of the system.

The following SNMP traps notify an administrator when users are added or removed:

- **starLocalUserAdded** – indicates that a new local user account has been added to the system.
- **starLocalUserRemoved** – indicates that a local user account has been removed from the system.

Notification of Changes in Privilege Levels

Whenever a user's privilege level is increased or decreased, an SNMP notification will be sent out. A malicious actor may gain access to more privileged commands by somehow promoting" their privileges. Once this is done, they could then "demote" the privileges of all the other users, thereby locking the proper administrators out of the system.

The **starLocalUserPrivilegeChanged** trap indicates that a local user's privilege level has been changed.

User Access to Operating System Shell

The **starOsShellAccessed** trap indicates that a user has accessed the operating system shell.

Test-Commands

Users with Security Administrator or Administrator privilege can enable the display of previously hidden test-commands. The CLI test-commands mode displays new command keywords for existing commands, as well as new commands.

Caution CLI test-commands are intended for diagnostic use only. Access to these commands is not required during normal system operation. These commands are intended for use by Cisco TAC personnel only. Some of these commands can slow system performance, drop subscribers, and/or render the system inoperable.

Enabling cli test-commands Mode

To enable access to test-commands, a Security Administrator must log into the Global Configuration mode and enter **cli hidden**.

This command sequence is shown below.

```
[local]host_name# config
[local]host_name(config)# cli hidden
[local]host_name(config)#
```

By default **cli hidden** is disabled.

Important Low-level diagnostic and test commands/keywords will now be visible to a user with Administrator or higher privilege. There is no visual indication on the CLI that the test-commands mode has been enabled.

Enabling Password for Access to CLI-test commands

A Security Administrator can set a plain-text or encrypted password for access to CLI test commands. The *password* value is stored in **/flash** along with the boot configuration information. The **show configuration** and **save configuration** commands will never output this value in plain text.

The Global Configuration mode command **tech-support test-commands [encrypted] password** *new_password* **[old-password** *old_password* **]** sets an encrypted or plain-text password for access to CLI test-commands.

This command sequence is shown below.

```
[local]host_name# config
[local]host_name(config)# tech-support test-commands password new_password [
old-password old_password ]
[local]host_name(config)#
```

If the new password replaces an existing password, you must enter the old password for the change to be accepted.

If the old password is not entered or does not match the existing configured value, the following error message appears: "tech-support password is already configured". A prompt then appears to accept entry of the old password: "Enter old tech-support password:".

Entering **old-password** *old_password* allows you to replace the existing password without being prompted to enter the old password. If you incorrectly enter the old password or do not enter the old password, an error message appears: "Failure: Must enter matching old tech-support password to replace existing password".

The Quick Setup Wizard (Exec mode **setup** command) also prompts for entry of a tech-support test-commands password. If you have forgotten the old tech-support password, you can run **setup** directly from the Console port to enter a new tech-support password.

When a test-commands password is configured, the Global Configuration mode command **cli test-commands [encrypted] password** *password* requires the entry of the password keyword. If the **encrypted** keyword is specified, the *password* argument is interpreted as an encrypted string containing the password value. If the **encrypted** keyword is not specified, the *password* argument is interpreted as the actual plain text value

Important If **tech-support test-commands password** is never configured, StarOS will create a new password. If the **password** keyword is not entered for **cli test-commands**, the user is prompted (no-echo) to enter the password. Also, **cli hidden** must be enabled by an administrator to access the CLI test-commands.

Exec Mode cli test-commands

Exec mode commands are available to a privileged user who enters the command **cli test-commands** from Exec mode.

```
[local]host_name# cli test-commands [encrypted] password password
Warning: Test commands enables internal testing and debugging commands
USE OF THIS MODE MAY CAUSE SIGNIFICANT SERVICE INTERRUPTION
```

Important An SNMP trap (starTestModeEntered) is generated whenever a user enters CLI test-commands mode.

Configuration Mode cli test-commands

Configuration commands which provided access to low-level software parameters are accessible only after a privileged user enters the command **cli test-commands** from Global Configuration mode.

```
[local]host_name# config
[local]host_name(config)# cli test-commands [encrypted] password password
Warning: Test commands enables internal testing and debugging commands
USE OF THIS MODE MAY CAUSE SIGNIFICANT SERVICE INTERRUPTION
```

Important An SNMP trap (starTestModeEntered) is generated whenever a user enters CLI test-commands mode.

Using COTS Hardware for Encryption

StarOS VPC instances perform encryption and tunneling of packets in the software. If, however, your commercial off-the shelf (COTS) server uses the Intel Communications Chipset 89xx and you configure the VPC virtual machines to passthrough this chipset, then the VPC instances automatically utilize this hardware

chip for encryption and decryption of packets. The Intel Communications Chipset 89xx is also known as Coleto Creek.

Note All service function (SF) VMs must use the Intel Communications chipset in order for the VPC to use the hardware chipset for encryption and decryption.

To determine if your COTS server uses this chipset, use the **show hardware** command to display information for all slots. This example illustrates sample output from the **show hardware** command for a VPC-SI instance on hardware that uses the Coleto Creek crypto accelerator:

```
[local]swch32# show hardware
System Information:
  Platform                : KVM Guest
  UUID/Serial Number      : 014A4D4F-7644-4CF1-C408-8ABB631B3E34
  CPU Packages            : 1 [#0]
  CPU Nodes               : 1
  CPU Cores/Threads       : 16
  Memory                  : 16384M (qvpc-si-medium)
  Crypto Accelerator      : Coleto Creek A0
Storage Devices:
  Virtual Flash           : Present
    Type                  : 4096M disk
    Model                 : ATA-QEMUHARDDISK
    Serial Number         : QM00001
  Hard Drive 1            : Present
    Type                  : 16384M disk
    Model                 : ATA-QEMUHARDDISK
    Serial Number         : QM00002
  Hard Drive 2            : Not Present
  USB 1                   : Not Present
  USB 2                   : Not Present
  CDROM 1                 : Present
    Type                  : cdrom
    Model                 : QEMU-QEMUDVD-ROM
Network Interfaces:
  loeth0   addr 52:54:00:ae:b7:72 at virtio1, 1af4:0001 (virtio_net)
    RxQ(s)/RINGSZ/COALESCE: 1/256/-NA-
    TxQ(s)/RINGSZ/COALESCE: 1/256/-NA-
    NODE-ID               : -NA-
  port1_10 addr 00:1b:21:87:14:ac at 0000:00:06.0, 8086:10fb (ixgbe)
    RxQ(s)/RINGSZ/COALESCE: 16/4096/500
    TxQ(s)/RINGSZ/COALESCE: 16/4096/0
    NODE-ID               : -NA-
  port1_11 addr 00:1b:21:87:14:ad at 0000:00:07.0, 8086:10fb (ixgbe)
    RxQ(s)/RINGSZ/COALESCE: 16/4096/500
    TxQ(s)/RINGSZ/COALESCE: 16/4096/0
    NODE-ID               : -NA-
```

This example illustrates sample output from the **show hardware** command for a VPC-SI instance on hardware that does not have a crypto accelerator installed:

```
[local]swch81# show hardware
System Information:
  Platform                : KVM Guest
  UUID/Serial Number      : E0A26495-F822-4AC0-914D-B51332177C4D
  CPU Packages            : 1 [#0]
  CPU Nodes               : 1
  CPU Cores/Threads       : 16
  Memory                  : 32768M (qvpc-si-medium)
```

```
  Crypto Accelerator      : None
Storage Devices:
  Virtual Flash           : Present
    Type                  : 4096M disk
    Model                 : ATA-QEMUHARDDISK
    Serial Number         : QM00001
  Hard Drive 1            : Present
    Type                  : 16384M disk
    Model                 : ATA-QEMUHARDDISK
    Serial Number         : QM00002
  Hard Drive 2            : Not Present
  USB 1                   : Not Present
  USB 2                   : Not Present
  CDROM 1                 : Present
    Type                  : cdrom
    Model                 : QEMU-QEMUDVD-ROM
Network Interfaces:
  loeth0   addr 52:54:00:e9:70:05 at virtio1, 1af4:0001 (virtio_net)
    RxQ(s)/RINGSZ/COALESCE: 1/256/-NA-
    TxQ(s)/RINGSZ/COALESCE: 1/256/-NA-
    NODE-ID               : -NA-
  port1_10 addr 52:54:00:22:f7:85 at virtio2, 1af4:0001 (virtio_net)
    RxQ(s)/RINGSZ/COALESCE: 8/256/-NA-
    TxQ(s)/RINGSZ/COALESCE: 8/256/-NA-
    NODE-ID               : -NA-
  port1_11 addr 52:54:00:3e:67:f9 at virtio3, 1af4:0001 (virtio_net)
    RxQ(s)/RINGSZ/COALESCE: 8/256/-NA-
    TxQ(s)/RINGSZ/COALESCE: 8/256/-NA-
    NODE-ID               : -NA-
```

Random Number Generator Support for OS and Platforms

Feature Summary and Revision History

Summary Data

Applicable Product(s) or Functional Area	All
Applicable Platform(s)	• VPC-DI • VPC-SI
Feature Default	Disabled - Configuration required
Related Changes in This Release	Not applicable
Related Documentation	• *VPC-DI System Administration Guide* • *VPC-SI System Administration Guide*

Revision History

Important Revision history details are not provided for features introduced before releases 21.2 and N5.1.

Revision Details	Release
First introduced.	21.13

Feature Description

A few of the features deployed on the ASR 5500 and VPC platforms require random numbers for performing certain tasks. While it uses the kernel random number generator for these tasks, the numbers generated may or may not be sufficiently random as per the security standards. However, hardware or host-provided random numbers are considered reliable and meet security standards.

The Random Number Generator Support for OS and Platforms feature addresses this security compliance requirement. It enables the system administrator to configure hardware random number generator (HWRNG) on their host machines.

When configured, the system uses the the hardware random number generators.

Note This feature works only when HWRNG support is available on the host.

When HWRNG support is available, add the following configuration to the `libvirt xml` file on the host. This adds `virtio_rng` support to the client (StarOS).

```
<rng model='virtio'>
      <backend model='random'>/dev/random</backend>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x07' function='0x0'/>
</rng>
```

Note If there is a conflict in using slot number 7 (as shown in the preceding configuration) in the configuration, use the next available slot.

This configuration must be applied on the supported platforms based on the respective deployment configurations.

No configuration changes are required on the client. The client (StarOS) picks up `virtio_rng` automatically if the support is enabled on the host.

CHAPTER 11

Secure System Configuration File

Feature Summary and Revision History

Summary Data

Applicable Product(s) or Functional Area	All
Applicable Platform(s)	ASR 5500 VPC-DI VPC-SI
Feature Default	Disabled
Related Changes in This Release	Not Applicable
Related Documentation	• *ASR 5500 System Administration Guide* • *VPC-DI System Administration Guide* • *VPC-SI System Administration Guide*

Revision History

Revision Details	Release
First Introduced.	21.3

Feature Description

A system configuration file contains crucial configuration information used to setup and operate the operator's network. The configuration file must be properly authenticated before it is loaded to avoid unauthorized changes to the file that could harm the network.

This feature enables the system configuration file to be signed with an RSA key to ensure the integrity and authenticity of the configuration file before it is loaded. The operator can sign the configuration file with a private key, and the system uses a public key to validate the signed configuration file before loading it.

How System Configuration Files are Secured

Create a Digital Signature

The operator can sign the configuration file using the following steps:

1. Perform an SHA512 hash on the configuration file to create a message digest.

   Example (Linux/OpenSSL):

   ```
   openssl dgst -sha512 -binary -out digest cfg_file
   ```

2. Create a digital signature by encrypting the message digest value with the RSA private key.

   Example (Linux/OpenSSL):

   ```
   openssl pkeyutl -sign -in digest -inkey pri_key.pem -out sig \
   -pkeyopt digest:sha512 -pkeyopt rsa_padding_mode:pss \
   -pkeyopt rsa_pss_saltlen:-2
   ```

3. Convert the digital signature to a base64 format (A '#' is added at the beginning, and a new line at the end).

   Example (Linux/OpenSSL):

   ```
   echo -n "#" > sig_base64
   base64 sig -w 0 >> sig_base64
   echo "" >> sig_base64
   ```

4. Append the original configuration file with the digital signature.

   Example (Linux/OpenSSL):

   ```
   cat sig_base64 cfg_file > signed_cfg_file
   ```

Generating the Public and Private Keys

The RSA public key is stored in PEM format (.pem file), and can be generated using one of the following OpenSSL commands in the example below:

```
openssl rsa -in pri_key.pem - pubout -out pub_key.pem
--or--
openssl rsa -in pri_key.pem -RSAPublicKey_out -out pub_key.pem
```

An RSA private key in PEM format can be generated using the OpenSSL command in the following example:

```
openssl genrsa -out pri_key.pem 2048
```

For more information on the **openssl rsa** and **openssl genrsa** commands, refer their respective OpenSSL manual pages.

Validate the Digital Signature

When signature verification is enabled, validation of the digital signature occurs when the system boots up and loads the configuration file (or any time when the config file is loaded). The system determines if signature verification is enabled (or disabled) by looking for the *.enable_cfg_pubkey* file in the secure directory. For more information, refer Enable or Disable Signature Verification, on page 160.

The system validates the signed configuration file using the following steps:

1. Extract the RSA public signing key from the flash.
2. Extract the configuration file's digital signature (the first line).
3. Convert the signature from base64 to binary format.
4. Decrypt the signature using the RSA public key.
5. Calculate the SHA512 hash for the plain config file resulting in a message digest.
6. Compare the decrypted signature value and newly calculated message digest. If they match, the configuration file is successfully validated.

Configuring Signature Verification

Import RSA Public Key for Verification

To verify the signed configuration file, an RSA public key (in PEM format) must be imported. Use the following command to import the RSA public key:

Important This command can only be executed from the console.

cfg-security import public-key url *url_address*

Notes:

- Any existing .pem file will be replaced with the new .pem file when the command is executed.
- *url_address* may refer to a local or a remote file, and must be entered using the following format:

  [file:]{/flash | /usb1 | /hd-raid | /sftp}[/*directory***]**/*filename*

  tftp://*host***[:***port***][**/<*directory*>**]**/*filename*

  ftp://**[***username***[:***password***]@]***host***[:***port***][**/*directory***]**/*filename*

  sftp://**[***username***[:***password***]@]***host***[:***port***][**/*directory***]**/*filename*

  http://**[***username***[:***password***]@]***host***[:***port***][**/*directory***]**/*filename*

  https://**[***username***[:***password***]@]***host***[:***port***][**/*directory***]**/*filename*

Enable or Disable Signature Verification

Use the following command to enable (or disable) signature verification in the configuration file:

Important This command can only be executed from the console.

```
[ no ] cfg-security sign
```

Notes:

- Enabling signature verification (**cfg-security sign** command) will create an empty file named *.enable_cfg_pubkey* in the same directory where the PEM file exists.
- Use the **no cfg-security sign** command to disable verification of signature in the configuration file. Disabling signature verification (**no cfg-security sign** command) will remove the *.enable_cfg_pubkey* file.
- The system looks for the *.enable_cfg_pubkey* file to determine if signature verification is enabled or disabled.

CHAPTER 12

Software Management Operations

This chapter provides information about software management operations on the system.

Understanding the Local File System

The local file system on the VPC VM is made up of files that are stored on the following:

- **/flash** Flash memory allocated as vHDD-1 on the M via the hypervisor is the default storage media for the StarOS image, CLI configuration, and crash log files used by the system.
- **/hd-raid** This is the storage space allocated as vHDD-2 on the CF VM by the hypervisor. It is used to store CDRs (Charging Data Records) and UDRs (Usage Data Records).

File Types Used by the Local File System

The following file types can be located in the local file system:

- **Operating System Software Image File**: This binary file type is identified by its **.bin** extension. The file is the operating system that is loaded by the system upon startup or reloading. This is an executable, read-only file that cannot be modified by end users.
- **CLI Configuration File**: This file type is identified by its **.cfg** extension. These are text files that contain CLI commands that work in conjunction with the operating system software image. These files determine services to be provided, hardware and software configurations, and other functions performed by the system. The files are typically created by the end user. You can modify the files both on and off-line and use descriptive long filenames.
- **System File**: Only one file identified by a **.sys** extension is used by the system. The boot.sys file contains system-specific information, which describes how the system locates, and in what priority it loads, file groups (paired .bin and .cfg files) from its boot stack.

- **Abridged Crash Log**: The abridged crash log, identified by its **crashlog** filename, contains summary information about software or hardware failures that occur on the system. This file is located in the **/flash/crsh2/** directory on the device. You can view the contents of this file through the CLI, but you cannot modify the file.

Understanding the boot.sys File

The system uses the boot.sys file to store the prioritized boot stack parameters and file groups the system uses during startup. Modify this file only through system CLI commands and not through external means. Boot parameters contain information the system needs to locate the operating system image file, including:

- **bootmode**: This setting is typically configured to normal, and identifies how the system starts.
- **boot stack information**: The boot stack is made up of prioritized file group entries that designate the operating system image file and the CLI configuration file to load.

When a system is started for the first time, the boot.sys file is configured to use the normal boot mode and load the operating system software image from the /flash directory.

There is no CLI configuration file contained on the local file system. This causes the system to automatically start its CLI-based Quick Setup Wizard upon the first successful boot. Refer to *Getting Started* for more information on using the Quick Setup Wizard.

Maintaining the Local File System

Use CLI commands to manage and maintain the devices that make up the local file system. Execute all the commands described in this section in the Exec Mode. Unless otherwise specified, you must have security administrator or administrator privileges to execute these commands.

Note You must be logged into the active CF VM to run the commands described below.

File System Management Commands

Use the commands in this section to manage and organize the local file system.

Important For complete information on the commands listed below, see the *Exec Mode Commands* chapter of the *Command Line Interface Reference.*

Creating Directories

Use the **mkdir** command to create a new directory on the specific local device. This directory can then be incorporated as part of the path name for any file located in the local file system.

```
[local]host_name# mkdir { /flash | /usb1 | /hd-raid } /dir_name
```

Use the following command to create a directory named *configs*:

```
[local]host_name# mkdir /flash/configs
```

Renaming Files and Directories

Use the **rename** command to change the name of a file from its original name to a different name. Remember to use the same file extension, if applicable, to ensure that the file type remains unchanged.

Use the following command to rename a file named *iot_test.cfg* to *iot_accept.cfg* on the */flash* local device.

```
[local]host_name# rename /flash/iot_test.cfg /flash/iot_accept.cfg -noconfirm
```

Important Use the **rename** command only within the same local device. You cannot rename a file and place it onto another local device at the same time. To move a renamed file, you must use the **copy** command.

Copying Files

These instructions assume that you are at the root prompt for the Exec mode. To save your current configuration, enter the following command:

```
[local]host_name# copy from_url to_url [-noconfirm]
```

To copy a configuration file called *system.cfg* from a directory that was called *cfgfiles* to a directory named *configs_old*, enter the following command:

```
[local]host_name# copy /flash/cfgfiles/system.cfg
/flash/configs_old/system_2011.cfg
```

To copy a configuration file called *init_config.cfg* to the root directory of a TFTP server with a hostname of *config_server*, enter the following command:

```
[local]host_name# copy /flash/cfgfiles/init_confg.cfg
tftp://config_server/init_config.cfg
```

Deleting Files

The **delete** command removes a designated file from its specified location on the local file system.

Important This command does not support wildcard entries; each filename must be specified in its entirety.

Caution Do not delete the boot.sys file. If deleted, the system will not reboot on command and will be rendered inoperable.

```
[local]host_name# delete { /flash | /usb1 | /hd-raid }/filename [ -noconfirm
]
```

The following command deletes a file named *test.cfg* from the */flash* directory.

```
[local]host_name# delete /flash/test.cfg
```

Removing Directories

The **rmdir** command deletes a current directory on the specific local device. This directory can then be incorporated as part of the path name for any file located in the local file system.

Important The directory you want to remove (delete) must be empty before executing the **rmdir** command. If the directory is not empty, the CLI displays a "Directory not empty" message and will not execute.

```
[local]host_name# rmdir url /dir_name
```

The following command deletes an empty directory named *configs* in the */flash* directory.

```
[local]host_name# rmdir /flash/configs
```

Formatting Local Devices

The **format** command performs a low-level format of a local device. This operation formats the device to use the FAT16 formatting method, which is required for proper read/write functionality with the operating system.

Important Local devices that have been formatted using other methods such as NTFS or FAT32 may be used to store various operating system, CLI configuration, and crash log files. However, when placing a new local device into the MIO/UMIO/MIO2 for regular use, you should format the device via the system prior to use. This ensures that the proper file allocation table format is used, preventing any possible discrepancies between other formats used with other operating systems.

Caution The **filesystem format** command removes all files and information stored on the device.

To format a local device for use by the local file system, enter the following command:

```
[local]host_name# filesystem format { /flash | /usb1 | /hd-raid }
```

Applying Pre-existing CLI Configuration Files

A pre-existing CLI configuration file is any .cfg file created to provide utility functions (such as clearing all statistics during testing) or created off-line using a text editor. There may be pre-existing configuration files stored on the local file system that can be applied to a running system at any time.

Caution If a configuration file is applied to a system currently running another CLI configuration, any like contexts, services, logical interfaces, physical ports, IP address pools, or other configured items will be overwritten if the same command exists in the configuration file being applied. Take caution to ensure that you are knowledgeable of the contents of the file being applied and understand what the service ramifications are if a currently running command is overwritten. Also note that changes will not be saved automatically.

A CLI configuration file, or script containing CLI commands, can be applied to a running system by entering the following command at the Exec mode prompt:

```
[local]host_name# configure url [ verbose ]
```

url specifies the location of the CLI configuration file to be applied. It may refer to a local or a remote file.

The following command applies a pre-existing CLI configuration file named *clearcmds.cfg* in the */flash* directory.

```
[local]host_name# configure /flash/clearcmds.cfg
```

Viewing Files on the Local File System

This section describes how to view a variety of files.

Viewing the Contents of a Local Device

The contents, usage information, and file system directory structure of any local device can be viewed by entering the following command at the Exec mode prompt:

```
directory { /flash | /usb1 | /hd-raid }
```

Viewing CLI Configuration and boot.sys Files

The contents of CLI configuration and boot.sys files, contained on the local file system, can be viewed off-line (without loading them into the OS) by entering the following command at the Exec mode prompt:

```
[local]host_name# show file url { /flash | /usb1 | /hd-raid } filename
```

Where: *url* is the path name for the location of the file and *filename* is the name of the file, including any extension.

Important Operator and inspector-level users can execute the **show file** command but cannot execute the **directory** command.

Validating an Operating System File

The operating system software image file, identified by its .bin extension, is a non-readable, non-editable file that executes on the system, creating its runtime operating system (OS).

It is important to verify a new operating system image file before attempting to load it. To accomplish this, a proprietary checksum algorithm is used to create checksum values for each portion of the application stored within the .bin file during program compilation.

This information can be used to validate the actual file against the checksum values stored within the file during its compilation. If any portion of the image file has become corrupted (for example, the file was truncated or was transferred using ASCII mode instead of binary mode), then this information is reported and the file is deemed unusable.

To validate an operating system software image file, enter the following command at the Exec mode prompt:

```
[local]host_name# show version { /flash | /usb1 | /hd-raid } /[directory]/filename
 [all]
```

The output of this command displays the following information:

- Version number

- Description
- Date
- Boot Image
- Size
- Flags
- Platform

If an invalid file is found, the system displays a failure message similar to these:

```
Failure: Image /flash/image_version.bin CRC check failed!
Failure: /flash/image_version.bin, has a bad magic number
```

Configuring the Boot Stack

The boot stack consists of a prioritized listing of operating system software image-to-CLI configuration file associations. These associations determine the software image and configuration file that gets loaded during system startup or upon a reload/reboot. Though multiple associations can be configured, the system uses the association with the highest priority. If there is an error processing this association (for example, unable to locate one of the files), the system attempts to use the association with the next highest priority.

For VPC-SI and VPC-DI platforms, when the configuration file in the highest configured boot priority is not available (but the image file is), the system boots up with the configuration setup wizard after reloading instead of using the next available boot system priority. Priorities range from 1 to 100, with 1 being the highest priority. The maximum number of boot stack entries that may be configured in the boot.sys file is 10.

Boot stack information is contained in the boot.sys file, described in Understanding the boot.sys File, on page 162. In addition to boot stack entries, the boot.sys file contains any configuration commands required to define the system boot method as explained in the section that follows.

System Boot Methods

The local-boot method uses software image and configuration files stored locally on the system. On system startup or reboot, the system looks on one of its local devices or **/hd-raid** for the specific software image and accompanying configuration text file. When using the local-booting method, you only need to configure boot stack parameters.

The system can also be configured to obtain its software image from a specific external network server while it is paired with a configuration text file that resides on the system. When using network booting, you need to configure the following:

- Boot stack parameters, which define the files to use and in what priority to use them
- Boot interface and network parameters defining the remote management LAN interface and the methods to use to reach the external network server
- Network booting delay time and optional name server parameters defining the delay period (in seconds) to allow for network communications to be established, and the IP address of any Domain Name Service (DNS) name server that may be used

Viewing the Current Boot Stack

To view the boot stack entries contained in the boot.sys file run the Exec mode **show boot** command.

Important Operator and inspector-level users can execute the **show boot** command.

The examples below shows the command output for a local booting configuration. Notice that in these examples both the image file (operating system software) and configuration file (CLI commands) are located on the **/flash** device.

Important The StarOS image filename scheme changed with release 16.1. Pre-16.1, format = "production.*image*.bin". For 16.1 onwards, format = "asr5500-*image_number*.bin". This change is reflected in the examples provided below.

Example 1 – *StarOS releases prior to 16.1*:

```
boot system priority 18 \
    image /flash/15-0-builds/production.45666.bin \
    config /flash/general_config.cfg

boot system priority 19 \
    image /flash/15-0-builds/production.45717.bin \
    config /flash/general_config_3819.cfg

boot system priority 20 \
    image /flash/15-0-builds/production.45069.bin \
    config /flash/general_config_3665.cfg
```

Example 2 – *StarOS release 16.1 onwards*:

```
boot system priority 18 \
    image /flash/16-1-builds/asr5500-16.1.3.bin \
    config /flash/general_config.cfg

boot system priority 19 \
    image /flash/16-1-builds/asr5500-16.1.1.bin \
    config /flash/general_config_3819.cfg

boot system priority 20 \
    image /flash/16-1-builds/asr5500-16.1.0.bin \
    config /flash/general_config_3665.cfg
```

The example below shows the output for a combination network booting and local booting configuration. Notice in this example that the first two boot stack entries (Priorities 18 and 19) load the image file (operating system software) from an external network server using the Trivial File Transfer Protocol (TFTP), while all configuration files are located on the **/flash** device.

Also notice the boot network interface and boot network configuration commands located at the top of the boot stack. These commands define what remote management LAN interface(s) to use and information about communicating with the external network server that hosts the operating system software image file.

```
boot networkconfig static ip address mio1 192.168.1.150 netmask 255.255.255.0
boot delay 15
boot system priority 18 image tftp://192.168.1.161/tftpboot/image_version.bin \config
/flash/general_config.cfg
boot system priority 19 image tftp://192.168.1.161/tftpboot/image_version.bin \config
```

```
/flash/general_config.cfg
boot system priority 20 image /flash/image_version.bin \config /flash/general_config.cfg
```

To identify the boot image priority that was loaded at the initial boot time enter:

show boot initial-config

The example below displays the output:

```
[local]host_name# show boot initial-config
Initial (boot time) configuration:
      image tftp://192.168.1.161/tftpboot/image_version.bin \
       config /flash/config_name.cfg
      priority 1
```

Adding a New Boot Stack Entry

Important Before performing this procedure, verify that there are less than 10 entries in the boot.sys file and that a higher priority entry is available (i.e. that minimally there is no priority 1 entry in the boot stack). Refer to *Viewing the Current Boot Stack* for more information.

If priority 1 is in use, then you must renumber the existing entry(ies) to ensure that at least that priority is available. The maximum number of boot stack entries that can be contained in the *boot.sys* file is 10. If there are already 10 entries in the boot stack, you must delete at least one of these entries (typically, the lowest priority) and, if necessary, renumber some or all of the other entries before proceeding. Refer to Deleting a Boot Stack Entry, on page 168 for more information.

This procedure details how to add new boot stack entries to the *boot.sys* file. Make sure you are at the Exec mode prompt and enter the following commands:

```
configure
 boot system priority number image image_url config cfg_url
```

The following command creates a new boot stack entry, using a boot priority of *3*.

```
boot system priority 3 image /flash/image_filename.bin config
/flash/config_name.cfg
```

Important Boot stack changes saved to the boot.sys file are not executed until the system is rebooted.

Synchronize the local file systems on the CF VMs with the following command:

filesystem synchronize all

Deleting a Boot Stack Entry

This procedure details how to remove an individual boot stack entry from the boot.sys file. Make sure you are at the Exec mode prompt and enter the following commands:

```
configure
   no boot system priority number
```

Where *number* specifies the boot priority used for the boot stack entry. This command removes that specific entry from the boot stack, causing the boot.sys file to be overwritten.

Upgrading the Operating System Software

This section describes how to manually perform the StarOS binary image upgrade procedure.

Prior to initiating the StarOS software upgrade procedure, make sure the conditions described in the *Prerequisites* section are met.

Caution Undeploying/redeploying VPC is not supported after the bin upgrade. Deactivating VPC removes the upgraded StarOS bin image.

To upgrade the StarOS software manually:

1. *Obtain VIP Addresses for AutoVNF, CF, ESC and UEM*
2. Identify OS Release Version and Build Number, on page 171
3. Download the Software Image from the Support Site, on page 172
4. *Verify Zookeeper Database*
5. *Verify ESC Database*
6. Verify Free Space on the /flash Device, on page 174
7. Transfer StarOS Image to /flash, on page 175
8. Save the Running Configuration, on page 179
9. Synchronize File Systems, on page 187
10. Reboot the System, on page 181

Prerequisites

Prior to performing an upgrade of StarOS software containing CF and SF VNFCs, check if the following prerequisites are met:

- You'll need the login credentials and IP address of AutoDeploy, AutoVNF, ESC, UEM, and CF VMs. You should have administrative rights to the OpenStack setup.
- Verify the OpenStack status. The Ansible output should all pass.

```
cd /home/stack/
source stackrc
cd /home/stack/ansible/
ansible-playbook -i inventory openstack_verify.yml
```

- Check if the health of AutoVNF/ESC/EM/VNF VM is normal through the UltraM health logs on AutoIT. If any of the VM(s) are not normal, then take necessary actions to rcover the health of the corresponding VM(s).

- You should have the new StarOS binary image file (for manual upgrade).
- Ensure that there are no pending transactions between ESC, UEM and CF.
- Be sure to take a backup of the original StarOS bin file.

Obtain VIP Addresses for AutoVNF, CF, ESC and UEM

This section provides instructions that are applicable only to the upgrade of CF and SF VNFCs.

To collect the VIP addresses for AutoVNF, CF, ESC and UEM VMs:

1. Log on to the AutoDeploy VM as the default user, *ubuntu*.

   ssh ubuntu@*<ad_vm_address>*

2. Switch to the *root* user.

   sudo -i

3. Enter the ConfD CLI.

   confd_cli -u admin -C

4. Enter the *admin* user credentials when prompted.
5. Collect the VIP address of AutoVNF, ESC, UEM and CF VMs.

   show vnfr

   Example output:

```
vnfr autoit-f-autovnf
 vnfd     f-autovnf
 vnf-type usp-uas
 state    deployed
 external-connection-point avf
  virtual-link-ref    management
  ip-address          192.168.100.26
  floating-ip-address 10.225.202.94
vnfr sj-autovnf-esc
 vnfd     esc
 vnf-type esc
 state    deployed
 external-connection-point esc
  virtual-link-ref management
  ip-address       192.168.100.22
```

```
vnfr sj-autovnf-vpc
 vnfd     vpc
 vnf-type ugp
 state    alive
 external-connection-point cf
  virtual-link-ref management
  ip-address       192.168.100.38
 external-connection-point em
  virtual-link-ref management
  ip-address       192.168.100.21
```

Identify OS Release Version and Build Number

The operating system can be configured to provide services and perform pre-defined functions through commands issued from the CLI.

The operating system software is delivered as a single binary file (**.bin** file extension) and is loaded as a single instance for the entire system.

- *For StarOS releases prior to 16.1*, the image filename is identified by its release type, build number and platform type. For example: **production.***build_number***.asr5500.bin**. For example, **production.54029.asr5500.bin**.
- *For StarOS release 16.1 onwards*, the image filename is identified by a suffix specifying its platform type and release number. For example, **asr5500-***release_number***. bin**. For example, **asr5500-16.1.0.bin**.

For StarOS releases 20.0 and higher, a starfile image must be signed with an REL key before being released. A deployable image will be signed with an REL key having a ".bin.SPA" extension, where "A" identifies the revision level of the signing key. For example, **asr5500-20.0.0.bin.SPA**. If a signing key becomes compromised, a new key is created and the revision level increments to "B".

For StarOS releases 20.0 and higher Trusted images have been introduced. The difference between a Trusted build and a Normal build is the absence of unsecure programs ftpd, telnet and tcpdump, as well as the addition of a staros.conf file for security options. Trusted images are identifiable by the presence of "_T" in the platform name. For example, **asr5500_T-20.0.0.bin.SPA**.

To identify the StarOS software version and build information:

1. Log on to the VNF to be upgraded.
2. Enter the following Exec mode command in the StarOS command line interface:

   show version

   Example output:

   ```
   Active Software:
     Image Version:            21.9.0.69918
   ```

```
Image Build Number:        69918

Image Description:         Deployment_Build

Image Date:                Sun Jul 22 12:08:55 EDT 2018

Boot Image:                /flash/staros.bin

Source Commit ID:          94797337b6c1691541ea0dd86f2f29b0f2c3630c
```

3. Execute the following Exec mode command to display additional information about the StarOS build release.

   show build

Download the Software Image from the Support Site

This section provides instructions that are applicable only to the upgrade of CF and SF VNFCs.

Access to the Cisco support site and download facility is username and password controlled. You must have an active customer account to access the site and download the StarOS image.

Download the software image to a network location or physical device (USB stick) from which it can be uploaded to the */flash* device. Contact your Cisco representative or Cisco TAC for additional information.

For UGP-based VNF, perform the following steps to download the new bin file to AutoVNF or OSPD VM.

1. Log on to the AutoVNF of the corresponding VNF.

   ssh ubuntu@*<ad_vm_address>*

   Command example:

   ssh ubuntu@10.225.202.94

2. Create a directory to download the new StarOS qvpc-di binary file to AutoVNF/OSPD.

   cd /home/ubuntu/

   mkdir StarOSBinUpgrade

3. Download the new StarOS qvpc-di binary file from the Cisco support site and copy the file to the *StarOSBinUpgrade* directory.

   cd StarOSBinUpgrade

   Then, use the following command to verify if the directory contains the new bin file.

   ls -lrt /home/ubuntu/StarOSBinUpgrade

   Example output:

```
total 172560

-r--r--r-- 1 ubuntu ubuntu 176698880 Jul 24 23:29 qvpc-di-21.9.0.69932.bin
```

Verify Zookeeper Database

This section provides instructions that are applicable only to the upgrade of CF and SF VNFCs.

To verify the zookeeper database:

1. Log on to the AutoVNF using the floating IP.

   ssh ubuntu@<*ad_vm_address*>

   Command example:

   ssh ubuntu@10.225.202.94

2. Log on to the UEM VM using the VIP address fetched in the Obtain VIP Addresses for AutoVNF, CF, ESC and UEM, on page 170.

   ssh ubuntu@<*vip-addr*>

   Command example:

   ssh ubuntu@192.168.100.21

3. Become the *root* user.

   sudo -i

4. Collect the UEM orchestration IP address for Zookeeper database connection.

   #ifconfig

   ```
   eth0      Link encap:Ethernet  HWaddr fa:16:3e:71:1d:08
                inet addr:192.168.200.12  Bcast:192.168.200.255  Mask:255.255.255.0
   ```

5. Navigate to the */opt/cisco/usp/packages/zookeeper/*<*current*>*/bin* directory.

6. Execute the following script from the command line to access the UEM Zookeeper database.

   zkCli.sh -server *ip_addr:port_num*

   For example:

   zkCli.sh -server 192.168.200.12:2181

7. Check the zookeeper database and ensure that there are no pending requests between UEM and CF VMs.

   ls /request

   Example output:

   ```
   []
   <Ctrl+D to exit Zookeeper shell>
   ```

Verify ESC Database

This section provides instructions that are applicable only to the upgrade of CF and SF VNFCs.

To verify the ESC database:

1. Log on to the AutoVNF using the floating IP.

   ssh ubuntu@<*ad_vm_address*>

   Command example:

```
ssh ubuntu@10.225.202.94
```

2. Log on to the ESC VM using the VIP address fetched in the Obtain VIP Addresses for AutoVNF, CF, ESC and UEM, on page 170.

   ssh admin@<*vip-addr*>

   Command example:

   ```
   ssh admin@192.168.100.22
   ```

3. Check the ESC database to ensure there are no pending transactions.

   ```
   sudo /opt/cisco/esc/pgsql/bin/psql -U esc -p 7878 -h localhost -c
   'select * from esc_schema.workitem';
   ```

   ```
   config_id | request_id | mo_type | config_action | config_state
   -----------+------------+---------+---------------+--------------
   (0 rows)
   ```

4. Execute the following command to check the transaction details.

   ```
   escadm ip_trans
   ```

   Example output:

   ```
   Number of in-progress transaction events = 0
   ```

Verify Free Space on the /flash Device

Verify that there is enough free space on the **/flash** device to accommodate the new StarOS image file.

To verify the available space on *flash* directory:

1. Log on to the CF VM using the previously fetched VIP address in the Obtain VIP Addresses for AutoVNF, CF, ESC and UEM, on page 170.

   ssh ubuntu@<*vip-addr*>

   Command example:

   ```
   ssh ubuntu@192.168.100.38
   ```

2. Enter the following Exec mode command:

   ```
   [local]host_name# directory /flash
   ```

   The following is an example of the type of directory information displayed:

   ```
   -rwxrwxr-x    1  root     root         7334 May    5 17:29 asr-config.cfg
   -rwxrwxr-x    1  root     root         399 Jun    7 18:32 system.cfg
   -rwxrwxr-x    1  root     root         10667 May 14 16:24 testconfig.cfg
   -rwxrwxr-x    1  root     root         10667 Jun    1 11:21 testconfig_4.cfg
   -rwxrwxr-x    1  root     root         5926 Apr    7 16:27  tworpcontext.cfg
   -rwxrwxr-x    1  root     root         15534 Aug    4 13:31 test_vlan.cfg
   -rwxrwxr-x    1  root     root         2482 Nov 18 11:09 gateway2.cfg
   -rwxrwxr-x    1  root     root         159106048 Dec 31  2011 image_filename
   1136352  /flash
   Filesystem           1k-blocks      Used Available Use% Mounted on
   /var/run/storage/flash/part1 3115468   1136352  30018336   4%
   /mnt/user/.auto/onboard/flash
   ```

Note the "Available" blocks in the last line of the display. After displaying the directory information, the CLI returns to root and the following prompt appears:

```
[local]host_name#
```

Transfer StarOS Image to /flash

For StarOS-based VNF, transfer the new operating system image file to the */flash* directory on the MIO/UMIO/MIO2 VPC-DI active CF or VPC-SI using one of the following methods:

- Transfer the file to the */flash* device using an FTP client with access to the system.

Important Whenever transferring a operating system software image file using the file transfer protocol (FTP), the FTP client must be configured to transfer the file using binary mode. Failure to use binary transfer mode will make the transferred operating system image file unusable. In release 20.0 and higher Trusted StarOS builds, FTP is not supported.

- Transfer the file to the */flash* device using an SFTP client with access to the system.

For UGP-based VNF, copy the new StarOS bin to the active CF by following these steps.

1. Log on to the AutoVNF or OSPD VM where the new bin file is downloaded.

   ssh ubuntu@*<ad_vm_address>*

   Command example:

   ssh ubuntu@10.225.202.94

2. Navigate to the directory where the new bin file is downloaded from the Cisco support site.

   cd /home/ubuntu/StarOSBinUpgrade/ && ls -lrt

   Example output:

   ```
   total 172560
   -r--r--r-- 1 ubuntu ubuntu 176698880 Jul 24 23:29 qvpc-di-21.9.0.69932.bin
   ```

3. SFTP to the CF VM.

   For example:

   sftp ubuntu@192.168.100.38

4. Navigate to the *sftp* directory.

   *#sftp>***pwd**

   ```
   Remote working directory: /
   ```

   *#sftp>***ls**

   ```
   hd-raid sftp
   ```

   *#sftp>***cd sftp**

5. Upload the new binary file to the *sftp* directory.

```
#sftp>put image_filename.bin
```

Example output:

```
#sftp>put qvpc-di-21.9.0.69932.bin
Uploading qvpc-di-21.9.0.69932.bin to /.auto/onboard/flash/sftp/qvpc-di-21.9.0.69932.bin
qvpc-di-21.9.0.69932.bin                          100%  169MB 168.5MB/s   00:01
```

6. Log on to the CF VM using the VIP address fetched in the Obtain VIP Addresses for AutoVNF, CF, ESC and UEM, on page 170.

   ssh ubuntu@<*vip-addr*>

   Command example:

   ```
   ssh ubuntu@192.168.100.38
   ```

7. Copy the new bin from *sftp* to *flash* directory.

   ```
   copy /flash/sftp/image_filename.bin /flash/updated.bin
   ```

   Example output:

   ```
   #copy /flash/sftp/qvpc-di-21.9.0.69932.bin /flash/updated.bin

   ************************************************************

   Transferred 176698880 bytes in 2.718 seconds (63486.9 KB/sec)
   ```

8. Delete the new bin from *sftp* directory.

   ```
   delete /flash/sftp/image_filename.bin
   ```

   Example output:

   ```
   delete /flash/sftp/qvpc-di-21.9.0.69932.bin
   Are you sure? [Yes|No]: yes
   File /flash/sftp/qvpc-di-21.9.0.69932.bin removed
   ```

9. Verify that the image file was successfully transferred to the */flash* device by running the following Exec mode command:

   ```
   [local]host_name# directory /flash
   ```

   The image filename should appear in the displayed output.

10. Execute the following command to verify the build information.

    ```
    show version /flash/image_filename.bin
    ```

Saving a Copy of the Current Configuration File

Prior to upgrading to a new software release, you should copy and rename the current configuration file to the **/flash** device and to an off-chassis location (external memory device or network URL). This renamed copy assures that you will have a fallback, loadable configuration file should a problem be encountered during the upgrade.

Downgrading from Release 15.0 to 14.0

Release 14 and Release 15 chassis IDs use different encryption formats. Release 15 will recognize a Release 14 chassis ID and consider it as valid. Upgrading from 14.x to 15.0 will not require changing the chassis ID or configuration file.

However, if the chassis key is reset in Release 15 through the setup wizard or **chassis-key** CLI command, a new chassis ID will be generated in Release 15 format (44 instead of 16 characters). Release 14 builds will not recognize the 44-character chassis ID. If the chassis is subsequently downgraded to Release 14, a new 16-character chassis ID will be generated. To accommodate the old key format, you must save the configuration file in pre-v12.2 format before the downgrade. If you attempt to load a v15 configuration file on the downgraded chassis, StarOS will not be able to decrypt the password/secrets stored in the configuration file.

Downgrading from Release 20.0

Prior to release 20.0, local-user passwords were hashed with the MD5 message digest-algorithm and saved in the database. In release 20. 0, PBKDF2 (Password Based Key Derivation Function - Version 2) is now used to derive a key of given length, based on entered data, salt and number of iterations. Local-user account passwords are hashed using the PBKDF2 method with a randomly generated salt coupled with a large number of iterations to make password storage more secure.

Since hash functions are one-way, it is not possible to convert PBKDF2 hashed passwords to the MD5 format. The local-user database must be downgraded prior to reverting to StarOS releases prior to 20.0.

To downgrade the local-user database to use the MD5 hash algorithm, a Security Administrator must run the Exec mode **downgrade local-user database** command. StarOS prompts for confirmation and requests the Security Administrator to reenter a password. The entered password re-authenticates the user prior to executing the downgrade command. After verification, the password is hashed using the appropriate old/weak encryption algorithm and saved in the database to allow earlier versions of StarOS to authenticate the Security Administrator.

The downgrade process does not convert PBKDF2 hashed passwords to MD5 format. The downgrade process re-reads the database (from the /flash directory), reconstructs the database in the older format, and writes it back to the disk. Since the PBKDF2 hashed passwords cannot be converted to the MD5 hash algorithm, and earlier StarOS releases cannot parse the PBKDF2 encryption algorithm, StarOS suspends all those users encrypted via the PBKDF2 algorithm. Users encrypted via the MD5 algorithm ("Weak Hash" flag) can continue to login with their credentials. After the system comes up with the earlier StarOS release, suspended users can be identified in the output of the **show local-user [verbose]**command.

To reactivate suspended users a Security Administrator can:

- Set temporary passwords for suspended users, using the Exec mode **password change local-user** *username* command.
- Reset the suspend flag for users, using the Configuration mode **no suspend local-user** *username* command.

Off-line Software Upgrade

An off-line software upgrade can be performed for any system, upgrading from any version of operating system software to any version, regardless of version number. This process is considered off-line because while many of the steps can be performed while the system is currently supporting sessions, the last step of this process requires a reboot to actually apply the software upgrade.

This procedure assumes that you have a CLI session established and are placing the new operating system image file onto the local file system. To begin, make sure you are at the Exec mode prompt:

```
[local]host_name#
```

To perform offline software upgrade:

1. Configure a Newcall Policy, on page 178
2. Configure a Message of the Day Banner, on page 179
3. Back up the Current CLI Configuration File , on page 179
4. Save the Running Configuration, on page 179
5. Create a New Boot Stack Entry, on page 181
6. Synchronize File Systems, on page 187
7. Reboot the System, on page 181

Configure a Newcall Policy

Configure a newcall policy from the Exec mode to meet your service requirements. When enabled the policy redirects or rejects new calls in anticipation of the system reload that completes the upgrade process. This reduces the amount of service disruption to subscribers caused by the system reload that completes the upgrade.

Important Newcall policies are created on a per-service basis. If you have multiple services running on the chassis, you can configure multiple newcall policies.

The syntax for newcall policies is described below:

```
[local]host_name# newcall policy { asngw-service | asnpc-service | sgsn-service
 } { all | name service_name } reject
[local]host_name# newcall policy { fa-service | lns-service | mipv6ha-service
 } { all | name service_name } reject
[local]host_name# newcall policy { ha-service | pdsn-service |
pdsnclosedrp-service  } { all | name service_name } { redirect target_ip_address
 [ weight  weight_num ] [ target_ipaddress2 [ weight weight_num ] ...
target_ip_address16 [ weight weight_num ] | reject }
[local]host_name# newcall policy ggsn-service { apn name apn_name | all | name
 service_name } reject
[local]host_name# newcall policy hnbgw-service { all | name service_name } reject
[local]host_name# newcall policy { pcc-af-service | pcc-policy-service } {
all | name service_name } reject
[local]host_name# newcall policy {pcc-af-service | pcc-policy-service } { all
 | name service_name } reject
[local]host_name# newcall policy mme-service { all | name service_name } reject
```

For complete information about the above commands, see the *Exec Mode Commands* chapter of the *Command Line Interface Reference*.

Configure a Message of the Day Banner

Optional: Configure a "Message of the Day" banner informing other management users that the system will be rebooted by entering the following command from the Global Configuration mode prompt.

```
[local]host_name(config)# banner motd "banner_text"
```

banner_text is the message that you would like to be displayed and can be up to 2048 alphanumeric characters. Note that *banner_text* must begin with and end in quotation marks (" "). For more information in entering CLI banner information, see the *CLI Reference*. The banner is displayed when an administrative user logs onto the CLI.

Back up the Current CLI Configuration File

Back up the current CLI configuration file by entering the following command:

```
[local]host_name# copy from_url to_url [ -noconfirm ]
```

This creates a mirror-image of the CLI configuration file linked to the operating system defined in the current boot stack entry.

The following command example creates a backup copy of a file called *general.cfg* located on the **/flash** device to a file called *general_3652.cfg*:

```
[local]host_name# copy /flash/general.cfg /flash/general_3652.cfg
```

Save the Running Configuration

Save the currently running, upgraded configuration prior to rebooting the chassis.

To save the boot configuration:

1. Log on to the VNF using the previously fetched VIP address in the Obtain VIP Addresses for AutoVNF, CF, ESC and UEM, on page 170 .

   ```
   ssh ubuntu@<vip-addr>
   ```

   Command example:

   ```
   ssh ubuntu@192.168.100.38
   ```

2. *Optional*. Execute the following command in the Exec mode.

   ```
   chassis key value 1234
   ```

   ```
   Save config before reload chassis, EVEN IF the same old key value is used.
   Old config scripts will become invalid after reload.
   ```

Important This step is optional, and needed only if the chassis key is not set.

3. Save the boot configuration in the flash directory.

   ```
   save configuration /flash/system.cfg
   ```

   ```
   Warning: About to overwrite boot configuration file
   Are you sure? [Yes|No]: yes
   ```

   This will update the boot configuration to use the new bin image.

   Use the following command to check the boot configuration.

 # show boot

```
Monday May 21 20:39:57 UTC 2018

boot system priority 8 \
    image /flash/sftp/production.YYYYY.qvpc-di.bin \
    config /flash/sftp/tb5_vnf1_dayN.cfg

boot system priority 9 \
    image /flash/staros.bin \
    config /flash/sftp/tb5_vnf1_dayN.cfg

boot system priority 10 \
    image /flash/staros.bin \
    config /flash/system.cfg
```

4. Enter the configuration mode to change the boot priority of new StarOS bin file.

   #config

   #boot system priority 1 image /flash/updated.bin config /flash/system.cfg

   #end

5. Verify the new boot priority.

   #show boot

```
boot system priority 1 \

    image /flash/updated.bin \

    config /flash/system.cfg

boot system priority 10 \

    image /flash/staros.bin \

    config /flash/system.cfg
```

6. Verify whether the flash directory contains the boot configuration and new bin.

   dir /flash

```
total 320376
-rw-rw-r-- 1 root root 134 May 3 10:11 boot.sys
-rw-rw-r-- 1 root root 3920672 May 11 19:49 crashlog2
drwxrwxr-x 2 root root 4096 May 11 19:49 crsh2
-rw-rw-r-- 1 root root 156 May 11 19:49 module.sys
drwxrwxr-x 3 root root 4096 May 11 19:49 patch
drwxrwxr-x 2 root root 4096 May 11 19:49 persistdump
-rw-rw-r-- 1 root root 79 May 11 19:49 restart_file_cntr.txt
drwxrwxr-x 3 root root 4096 May 11 20:07 sftp
-rw-rw-r-- 1 root root 160871936 May 3 10:11 staros.bin
-rw-rw-r-- 1 root root 5199 May 11 19:57 system.cfg
-rw-rw-r-- 1 root root 163227136 May 11 20:07 updated.bin
320476 /flash
Filesystem 1K-blocks Used Available Use% Mounted on
/var/run/storage/boot1/part2
4112620 320476 3792144 8% /mnt/user/.auto/onboard/flash
```

Create a New Boot Stack Entry

Create a new boot stack entry for the new file group, consisting of the new operating system image file and the currently used CLI configuration file by entering the following Global Configuration command:

```
[local]host_name(config)# boot system priority number image image_url /flash filename
 config cfg_url /flash/filename
```

Assign the next highest priority to this entry, by using the <N-1> method, wherein you assign a priority number that is one number less than your current highest priority.

Important Run the Exec mode **show boot** command to verify that there are less than 10 entries in the boot.sys file and that a higher priority entry is available (minimally there is no priority 1 entry in the boot stack).

If priority 1 is in use, you must renumber the existing entries to ensure that at least that priority is available.

The maximum number of boot stack entries that can be contained in the boot.sys file is 10. If there are already 10 entries in the boot stack, you must delete at least one of these entries (typically, the lowest priority) and, if necessary, renumber some or all of the other entries before proceeding. Use the no boot system priority command to delete a book stack entry.

```
[local]host_name# configure
[local]host_name(config)# no boot system priority number
```

To add new boot stack entries to the boot.sys file enter the following commands:

```
[local]host_name# configure
[local]host_name(config)# boot system priority number image image_url config cfg_url
```

For information on using the **boot system priority** command, refer to the Adding a New Boot Stack Entry, on page 168 .

Synchronize File Systems

Synchronize the local file systems on the management cards by entering the following command:

```
[local]host_name# filesystem synchronize all
```

Reboot the System

To reboot the system (VNF):

1. Log on to the VNF using the previously fetched VIP address in the Obtain VIP Addresses for AutoVNF, CF, ESC and UEM, on page 170 .

   ```
   ssh ubuntu@<vip-addr>
   ```

   Command example:

   ```
   ssh ubuntu@192.168.100.38
   ```

2. Enter the following Exec mode command:

   ```
   [local]host_name# reload [-noconfirm]
   ```

   As the system reboots, it loads the new operating system software image and its corresponding CLI configuration file using the new boot stack entry configured earlier.

3. *Optional for PDSN:* If you are using the IP Pool Sharing Protocol during your upgrade, refer to *Configuring IPSP Before the Software Upgrade* in the *PDSN Administration Guide*.

4. After the reload is complete, log on to the VNF and make sure it is loaded with the intended StarOS version and all the cards have booted up and are in active or stand-by state as expected.

   show version

   Example output:

```
Active Software:
  Image Version:                  21.9.0.69977
  Image Build Number:             69977
  Image Description:              Build
  Image Date:                     Mon Jul 30 06:48:34 EDT 2018
  Boot Image:                     /flash/updated.bin
  Source Commit ID:               abde005a31c93734c89444b8aec2b6bb2d2e794d
```

   show card table

   Example output:

```
Slot         Card Type                                Oper State     SPOF  Attach
-----------  ---------------------------------------  -------------  ----  ------
 1: CFC      Control Function Virtual Card            Active         No
 2: CFC      Control Function Virtual Card            Standby        -
 3: FC       4-Port Service Function Virtual Card     Standby        -
 4: FC       4-Port Service Function Virtual Card     Standby        -
 5: FC       4-Port Service Function Virtual Card     Standby        -
 6: FC       4-Port Service Function Virtual Card     Standby        -
 7: FC       4-Port Service Function Virtual Card     Standby        -
 8: FC       4-Port Service Function Virtual Card     Standby        -
 9: FC       4-Port Service Function Virtual Card     Standby        -
10: FC       4-Port Service Function Virtual Card     Standby        -
```

5. Run the following Exec mode command to display additional information about the running StarOS build release.

   show build

6. *Optional*. Verify the operational state of CF and SF VNFCs.

Note This step is relevant only for the upgrade of CF and SF VNFCs.

   a. Repeat the steps in *Verify Zookeeper Database* and *Verify ESC Database* sections.

   b. Log on to the UEM using either the floating IP or from the AutoVNF using the UEM VIP.

      ssh ubuntu@*<vip-addr>*

      Command example:

      ssh ubuntu@192.168.100.21

   c. Become the *root* user.

      sudo -i

   d. Collect the UEM orchestration IP address for Zookeeper database connection.

#ifconfig

```
eth0      Link encap:Ethernet  HWaddr fa:16:3e:71:1d:08

          inet addr:192.168.200.12  Bcast:192.168.200.255  Mask:255.255.255.0
```

e. Navigate to the */opt/cisco/usp/packages/zookeeper/<current>/bin* directory.

f. Run Zookeeper tool to access the UEM Zookeeper database.

zkCli.sh -server *<vip-addr>:port_num*

Command example:

zkCli.sh -server 192.168.200.12:2181

Make sure there are no outstanding requests between UEM and CF.

g. Verify the "state": "alive" for each of the CFs and SFs using the following commands:

get /oper/vnfs/*vnf_name***/vdus/***vdu_name***/cf1**

get /oper/vnfs/*vnf_name***/vdus/***vdu_name***/cf2**

get /oper/vnfs/*vnf_name***/vdus/***vdu_name***/sf1**

get /oper/vnfs/*vnf_name***/vdus/***vdu_name***/sf2**

Command examples:

get /oper/vnfs/tb1-autovnf1_vpc-vpc-core/vdus/vdu-cf1/cf1

get /oper/vnfs/tb1-autovnf1_vpc-vpc-core/vdus/vdu-cf1/cf2

get /oper/vnfs/tb1-autovnf1_vpc-vpc-core/vdus/vdu-sf1/sf1

get /oper/vnfs/tb1-autovnf1_vpc-vpc-core/vdus/vdu-sf1/sf2

h. Look for the state Alive in the console output.

```
zk: localhost:2181(CONNECTED) 2] get
/oper/vdus/control-function/BOOT_generic_di-chassis_CF1_1
{"id":"BOOT_generic_di-chassis_CF1_1","state":"alive","vnfcId":"cf-vnfc-di-chassis","uuid":"c4",
"host":"tb5-ultram-osd-compute-2.localdomain","vimId":"523b921c-7266-4fd5-90bb-5157cffc6951",
"cpts":[{"cpid":"di_intf1","state":"alive","subnet":"6102e9b5-8555-41f5-8cdc-0b47d30a6f7a",
"netmask":"255.255.255.0","dhcp":true,"vl":"vl-vnf1-DI-INTERNAL1-CAT","vnfc":"cf-vnfc-di-chassis",
"port_id":"19539aea-edbf-4acf-a57c-af5627d859ea","ip_address":"192.168.10.3",
"mac_address":"fa:16:3e:19:80:ed","network":"0d72f553-5a9c-4904-b3ea-83371a806e23"},
{"cpid":"di_intf2","state":"alive","nicid":1,"subnet":"30002d02-761d-4ccb-8a9e-d6188cdf54a3",
"netmask":"255.255.255.0","dhcp":true,"vl":"vl-vnf1-DI-INTERNAL2-CAT","vnfc":"cf-vnfc-di-chassis",
"port_id":"ff1da1e1-ecf3-477d-98b7-398c3c77fc8d","ip_address":"192.168.11.13",
"mac_address":"fa:16:3e:89:88:23","network":"9f109c0a-b1e7-4d90-a746-5de4ab8ef536"},
{"cpid":"orch","state":"alive","nicid":2,"subnet":"729e9dd2-3c75-43eb-988a-769016f2f44c",
"netmask":"255.255.255.0","dhcp":true,"vl":"vl-vnf1-UAS-ORCH-CAT","vnfc":"cf-vnfc-di-chassis",
"port_id":"81370948-f686-4812-820c-20ec5d3d3cdd","ip_address":"172.168.11.17","mac_address":"fa:16:3e:1d:0b:56",
"network":"9a286170-e393-4ba5-abce-147a45fb337a"},{"cpid":"mgmt","state":"alive","nicid":3,
"subnet":"9778a11b-1714-4e84-bbc2-86c84b111e8e","netmask":"255.255.255.0","dhcp":true,"vl":"vl-vnf1-UAS-MGMT-CAT",
"vnfc":"cf-vnfc-di-chassis","port_id":"6130cbb4-3dd8-4822-af90-50dac98f2f0d",
"ip_address":"172.168.10.17","mac_address":"fa:16:3e:42:92:47","network":"e278b524-e9a9-48c1-a45b-956a8c3ea583"}],
"monitor":true,"vduId":"control-function"}
cZxid = 0x100000051
ctime = Fri May 18 19:04:40 UTC 2018
mZxid = 0x10000024a
mtime = Mon May 21 17:48:19 UTC 2018
pZxid = 0x100000051
```

```
cversion = 0
dataVersion = 12
aclVersion = 0
ephemeralOwner = 0x0
dataLength = 1625
numChildren = 0
```

Note You can use use **CTRL+D** to exit the zookeeper CLI.

i. From the UEM VM as a root user, log on to the *ncs_cli* and check for devices live status.

~$ sudo -i

ncs_cli -C -u admin

 # show devices device *device_name* **live-status**

Verify that the command output reflects the correct 'state' and 'card-state' of each card.

Example output:

 # show devices device tb1-autovnf1_vpc-vpc-core-cf-nc live-status

```
<snip>

VNFC  CURRENT VNFC  VDU   CARD  CARD  NUMBER CPU   DISK  START UPTIME NOVA  ID    DATE  FROM  TO
REF   STATE   INSTANCE REF  TYPE  SLOT  OF     UTILIZATION SPACE TIME       LAUNCH       AND   STATE STATE
            ID               ID    CORES                                   CMD        TIME
-------------------------------------------------------------------------------------------------------
cf1   -     cf1   cf    [illegible] 1     -     -     -     -     -     -

cf2   -     cf2   cf    [illegible] 2     -     -     -     -     -     -

sf1   -     sf1   sf    [illegible] 3     -     -     -     -     -     -

sf2   -     sf2   sf    [illegible] 4     -     -     -     -     -     -


live-status vnfd sj-autovnf-vpc-abc

 version            6.0

 vnfm vim-tenant-name abc

 vnfm tenant-name abc

 vnfm ipaddr     192.168.100.22

 vnfm port       830

 vnfm username   ubuntu

 vnfm password   "$4$+HLzhFFzHq66nqtTsc00CfiODYHqlUSVmkn1tRe1f84byNakWEa9sJ8sY/
cwfFME3aG0UaBC\nvvNNAMkuXQI9Ksfu5IiQQ9ViWbbHwl6IEFQ="

 virtual-link vl-di-internal1

  auto-vnf-connection-ref di-internal1

 virtual-link vl-management
```

```
 auto-vnf-connection-ref management

virtual-link vl-orchestration

 auto-vnf-connection-ref orchestration

virtual-link vl-abc-vpc-svc

 auto-vnf-connection-ref sj-autovnf-abc-vpc-svc

vdu cf

 ssh-keygen         false

 vm-image           076c887a-a12c-4a0b-b4d6-b2d213f64b9e

 lifecycle-event-initialization staros_config.txt

  source-url http://192.168.100.9:5000/config/sj-autovnf-vpc-abc/cf/staros_config.txt


 lifecycle-event-initialization staros_param.cfg

  source-url http://192.168.100.9:5000/config/sj-autovnf-vpc-abc/cf/staros_param.cfg


 ned cisco-staros-nc

  user        "$4$+HLzsE1kLJOeuoFWyOsmBWY2LHjOi2WtJdKy/OIux7YHhsNY/
O8hnA9/WwWuFD5trHrW3ZHs\nLo4TfiAKqYwxdNKqFYyoTxH2hrLJV5DgwmE="

  password
"$4$+HLzsXtCHJ2vsYZD5s0RGtBRY/dHDU1mgHJX7wCt3o1DMtQZqpBLDcNSJumC7n5rnkVxwI1s\
ncJYeCOFLrqpLHXm3xtXyMdtT7WVzvRMtdao="

  netconf

  port-number 830

 card-type          control-function

 usp-auto-vnf-id    sj-autovnf-vpc-abc-cf

 vnfc cf-vnfc-ugp

<snip>
```

Save the Running Configuration

Save the currently running, upgraded configuration prior to rebooting the chassis.

To save the boot configuration:

1. Log on to the VNF using the previously fetched VIP address in the Obtain VIP Addresses for AutoVNF, CF, ESC and UEM, on page 170 .

   ssh ubuntu@*<vip-addr>*

   Command example:

   ssh ubuntu@192.168.100.38

2. *Optional*. Execute the following command in the Exec mode.

   chassis key value 1234

   ```
   Save config before reload chassis, EVEN IF the same old key value is used.
   Old config scripts will become invalid after reload.
   ```

Important This step is optional, and needed only if the chassis key is not set.

3. Save the boot configuration in the flash directory.

   save configuration /flash/system.cfg

   ```
   Warning: About to overwrite boot configuration file
   Are you sure? [Yes|No]: yes
   ```

   This will update the boot configuration to use the new bin image.

   Use the following command to check the boot configuration.

   # show boot

   ```
   Monday May 21 20:39:57 UTC 2018

   boot system priority 8 \
       image /flash/sftp/production.YYYYY.qvpc-di.bin \
       config /flash/sftp/tb5_vnf1_dayN.cfg

   boot system priority 9 \
       image /flash/staros.bin \
       config /flash/sftp/tb5_vnf1_dayN.cfg

   boot system priority 10 \
       image /flash/staros.bin \
       config /flash/system.cfg
   ```

4. Enter the configuration mode to change the boot priority of new StarOS bin file.

   #config

   #boot system priority 1 image /flash/updated.bin config /flash/system.cfg

   #end

5. Verify the new boot priority.

   #show boot

   ```
   boot system priority 1 \

       image /flash/updated.bin \

       config /flash/system.cfg

   boot system priority 10 \

       image /flash/staros.bin \

       config /flash/system.cfg
   ```

6. Verify whether the flash directory contains the boot configuration and new bin.

```
dir /flash
```

```
total 320376
-rw-rw-r-- 1 root root 134 May 3 10:11 boot.sys
-rw-rw-r-- 1 root root 3920672 May 11 19:49 crashlog2
drwxrwxr-x 2 root root 4096 May 11 19:49 crsh2
-rw-rw-r-- 1 root root 156 May 11 19:49 module.sys
drwxrwxr-x 3 root root 4096 May 11 19:49 patch
drwxrwxr-x 2 root root 4096 May 11 19:49 persistdump
-rw-rw-r-- 1 root root 79 May 11 19:49 restart_file_cntr.txt
drwxrwxr-x 3 root root 4096 May 11 20:07 sftp
-rw-rw-r-- 1 root root 160871936 May 3 10:11 staros.bin
-rw-rw-r-- 1 root root 5199 May 11 19:57 system.cfg
-rw-rw-r-- 1 root root 163227136 May 11 20:07 updated.bin
320476 /flash
Filesystem 1K-blocks Used Available Use% Mounted on
/var/run/storage/boot1/part2
4112620 320476 3792144 8% /mnt/user/.auto/onboard/flash
```

Synchronize File Systems

To synchronize the file systems:

1. Log on to the VNF using the previously fetched VIP address in the Obtain VIP Addresses for AutoVNF, CF, ESC and UEM, on page 170.
2. Synchronize the local file systems on the management cards by entering the following command:

   ```
   [local]host_name# filesystem synchronize all
   ```

   Example output:

   ```
   Updating /flash/system.cfg

   ********************************************************************************

   Updating /flash/updated.bin

   ********************************************************************************

   Updating /flash/sftp/yang/cisco-staros-bulkstats-config.yang

   ********************************************************************************

   Updating /flash/sftp/yang/cisco-staros-bulkstats-schema-types.yang

   ********************************************************************************

   Updating /flash/sftp/yang/cisco-staros-bulkstats.yang

   ********************************************************************************

   Updating /flash/sftp/yang/cisco-staros-cli-config.yang

   ********************************************************************************

   Updating /flash/sftp/yang/cisco-staros-confd-config.yang

   ********************************************************************************

   Updating /flash/sftp/yang/cisco-staros-config.yang
   ```

```
*******************************************************************************
Updating /flash/sftp/yang/cisco-staros-exec.yang
*******************************************************************************
Updating /flash/sftp/yang/cisco-staros-kpi.yang
*******************************************************************************
Updating /flash/sftp/yang/cisco-staros-notif.yang
*******************************************************************************
Updating /flash/boot.sys
*******************************************************************************
12 updated on card 2
           /flash/system.cfg
           /flash/updated.bin
           /flash/sftp/yang/cisco-staros-bulkstats-config.yang
           /flash/sftp/yang/cisco-staros-bulkstats-schema-types.yang
           /flash/sftp/yang/cisco-staros-bulkstats.yang
           /flash/sftp/yang/cisco-staros-cli-config.yang
           /flash/sftp/yang/cisco-staros-confd-config.yang
           /flash/sftp/yang/cisco-staros-config.yang
           /flash/sftp/yang/cisco-staros-exec.yang
           /flash/sftp/yang/cisco-staros-kpi.yang
           /flash/sftp/yang/cisco-staros-notif.yang
           /flash/boot.sys
```

Reboot the System

To reboot the system (VNF):

1. Log on to the VNF using the previously fetched VIP address in the Obtain VIP Addresses for AutoVNF, CF, ESC and UEM, on page 170 .

   ssh ubuntu@<*vip-addr*>

   Command example:

   ssh ubuntu@192.168.100.38

2. Enter the following Exec mode command:

   [local]*host_name*# **reload [-noconfirm]**

As the system reboots, it loads the new operating system software image and its corresponding CLI configuration file using the new boot stack entry configured earlier.

3. *Optional for PDSN:* If you are using the IP Pool Sharing Protocol during your upgrade, refer to *Configuring IPSP Before the Software Upgrade* in the *PDSN Administration Guide*.

4. After the reload is complete, log on to the VNF and make sure it is loaded with the intended StarOS version and all the cards have booted up and are in active or stand-by state as expected.

   show version

   Example output:

```
Active Software:
  Image Version:                  21.9.0.69977
  Image Build Number:             69977
  Image Description:              Build
  Image Date:                     Mon Jul 30 06:48:34 EDT 2018
  Boot Image:                     /flash/updated.bin
  Source Commit ID:               abde005a31c93734c89444b8aec2b6bb2d2e794d
```

   show card table

   Example output:

```
Slot         Card Type                                Oper State     SPOF  Attach
-----------  ---------------------------------------  -------------  ----  ------
 1: CFC      Control Function Virtual Card            Active         No
 2: CFC      Control Function Virtual Card            Standby        -
 3: FC       4-Port Service Function Virtual Card     Standby        -
 4: FC       4-Port Service Function Virtual Card     Standby        -
 5: FC       4-Port Service Function Virtual Card     Standby        -
 6: FC       4-Port Service Function Virtual Card     Standby        -
 7: FC       4-Port Service Function Virtual Card     Standby        -
 8: FC       4-Port Service Function Virtual Card     Standby        -
 9: FC       4-Port Service Function Virtual Card     Standby        -
10: FC       4-Port Service Function Virtual Card     Standby        -
```

5. Run the following Exec mode command to display additional information about the running StarOS build release.

   show build

6. *Optional*. Verify the operational state of CF and SF VNFCs.

Note This step is relevant only for the upgrade of CF and SF VNFCs.

   a. Repeat the steps in *Verify Zookeeper Database* and *Verify ESC Database* sections.

   b. Log on to the UEM using either the floating IP or from the AutoVNF using the UEM VIP.

      ssh ubuntu@*<vip-addr>*

      Command example:

      ssh ubuntu@192.168.100.21

   c. Become the *root* user.

      sudo -i

d. Collect the UEM orchestration IP address for Zookeeper database connection.

#ifconfig

```
eth0      Link encap:Ethernet  HWaddr fa:16:3e:71:1d:08

          inet addr:192.168.200.12  Bcast:192.168.200.255  Mask:255.255.255.0
```

e. Navigate to the */opt/cisco/usp/packages/zookeeper/<current>/bin* directory.

f. Run Zookeeper tool to access the UEM Zookeeper database.

zkCli.sh -server *<vip-addr>:port_num*

Command example:

zkCli.sh -server 192.168.200.12:2181

Make sure there are no outstanding requests between UEM and CF.

g. Verify the "state": "alive" for each of the CFs and SFs using the following commands:

get /oper/vnfs/*vnf_name***/vdus/***vdu_name***/cf1**

get /oper/vnfs/*vnf_name***/vdus/***vdu_name***/cf2**

get /oper/vnfs/*vnf_name***/vdus/***vdu_name***/sf1**

get /oper/vnfs/*vnf_name***/vdus/***vdu_name***/sf2**

Command examples:

get /oper/vnfs/tb1-autovnf1_vpc-vpc-core/vdus/vdu-cf1/cf1

get /oper/vnfs/tb1-autovnf1_vpc-vpc-core/vdus/vdu-cf1/cf2

get /oper/vnfs/tb1-autovnf1_vpc-vpc-core/vdus/vdu-sf1/sf1

get /oper/vnfs/tb1-autovnf1_vpc-vpc-core/vdus/vdu-sf1/sf2

h. Look for the state Alive in the console output.

```
zk: localhost:2181(CONNECTED) 2] get
/oper/vdus/control-function/BOOT_generic_di-chassis_CF1_1
{"id":"BOOT_generic_di-chassis_CF1_1","state":"alive","vnfcId":"cf-vnfc-di-chassis","uuid":"c4",
"host":"tb5-ultram-osd-compute-2.localdomain","vimId":"523b921c-7266-4fd5-90bb-5157cffc6951",
"cpts":[{"cpid":"di_intf1","state":"alive","subnet":"6102e9b5-8555-41f5-8cdc-0b47d30a6f7a",
"netmask":"255.255.255.0","dhcp":true,"vl":"vl-vnf1-DI-INTERNAL1-CAT","vnfc":"cf-vnfc-di-chassis",
"port_id":"19539aea-edbf-4acf-a57c-af5627d859ea","ip_address":"192.168.10.3",
"mac_address":"fa:16:3e:19:80:ed","network":"0d72f553-5a9c-4904-b3ea-83371a806e23"},
{"cpid":"di_intf2","state":"alive","nicid":1,"subnet":"30002d02-761d-4ccb-8a9e-d6188cdf54a3",
"netmask":"255.255.255.0","dhcp":true,"vl":"vl-vnf1-DI-INTERNAL2-CAT","vnfc":"cf-vnfc-di-chassis",
"port_id":"ff1da1e1-ecf3-477d-98b7-398c3c77fc8d","ip_address":"192.168.11.13",
"mac_address":"fa:16:3e:89:88:23","network":"9f109c0a-b1e7-4d90-a746-5de4ab8ef536"},
{"cpid":"orch","state":"alive","nicid":2,"subnet":"729e9dd2-3c75-43eb-988a-769016f2f44c",
"netmask":"255.255.255.0","dhcp":true,"vl":"vl-vnf1-UAS-ORCH-CAT","vnfc":"cf-vnfc-di-chassis",
"port_id":"81370948-f686-4812-820c-20ec5d3d3cdd","ip_address":"172.168.11.17","mac_address":"fa:16:3e:1d:0b:56",
"network":"9a286170-e393-4ba5-abce-147a45fb337a"},{"cpid":"mgmt","state":"alive","nicid":3,
"subnet":"9778a11b-1714-4e84-bbc2-86c84b111e8e","netmask":"255.255.255.0","dhcp":true,"vl":"vl-vnf1-UAS-MGMT-CAT",
"vnfc":"cf-vnfc-di-chassis","port_id":"6130cbb4-3dd8-4822-af90-50dac98f2f0d",
"ip_address":"172.168.10.17","mac_address":"fa:16:3e:42:92:47","network":"e278b524-e9a9-48c1-a45b-956a8c3ea583"}],
"monitor":true,"vduId":"control-function"}
cZxid = 0x100000051
ctime = Fri May 18 19:04:40 UTC 2018
mZxid = 0x10000024a
```

```
mtime = Mon May 21 17:48:19 UTC 2018
pZxid = 0x100000051
cversion = 0
dataVersion = 12
aclVersion = 0
ephemeralOwner = 0x0
dataLength = 1625
numChildren = 0
```

Note You can use use **CTRL+D** to exit the zookeeper CLI.

i. From the UEM VM as a root user, log on to the *ncs_cli* and check for devices live status.

~$ sudo -i

ncs_cli -C -u admin

 # show devices device *device_name* **live-status**

Verify that the command output reflects the correct 'state' and 'card-state' of each card.

Example output:

 # show devices device tb1-autovnf1_vpc-vpc-core-cf-nc live-status

```
<snip>

VNFC  CURRENT  VNFC  VDU   CARD  CARD  NUMBER  CPU   DISK  START  UPTIME  NOVA  ID    DATE  FROM  TO
REF   STATE    INSTANCE REF TYPE  SLOT  OF      UTILIZATION SPACE TIME       LAUNCH      AND   STATE STATE
       ID               ID    CORES                                CMD        TIME
-------------------------------------------------------------------------------------------------------
cf1   -    cf1   cf    control 1     -    -    -    -    -    -
cf2   -    cf2   cf    control 2     -    -    -    -    -    -
sf1   -    sf1   sf    service 3     -    -    -    -    -    -
sf2   -    sf2   sf    service 4     -    -    -    -    -    -


live-status vnfd sj-autovnf-vpc-abc

 version            6.0

 vnfm vim-tenant-name abc

 vnfm tenant-name abc

 vnfm ipaddr     192.168.100.22

 vnfm port       830

 vnfm username   ubuntu

 vnfm password   "$4$+HLzhFFzHq66nqtTsc00CfiODYHqlUSVmkn1tRe1f84byNakWEa9sJ8sY/
cwfFME3aG0UaBC\nvvNNAMkuXQI9Ksfu5IiQQ9ViWbbHwl6IEFQ="

 virtual-link vl-di-internal1

  auto-vnf-connection-ref di-internal1
```

```
 virtual-link vl-management

  auto-vnf-connection-ref management

 virtual-link vl-orchestration

  auto-vnf-connection-ref orchestration

 virtual-link vl-abc-vpc-svc

  auto-vnf-connection-ref sj-autovnf-abc-vpc-svc

 vdu cf

  ssh-keygen         false

  vm-image           076c887a-a12c-4a0b-b4d6-b2d213f64b9e

  lifecycle-event-initialization staros_config.txt

   source-url http://192.168.100.9:5000/config/sj-autovnf-vpc-abc/cf/staros_config.txt


  lifecycle-event-initialization staros_param.cfg

   source-url http://192.168.100.9:5000/config/sj-autovnf-vpc-abc/cf/staros_param.cfg


  ned cisco-staros-nc

   user        "$4$+HLzsE1kLJOeuoFWyOsmBWY2LHjOi2WtJdKy/OIux7YHhsNY/
O8hnA9/WwWuFD5trHrW3ZHs\nLo4TfiAKqYwxdNKqFYyoTxH2hrLJV5DgwmE="

   password
"$4$+HLzsXtCHJ2vsYZD5s0RGtBRY/dHDU1mgHJX7wCt3o1DMtQZqpBLDcNSJumC7n5rnkVxwI1s\
ncJYeCOFLrqpLHXm3xtXyMdtT7WVzvRMtdao="

   netconf

   port-number 830

  card-type          control-function

  usp-auto-vnf-id    sj-autovnf-vpc-abc-cf

  vnfc cf-vnfc-ugp

<snip>
```

Restoring the Previous Software Image

If for some reason you need to undo the upgrade, perform the upgrade again except:

- Specify the locations of the upgrade software image and configuration files.

then

- Specify the locations of the original software image and configuration files.

Managing License Keys

License keys define capacity limits (number of allowed subscriber sessions) and available features on your system. Adding new license keys allows you to increase capacity and add new features as your subscriber base grows.

New System License Keys

New systems are delivered with no license keys installed. In most cases, you receive the license key in electronic format (usually through e-mail).

When a system boots with no license key installed a default set of restricted session use and feature licenses is installed. The following Exec Mode command lists the license information:

```
[local]host_name# show license information
```

Important With no license key installed, the session use licenses for PDSN, HA, GGSN, and L2TP LNS are limited to 10,000 sessions.

Session Use and Feature Use Licenses

Session use and feature use licenses are software mechanisms that provide session limit controls and enable special features within the system. These electronic licenses are stored in the system's configuration file that is loaded as part of the system software each time the system is powered on or restarted.

- Session use licenses limit the number of concurrent sessions that a system is capable of supporting per service type and are acquired on an as-needed basis. This allows carriers to pay only for what they are using and easily increase capacity as their subscriber base grows.
- Feature use licenses enable specific features/functionality within the system and are distributed based on the total number of sessions supported by the system.

Installing New License Keys

Use the instructions below to install a new license key.

Cutting and Pasting the Key

If you have a copy of the license, use the following configuration to cut and paste just the license key part:

Step 1 From the Exec mode, enter the following:

```
configure
   license key license
   exit
```

license is the license key string. The license can be an alphanumeric string of 1 through 1023 characters that is case sensitive. Copy the license key as shown in the example below, including the "\ (double-quote slash). Please note: this is not a functional license.

```
"\
VER=1|C1M=000-0000-00|C1S=03290231803|C2M=11-1111-11-1|C2S=\
STCB21M82003R80411A4|DOI=0000000000|DOE=00000000|ISS=1|NUM=13459|0000000000000|
LSP=000000|LSH=000000|LSG=500000|LSL=500000\|FIS=Y|FR4=Y|FPP=Y|FCS=Y|FTC=Y|FMG=Y|
FCR=Y|FSR=Y|FPM=Y|FID=Y|SIG=MCwCF\Esnq6Bs/
XdmyfLe7rHcD4sVP2bzAhQ3IeHDoyyd6388jHsHD99sg36SG267gshssja77
end
```

Step 2 Verify that the license key just entered was accepted by entering the following command at the Exec mode prompt:

```
[local]host_name# show license key
```

The new license key should be displayed. If it is not, return to the Global configuration mode and re-enter the key using the **license key** command.

Important An invalid license will not be accepted. A Failure error will appear in the output of the **license key** command when you attempt to configure an invalid license key. If you use the **-force** option to install an invalid license key, the license will be placed into a 30-day grace period. StarOS will generate daily syslog error messages and SNMP traps during the grace period. The output of the **show license information** command will indicate "License State" as "Not Valid".

Step 3 Verify that the license key enabled the correct functionality by entering the following command:

```
[local]host_name# show license information
```

All license keys and the new session capacity or functionality enabled should be listed. If the functionality or session capacity enabled by the new key is incorrect, please contact your service representative.

Step 4 Save your configuration as described in the *Verifying and Saving Your Configuration* chapter.

Caution Failure to save the new license key configuration in the current CLI configuration file will result in the loss of any of the new features enabled by the license key once the system is reloaded.

Adding License Keys to Configuration Files

License keys can be added to a new or existing configuration file.

Important License key information is maintained as part of the CLI configuration. Each time a key is installed or updated, you must re-save the configuration file.

Step 1 Open the configuration file to which the new license key commands are to be copied.

Step 2 Copy the license as shown in the example, including the "\ (double-quote slash). Please note: this is not a functional license.

```
"\
VER=1|C1M=000-0000-00|C1S=03290231803|C2M=11-1111-11-1|C2S=\STCB21M82003R80411A4|
DOI=0000000000|DOE=00000000|ISS=1|NUM=13459|0000000000000|LSP=000000|LSH=000000|
LSG=500000|LSL=500000\|FIS=Y|FR4=Y|FPP=Y|FCS=Y|FTC=Y|FMG=Y|FCR=Y|FSR=Y|FPM=Y|FID=Y|
```

```
SIG=MCwCF\Esnq6Bs/XdmyfLe7rHcD4sVP2bzAhQ3IeHDoyyd6388jHsHD99sg36SG267gshssja77
end
```

Step 3 Paste the license key into the configuration

Important Paste the license key information at the beginning of the configuration file to ensure the system has the expected capacity and features before it configures contexts.

Step 4 Save your configuration as described in the *Verifying and Saving Your Configuration* chapter.

License Expiration Behavior

When a license expires, there is a built-in grace period of 30 days that allows normal use of the licensed session use and feature use licenses. This allows you to obtain a new license without any interruption of service.

The following Exec mode command lists the license information including the date the grace period is set to expire:

show license information

Requesting License Keys

License keys for the system can be obtained through your Cisco account representative. Specific information is required before a license key may be generated:

- Sales Order or Purchase Order information
- Desired session capacity
- Desired functionality

Viewing License Information

To see the license detail, enter the following command from the Exec mode:

[local]*host_name*# **show license information [full | key [full]]**

Activating New License Keys

To activate new license keys, you must reboot the active CF by exciting the Exec mode **reload** command.

Note All SFs are rebooted when the **reload** command is executed on the active CF.

Deleting a License Key

Use the procedure below to delete the session and feature use license key from a configuration. You must be a security administrator or administrator.

```
configure
  no license key
  exit
show license key
```

The output of this command should display: "No license key installed".

Managing Local-User Administrative Accounts

Unlike context-level administrative accounts which are configured via a configuration file, information for local-user administrative accounts is maintained in a separate file in flash memory and managed through the software's Shared Configuration Task (SCT). Because local-user accounts were designed to be compliant with ANSI T1.276-2003, the system provides a number of mechanisms for managing these types of administrative user accounts.

For additional information, see Disable AAA-based Authentication for Console, on page 119 and Limit local-user Login on Console/vty Lines, on page 120.

Configuring Local-User Password Properties

Local-user account password properties are configured globally and apply to all local-user accounts. The system supports the configuration of the following password properties:

- **Complexity:** Password complexity can be forced to be compliant with ANSI T1.276-2003.
- **History length:** How many previous password versions should be tracked by the system.
- **Maximum age:** How long a user can use the same password.
- **Minimum number of characters to change:** How many characters must be changed in the password during a reset.
- **Minimum change interval:** How often a user can change their password.
- **Minimum length:** The minimum number of characters a valid password must contain.
- **Expiry warning:** Password expiry warning interval in days.
- **Auto-generate:** Automatically generates password with option to specify length of password.

Refer to the **local-user password** command in the *Global Configuration Mode Commands* chapter of the *Command Line Interface Reference* for details on each of the above parameters.

Configuring Local-User Account Management Properties

Local-user account management includes configuring account lockouts and user suspensions.

Local-User Account Lockouts

Local-user accounts can be administratively locked for the following reasons:

- **Login failures:** The configured maximum login failure threshold has been reached. Refer to the **local-user max-failed-logins** command in the *Global Configuration Mode Commands* chapter of the *Command Line Interface Reference* for details
- **Password Aging:** The configured maximum password age has been reached. Refer to the **local-user password** command in the *Global Configuration Mode Commands* chapter of the *Command Line Interface Reference* for details.

Accounts that are locked out are inaccessible to the user until either the configured lockout time is reached (refer to the **local-user lockout-time** command in the *Global Configuration Mode Commands* chapter of the *Command Line Interface Reference*) or a security administrator clears the lockout (refer to the **clear local-user** command in the *Exec Mode Commands* chapter of the *Command Line Interface Reference*).

Important Local-user administrative user accounts could be configured to enforce or reject lockouts. Refer to the **local-user username** command in the *Global Configuration Mode Commands* chapter of the *Command Line Interface Reference* for details.

Local-User Account Suspensions

Local-user accounts can be suspended as follows:

```
configure
  suspend local-user name
```

A suspension can be removed by entering:

```
configure
  no suspend local-user name
```

Changing Local-User Passwords

Local-user administrative users can change their passwords using the **password change** command in the Exec mode. Users are prompted to enter their current and new passwords.

Security administrators can reset passwords for local-users by entering the following command from the root prompt in the Exec mode:

```
[local]host_name# password change username name
```

name is the name of the local-user account for which the password is to be changed. When a security administrator resets a local-user's password, the system prompts the user to change their password the next time they login.

All new passwords must adhere to the password properties configured for the system.

Resetting, Stopping, Starting or Deleting VMs in the VPC-DI Instance

You can create a script that with reset, stop, start or delete one or more VMs in the VPC-DI instance using the hypervisor.

KVM

You can create a KVM script that runs a series of commands to reset, stop, start, or delete specified VMs.

The actual script will contain command sequences appropriate to the installation requirements of the VPC-DI instance as deployed in your site.

Script Commands

Available commands include:

```
 reset
for CARD in {01,02}-cf {03,04,05,06...32}-sf do
  VMNAME={NAME}{INST}-{CARD}
  virsh reset {VMNAME}
done

 stop
for CARD in {01,02}-cf {03,04,05,06...32}-sf do
  VMNAME={NAME}{INST}-{CARD}
  virsh destroy {VMNAME}
done

 start
for CARD in {01,02}-cf {03,04,05,06...32}-sf do
  VMNAME={NAME}{INST}-{CARD}
  virsh start {VMNAME}
done

 delete
for CARD in {01,02}-cf {03,04,05,06...32}-sf do
  VMNAME={NAME}{INST}-{CARD}
  virsh undefine {VMNAME}
done
```

Sample Script

A sample script is provided below:

```
cat ./vm-start

 start local
for CARD in {01,02}-cf {03,04,05,06,17,18,19,20}-sf do
  VMNAME=BLADE5-DI-{CARD}
  virsh start {VMNAME}
done

 start remote
ssh testbed3 '/home/luser/vm-start-BLADE5-DI'

roottestc1b5:/home/luser cat ./vm-stop

 reset local
for CARD in {01,02}-cf {03,04,05,06,17,18,19,20}-sf do
  VMNAME=BLADE5-DI-{CARD}
  virsh destroy {VMNAME}
done

 reset remote
ssh testbed3 '/home/luser/vm-stop-BLADE5-DI'
```

VMware ESXi

VSphere GUI

To reset VMs in a VPC-DI instance you can use the vSphere GUI.

Step 1 From vSphere select the host for the target VPC-DI VMs.

Step 2 Select **Power Off** from the Commands list to shut down the VM(s).

Step 3 Select **Power On** from the Commands list to start the VM(s).

Step 4 If multiple vSphere hosts are used to provision the VMs in this VPC-DI instance, repeat steps 2 and 3 for each host.

PowerShell Script

You can also create a PowerShell script that uses the commands described in XREF to restart target VMs.

CHAPTER 13

Smart Licensing

Feature Summary and Revision History

Summary Data

Applicable Product(s) or Functional Area	All
Applicable Platform(s)	ASR 5500 VPC-SI VPC-DI
Feature Default	Disabled - Configuration Required
Related Changes in This Release	Not Applicable
Related Documentation	• *ASR 5500 System Administration Guide* • *Command Line Interface Reference* • *VPC-DI System Administration Guide* • *VPC-SI System Administration Guide*

Revision History

Revision Details	Release
Added Smart Licensing support for On/Off feature licenses.	21.6
First introduced.	21.3

Smart Software Licensing

Overview

StarOS 21.3 and higher releases support Smart Licensing. Smart Licensing is a cloud-based approach to licensing that simplifies the purchase, deployment, and management of Cisco software assets. Entitlements are purchased through your Cisco account via Cisco Commerce Workspace (CCW) and immediately deposited into your Virtual Account for usage. This eliminates the need to install license files on every device. Products that are smart enabled communicate directly to Cisco to report consumption. A single location is available to customers to manage Cisco software licenses—the Cisco Smart Software Manager (CSSM). License ownership and consumption are readily available to help make better purchase decision based on consumption or business need.

See https://www.cisco.com/c/en/us/buy/smart-accounts/software-licensing.html for more information about Cisco Smart Licensing.

Comparison Between Legacy Licensing and Smart Licensing

Cisco employs two types of license models - Legacy Licensing and Smart Software Licensing. **Legacy Licensing** consists of software activation by installing Product Activation Keys (PAK) on to the Cisco product. A Product Activation Key is a purchasable item, ordered in the same manner as other Cisco equipment and used to obtain license files for feature set on Cisco Products. **Smart Software Licensing** is a cloud based licensing of the end-to-end platform through the use of a few tools that authorize and deliver license reporting. Smart Software Licensing functionality incorporated into StarOS complete the product registration, authorization resulting in reporting services available to the end customer.

Evalutation Period

A 90 day evalutation period is granted for all licenses in use. During this period feature licenses can be used without limitation, and up to one counting license each can be used. The evaluation period ends when the system registers successfully with the CSSM or Cisco.com. Licensed functionality is blocked when this 90 day period expires.

Smart Licensing in Release 21.6

In Release 21.6 and higher, Smart Licensing supports on/off feature licenses. In previous releases Smart Licensing only supported counting licenses.

StarOS performs license enforcement for on/off feature licenses. Each on/off feature license is tied to service licenses, which potentially use those on/off features. When an Out of Compliance (OOC) is detected for an on/off license, new calls for the corresponding services will be dropped, subject to the following conditions:

- Each on/off feature license is given a 90 day grace (evaluation) period. During this period, the system generates SNMP traps to inform of the unavailability of valid licenses. To resolve the OOC, corrective action is needed such as purchasing and registering licenses for this feature, or disabling the feature.
- If the feature is still OOC after the 90 day grace period, StarOS enforces the OOC state based on a predefined policy for each license. If enforcement is required, new calls for the services corresponding to the on/off licenses are dropped.

Two new commands have been introduced to display details about the enforcement of Smart Licenses in use:

```
show license enforcement policy
show license enforcement status [ allowed | blocked ] [ feature | service
 ]
```

Smart Licensing in Release 21.3

When Smart Licensing is enabled in Release 21.3, it operates in mixed mode where both Legecy and Smart Licensing is used.

- Legacy (PAK-based) licenses continue to be used for On/Off features.
- Smart licensing is used for counting each service. In Release 21.3 only the counting licenses listed below in "Entitlement Tags" are controlled by Smart Licensing. All other counting licenses use Legacy (PAK-based) licenses.

Cisco Smart Software Manager

Cisco Smart Software Manager (CSSM) enables the management of software licenses and Smart Account from a single portal. The interface allows you to activate your product, manage entitlements, and renew and upgrade software. A functioning Smart Account is required to complete the registration process. To access the Cisco Smart Software Manager, see https://software.cisco.com.

Smart Accounts/Virtual Accounts

A Smart Account provides a single location for all Smart-enabled products and entitlements. It helps speed procurement, deployment, and maintenance of Cisco Software. When creating a Smart Account, you must have the authority to represent the requesting organization. After submitting, the request goes through a brief approval process.

A Virtual Account exists as a sub-account withing the Smart Account. Virtual Accounts are a customer-defined structure based on organizational layout, business function, geography or any defined hierarchy. They are created and maintained by the Smart Account administrator.

See https://software.cisco.com to learn about, set up, or manage Smart Accounts.

Request a Cisco Smart Account

A Cisco Smart Account is an account where all products enabled for Smart Licensing are deposited. A Cisco Smart Account allows you to manage and activate your licenses to devices, monitor license use, and track Cisco license purchases. Through transparent access, you have a real-time view into your Smart Licensing products. IT administrators can manage licenses and account users within your organization's Smart Account through the Smart Software Manager.

Step 1 In a browser window, enter the following URL:

```
https://software.cisco.com
```

Step 2 Log in using your credentials, and then click **Request a Smart Account** in the **Administration** area.

The **Smart Account Request** window is displayed.

Step 3 Under **Create Account**, select one of the following options:

- **Yes, I have authority to represent my company and want to create the Smart Account** – If you select this option, you agree to authorization to create and manage product and service entitlements, users, and roles on behalf of your organization.
- **No, the person specified below will create the account** – If you select this option, you must enter the email address of the person who will create the Smart Account.

Step 4 Under **Account Information**:

a) Click **Edit** beside **Account Domain Identifier**.

b) In the **Edit Account Identifier** dialog box, enter the domain, and click **OK**. By default, the domain is based on the email address of the person creating the account and must belong to the company that will own this account.

c) Enter the **Account Name** (typically, the company name).

Step 5 Click **Continue**.

The Smart Account request will be in pending status until it has been approved by the Account Domain Identifier. After approval, you will receive an email confirmation with instructions for completing the setup process.

Software Tags and Entitlement Tags

Tags for the following software and entitlements have been created to identify, report, and enforce licenses.

Software Tags

Software tags uniquely identify each licenseable software product or product suite on a device. The following software tags exist for the StarOS.

Product Type / Description	Software Tag
ASR5500 ASR-5500 Multimedia Core Platform	regid.2017-02.com.cisco.ASR5500,1.0_401f2e9e-67fd-4131-b61d-6e229d13a338
VPC_SI Virtualized Packet Core (Single instance)	regid.2017-02.com.cisco.VPC_SI,1.0_dcb12293-10c0-4e90-b35e-b10a9f8bfac1
VPC_DI Virtualized Packet Core (Distributed instance)	regid.2017-02.com.cisco.VPC_DI,1.0_5cb68f91-c1d6-48d6-9482-e9750203f5e6

Entitlement Service Tags

The following entitlement tags indentify licenses in use for each service type:

TagId	Service Type / Description	Entitlement Tag
0	ASR5K-00-PW10GTWY PDN GW Bundle, 10K Sessions	regid.2017-02.com.cisco. ASR5K-00-PW10GTWY, 1.0_85a577a1-017d-4dc6-8bf7-4fac7c4d465a
1	ASR5K-00-SG10 SAE GW Bundle, 10K Sessions	regid.2017-02.com.cisco. ASR5K-00-SG10, 1.0_e87d1e76-e34c-4699-94ad- 68eafa2da8b2

TagId	Service Type / Description	Entitlement Tag
2	ASR5K-00-GN10SESS GGSN Software License, 10K sessions	regid.2017-02.com.cisco. ASR5K-00-GN10SESS, 1.0_bf7eb1d7-35bf- 40c6-8763-a4770f5ebf64
3	ASR5K-00-SW10LIC Serving GW Bundle, 10K Sessions	regid.2017-02.com.cisco. ASR5K-00-SW10LIC, 1.0_ca9d09ee- 5e72-4695-97ef-d5117037cfb4
4	ASR5K-00-ME10LIC MME Bundle, 10K sessions	regid.2017-02.com.cisco. ASR5K-00-ME10LIC, 1.0_ 37a452c9-ff23-41fc-b96c-c8b2eb09254b
5	ASR5K-00-EG10S-K9 EPDG, 10k Sessions	regid.2017-02.com.cisco. ASR5K-00-EG10S-K9, 1.0_5c625d0f-8b2c-4c44-970b- 9668a072890f
6	ASR5K-00-SM10S-K9 SaMOG GW 10k Sessions	regid.2017-02.com.cisco. ASR5K-00-SM10S-K9, 1.0_289f4807-fb6e-4750-b047- a9d57e7b045f

Entitlement Feature Tags

The following entitlement tags indentify licenses in use for each feature type:

TagId	Feature / Description	Entitlement Tag
0	ASR5K-00-CS01GRET GRE Interface Tunnelling	regid.2017-11.com.cisco. ASR5K-00-CS01GRET, 1.0_284ba242-a2ef-4ad0-b445-b62ed3887d3b
1	ASR5K-00-CS01MPLS MPLS on StarOS	regid.2017-11.com.cisco. ASR5K-00-CS01MPLS, 1.0_1b1221cd-39de-4113-8709-dad022c0a063
2	ASR5K-00-GN01REC Session Recovery	regid.2017-11.com.cisco. ASR5K-00-GN01REC, 1.0_88542f00-077b-4ed0-9698-a3cec699a047
3	ASR5K-00-GN01ICSR Inter-Chassis Session Recovery	regid.2017-11.com.cisco. ASR5K-00-GN01ICSR, 1.0_d0bca3e3-1218-4e91-98f6-207a20a5401a
4	ASR5K-00-EP01VLE EPC Gw VoLTE enhancements	regid.2017-11.com.cisco. ASR5K-00-EP01VLE, 1.0_a6c3d615-0b37-4f4a-8c34-3090ed5e265c
5	ASR5K-00-EP01ICSE ICSR Enhanced Recovery for Data and Control Plane	regid.2017-11.com.cisco. ASR5K-00-EP01ICSE, 1.0_bf87e837-003e-4832-b29f-ce0416699b01
6	ASR5K-00-PWXICSRP ICSR/SR Performance Improvements	regid.2017-11.com.cisco. ASR5K-00-PWXICSRP, 1.0_079a8bd2-a97d-4d12-bc5a-b0f70eec6c26

TagId	Feature / Description	Entitlement Tag
7	ASR5K-00-CS01I-K9 IPSec on StarOS	regid.2017-11.com.cisco. ASR5K-00-CS01I-K9, 1.0_a6e5539c-ab69-4303-b2a8-e71c03ac2a08
8	ASR5K-00-EG01RS ePDG Reselection	regid.2017-11.com.cisco. ASR5K-00-EG01RS, 1.0_b7f0b7d1-0d68-4223-abb1-381c3ac968ec

Configuring Smart Licensing

Before you begin, ensure you have:

- Created a Smart Licensing account on https://software.cisco.com.
- Registered your products on https://software.cisco.com using the Product Instance Registration tokens created as part of Smart Account/Virtual Account.
- Enabled a communication path between the StarOS system to the CSSM server or Cisco.com.

Enable Smart Licensing

By default, Smart Licensing is disabled in StarOS. To enable Smart Licensing, enter the following Config mode commands:

```
config
   license smart enable
   end
```

Enter the following command to verify the configuration:

```
show configuration | grep license
```

Register the Device with Cisco

Using the id Product Instance Registration token provided when you registered the products on https://software.cisco.com, register the system using the following Exec mode command:

```
license smart register idtoken token
```

The system now automatically report entitlement usage count to the CSSM server and receive a compliance status. This also removes the system from "Evaluation Mode".

To show the compliance status, enter any of the following Exec mode commands:

```
show license status
show license summary
show license statistics
```

The registration for the system is renewed automatically every 180 days. If needed, use the following Exec mode command to renew the registration information manually:

```
license smart renew id
```

The license authorization for the system is renewed automatically every 30 days. If needed, use the following Exec mode command to renew the license authorization manually:

```
license smart renew auth
```

To unregister a device, enter the following Exec mode command:

```
license smart deregister
```

Handling Out of Compliance

If there are not enough licenses in the virtual account for a given SKU, CSSM sends an Out Of Compliance (OOC) message to the device. The system stops allowing additional sessions until the OOC state is cleared. The OOC state is cleared when the device receives an authorized response.

Monitoring and Troubleshooting Smart Licensing

Enter the following Exec mode command to verify the Smart Licensing configuration:

```
show configuration | grep license
```

The following Exec mode commands display information about Smart Licensing:

```
show license { all | enforcement | eval-period | smart-tags | statistics
 | status | summary | tech-support | udi | usage } [ | { grep grep_options
 | more } ]
```

Notes:

- **all** - Shows a superset of information that includes show status, show usage, show UDI, as well as the Smart Licensing agent version.
- **enforcement { policy | status [allowed | blocked] [feature | service] }** - Shows the enforcement policy applied or current enforcement status of Smart Licenses. Status information can be filtered to show only the licenses which are currently allowed or blocked, or by type (feature license or service license).
- **eval-period** - Shows information about the evaluation period. Licenses are granted a 90 day evaluation period until they are registered.
- **smart-tags [feature | service]** - Shows the features and services that are currently supported and the corresponding Smart Entitlement Tag.
- **statistics** - Shows individual feature license status.
- **status** - Shows overall Smart Licensing status information.
- **summary** - Shows summary of Smart Licensing status.
- **tech-support** - Shows information useful for debugging issues with Smart Licensing.
- **udi** - Shows details for all Unique Device Identifiers (UDI).
- **usage** - Shows the usage information for all entitlements that are currently in use.

The following SNMP alarms (traps) are produced to track issues with Smart Licensing.

- 1349: starSmartLicenseServiceOOC / starSmartLicenseServiceOOCClear
- 1350: starSmartLicenseEvalMode / starSmartLicenseEvalModeClear

- 1351: starSmartLicenseEvalModeExpire / starSmartLicenseEvalModeExpireClear
- 1352: starSmartLicenseCSSMConntectionFail / starSmartLicenseCSSMConnectionFailClear

Note For more information regarding these SNMP MIB alarms, see the *SNMP MIB Reference*.

Smart Licensing Bulk Statistics

The following statistics are provided in support of Smart Licensing

- current_sl_state – Current state of Smart Licensing, eitherenabled or disabled.
- current_sl_mode – Current mode of Smart Licensing, either Evaluation or Registered.
- total_sl_enabled – Total number of times Smart Licensing has been enabled.
- total_sl_disabled – Total number of times Smart Licensing has been disabled.
- mode – Counters for each service type when product in Evaluation and Registered mode.
- tagid – Tagid for each service type.
- cur_call_policy – Current call policy for new sessions of each service type, either ALLOW or BLOCK.
- cur_call_count – Current number of sessions/calls counted for the entire product for a particular service type.
- max_call_count – Maximum number of sessions/calls counted for the entire product for a particular service type.
- last_lic_count – License count last reported to Cisco licensing (CSSM) for particular service type.
- max_lic_count – Maximum license count reported to Cisco licensing (CSSM) for particular service type up to this point in time.
- policy_change_block_calls – Number of times particular service type blocked new sessions as a result of insufficient licenses available.

CHAPTER 14

Monitoring the System

This chapter provides information for monitoring system status and performance using the **show** commands found in the Command Line Interface (CLI). These command have many related keywords that allow them to provide useful information on all aspects of the system ranging from current software configuration through call activity and status.

The selection of keywords described in this chapter is intended to provide the most useful and in-depth information for monitoring the system. For additional information on these and other **show** command keywords, refer to the *Exec Mode show Commands* chapter of the *Command Line Interface Reference*.

Note A VPC-DI or VPC-SI virtual machine (VM) has no knowledge of the hypervisor under which it is running or the commercial off-the-shelf (COTS) server. To monitor the status of the hypervisor and COTS server, refer to the user documentation supplied with these components of this system.

Important In Release 21.1 and forward, use the **do show** command to run all Exec Mode **show** commands while in Global Configuration Mode. It is not necessary to exit the Config mode to run a **show** command. The pipe character **|** is only available if the command is valid in the Exec mode.

SNMP Notifications

In addition to the CLI, the system supports Simple Network Management Protocol (SNMP) notifications that indicate status and alarm conditions. Refer to the *SNMP MIB Reference* for a detailed listing of these notifications.

Monitoring System Status and Performance

This section contains commands used to monitor the status of tasks, managers, applications and other software components in the system. Output descriptions for most of the commands are located in the *Statistics and Counters Reference*.

Table 10: System Status and Performance Monitoring Commands

To do this:	Enter this command:
View Administrative Information	
Display Current Administrative User Access	
View a list of all administrative users currently logged on the system	**show administrators**
View the context in which the administrative user is working, the IP address from which the administrative user is accessing the CLI, and a system generated ID number	**show administrators session id**
View information pertaining to local-user administrative accounts configured for the system	**show local-user verbose**
View statistics for local-user administrative accounts	**show local-user statistics verbose**
View information pertaining to your CLI session	**show cli**
Determining System Uptime	
View system uptime (time since last reboot)	**show system uptime**
View NTP Server Status	
View NTP servers status	**show ntp status**
View System Resources	
View all system resources such as CPU resources and number of managers created	**show resources [cpu]**
View System Alarms	
View information about all currently outstanding alarms	**show alarm outstanding all verbose**
View system alarm statistics	**show alarm statistics**
View Congestion-Control Statistics	
View Congestion-Control Statistics	**show congestion-control statistics**
View Remote Management Statistics	
View SNMP notification statistics	**show snmp notifies**
View SNMP access statistics	**show snmp accesses**
View SNMP trap history	**show snmp trap history**
View SNMP Trap Statistics	**show snmp trap statistics**

To do this:	Enter this command:
View Port Counters	
View datalink counters for a specific port	**show port datalink counters** *slot#/port#*
View Port Network Processor Unit (NPU) counters for a specific port	**show port npu counters** *slot#/port#*
View System Information and Network Interfaces	
View information about system components, storage devices and network interfaces	**show hardware**
View Card Information and Statistics	
View diagnostics for all cards or for a card in a specific slot/port; (for VPC, slot = VM)	**show card diag** *slot/port*
View detailed information for all cards or a card in a specific slot/port (for VPC, slot = VM)	**show card info** *slot/port*
View operating status for all cards or VMs	**show card table**
View the contents of the boot configuration (param.cfg) file [VPC-DI]	**show cloud configuration**
View information about installed hardware and whether it is optimal or not for a specific card or all cards in the system [VPC-DI]	**show cloud hardware**
View monitored statistics about the VPC-DI network relative to a specific card [VPC-DI]	**show cloud monitor di-network**

Important The commands or keywords/variables that are available are dependent on platform type, product version, and installed license(s).

Important Some commands have different outputs depending on the platform type.

Monitoring the DI Network

The DI network is the private L2 network that interconnects the VMs. The DI network transports user traffic from the received VM to the serving Session Manager on a different VM, and also transports CF to SF communications such as CLI commands, health checks, status changes. If the link is compromised unexpected things can happen (such as slow response to CLI commands), potentially resulting in service interruption.

The available monitoring capabilities to verify the health of the DI network are detailed here:

Inter-SF DI Network Tests

Each SF periodically sends non-blocking UDP test packets to each of other active and standby SFs, and keeps track of the responses to calculate latency and packet loss. Test packets are sent once per second. Both jumbo and non-jumbo test packets are sent alternately. A non-jumbo UDP test packet has a payload size of 200 bytes, and a jumbo test packet has a payload size of 4000 bytes. These statistics are recorded:

- Dropped packet counts—On receiving a test packet from another SF, the receiving SF sends back a reply. If an SF does not receive a test packet reply within one second, it marks the packet as dropped.
- Dropped jumbo packets—Same calculation as dropped packet counts, but only counts jumbo test packets.
- Number of packets with long latency—If the SF receives a test packet reply after 200 milliseconds, it marks the packet as having long latency.

Note Counters are cleared after an SF reboot.

The reporting interval starts at 15 seconds and can range to 3600 seconds. If there is no error detected during an interval, no warning log is generated and the reporting interval doubles until the interval is 3600 seconds. When an error is detected during an interval, a warning log is generated and the reporting interval is reduced in half until there are no more packets dropped.

If there are any packets lost or long latency counts, a WARNING event is generated. An example warning is shown here:

```
2016-Jan-10+22:00:01.477 [hat 3081 warning] [5/0/5146 <hatcpu:50> hatcpu.c:1307] [software

internal system syslog] Over the past 15 seconds, tests from card 5 to 4 had 1 total drops,

0 jumbo drops, 0 long latency.
```

Use the command **show heartbeat stats card** *cardnumber* **cpu** *cpunumber* to view the statistics collected regarding inter-SF communications.

DI network monitoring is enabled by default. Use the command **debug heartbeat test** to stop and start SF packet tests on specific SFs, or to clear test packet counters on a specific SF.

You can also use the command **show cloud monitor di-network** to display the DI network monitoring statistics. Sample output from the **show cloud monitor di-network summary** command is shown here for Card number 3:

```
Card 3 Test Results:

ToCard   Health   5MinLoss   60MinLoss
   1      Good       0.0%       0.0%
   2      Good       0.0%       0.0%
   4       Bad       6.32%      5.36%
   5      Good       0.0%       0.0%
   6      Good       0.0%       0.0%
```

The display shows the test packet loss rate for the past five minutes and past 60 minutes. If the rate is larger than 1%, the health status is marked as "Bad".

SF to Standby CF DI Network Tests

During an SF boot up, each SF sends both non-jumbo and jumbo ping packets to the standby CF to ensure that the standby CF is reachable.

During SF normal operation, the SF periodically sends non-blocking UDP test packets to the standby CF, and keeps track of the responses to calculate latency and packet loss. This functionality is the same as described for the *Inter-SF DI Network Tests*.

SF Secondary IP Address DI Network Tests

During an SF boot up, each SF sends both non-jumbo and jumbo ping packets to the active CF using the SF primary IP address. In addition, each SF also sends non-jumbo ping packets to active CF using each of its secondary IP addresses. If any of these pings fails, the SF notifies the active CF and the SF reboots.

Standby CF to Active CF DI Network Tests

During the standby CF boot up, the standby CF sends both non-jumbo and jumbo ping packets to the active CF.

DI-Network Bulk Statistics

The **mon-di-net** schema provides the following bulk statistics for monitoring the health of the DI-network on a VPC-DI platform. This information is similar to that provided in the output of the **show cloud monitor di-network summary** Exec mode command.

- src-card – Source card slot number on which monitoring has been performed.
- dest-card – Destination card slot number to which traffic was routed.
- total-pkts-5mins – Total number of packets sent over the past 5 minutes.
- total-drops-5mins – Total number of packets that were dropped over the past 5 minutes.
- total-pkts-60mins – Total number of packets sent over the past 60 minutes.
- total-drops-60mins – Total number of packets that were dropped over the past 60 minutes.
- total-pkts – Total number of all packets sent.
- total-pkts-jumbo – Total number of jumbo packets sent.
- total-drops – Total number of jumbo and non-jumbo test packets that were dropped.
- total-drops-jumbo – Number of jumbo test packets that were dropped.
- latency-warnings – Total number of times the latency has exceeded the threshold.
- long-rtt – Longest Round Trip Time (RTT) in milliseconds.
- average-rtt – Average Round Trip Time (RTT) in milliseconds.

The **mon-di-net** BulkStats Mode command configures the collection of statistics for the Mon-DI-Net schema.

See the *Bulk Statistics* chapter for information about configuring bulk statistic collection.

DI-Network Heartbeat Thresholds

This feature adds the capability to define thresholds for the internal DI-network for percentage heartbeat loss in order to monitor the card-to-card network health in a VPC-DI deployment.

When heartbeat loss (on any of the cards) crosses a set limit of threshold, this feature raises alarms/SNMP trap to indicate the loss.

The internal High Availability Task (HAT) tracks the percentage heartbeat loss over the past 5 minutes and past 60 minutes between cards and can generate SNMP alarms if a threshold has been crossed or a previous alarm has been cleared.

There can be multiple cards in the system and any card can raise this same trap ID but with different card information.

The scope of this funtionality is across the system. It is not specific to any service and is configured at the Global Configuration mode.

See Configure DI-Network Heartbeat Thresholds, on page 219 for instructions to enable this feature.

Monitor VPC-DI Network

Feature Summary and Revision History

Summary Data

Applicable Product(s) or Functional Area	All
Applicable Platform(s)	VPC - DI
Feature Default	Enabled - Always-on
Related Changes in This Release	Not Applicable
Related Documentation	• *Command Line Interface Reference* • *VPC-DI System Administration Guide* • *Statistics and Counters Reference*

Revision History

Revision Details	Release
First introduced.	Pre 21.8

Feature Description

In a DI-network, packet loss occurs when the DI-network ports are saturated or the underlying network infrastructure is unreliable. The Monitor VPC-DI network feature enables the identification and quantification of Control Plane and Data Plane packet loss on the VPC-DI system.

VPC-DI collects and aggregates the Control Plane and Data Plane monitor data for use in CLI reports and threshold alarms.

The feature also provides the ability to set the criteria for the VPC-DI to declare a card fault. Currently, a card fault is raised when a fixed number of High Availability Task (HAT) Control Plane heartbeats between the Active CF and an SF cards are consecutively bounced. The number of consecutive misses can be configured using this feature. This feature adds a secondary Data Plane configuration parameter that can be used to effectively discriminate between the DI-network packet loss and packet processing failure scenarios.

How It Works

The Control Plane and Data Plane monitors generate two fundamental DI-network traffic types on a fixed or recurring basis and tracks losses. The tracking data is meant to provide a view into DI-network communication loss or disruption.

Control Plane packets are typically unicast bi-directional UDP/TCP streams between cards; essentially request and response pairs between StarOS proclets.

Data Plane traffic consists of unicast IP protocol 254 packets transferred between cards. This traffic is service port ingress or egress that the StarOs internally transfers to the appropriate application instance (ingress) or service port interface (egress) and is not acknowledged (that is, there are no response packets). For example, an ingress packet arriving on an SF3 port that the Session Manager instance services on SF5, traverses the DI-network from SF3 to SF5.

All operational cards (that is, CFs and SFs with an Active or Standby operational state) transmit and receive monitor packets. The monitor traffic is fully meshed wherein all cards transmit monitor packets to all other cards and receive monitor packets from all other cards.

Data Plane packets are generated at a rate of 10 per second. Control Plane monitor packets are generated at a rate of 5 per second. The packet headers for both are marked with default priority.

StarOS collects and aggregates the monitor transmit, receive, and drop data for all card connections. The **show cloud monitor controlplane** and **show cloud monitor dataplane** CLI commands display current 15 second, 5 minute, and 60 minute data. The 5 minute and 60 minute loss percentages are available as variables in the bulkstats mon-di-net schema. The 5 minute and 60 minute loss percentages are also accessible as threshold alarms/traps.

Note that low or non-zero drop percentages are normal. Because measurements involve correlation across card pairs that are not perfectly synchronized, a response can arrive in the interval adjacent to the one in which the request was generated. This is reflected as a drop in the request interval.

When seen on a sustained basis, higher drop or loss percentages may indicate DI-network configuration or operational issues, traffic overload, or VM or Host issues. The cloud monitor provides the ability to see and characterize DI-network traffic loss; further investigation is typically required to identify the root cause.

Limitations

The Monitor VPC-DI Network feature has the following limitations.

- Only supported on the VPC-DI platform.
- Not license-controlled.

Configuring the Monitor VPC-DI Network Feature

The following section provides information about the CLI commands available to enable or disable the feature.

Configuring Card Fault Detection

Use the following commands to configure secondary card fault detection criteria. This command is configured in the Global Configuration mode.

```
configure
  high-availability fault-detection card dp-outage seconds
  end
```

NOTES:

- **default**: Restores the default dp-outage value. The default value is 2 seconds.
- Note that the dp-outage deferral is limited. If the consecutive heartbeat bounces are 5 greater than the configured hb-loss parameter, then card failure is declared regardless of the dp-outage configuration.
- The **dp-outage** parameter is restricted to Administrator access on the VPC-DI platform.
- If this CLI is not configured, then the default dp-outage value is 2 seconds.

Configuring Packet Loss Threshold on Control Plane

Use the following commands to measure percentage packet loss over the corresponding time interval on the Control plane. The threshold alarm and SNMP trap are raised for any card to card connection that exceeds the configured loss percentage over the indicated time period. This command is configured in the Global Configuration mode.

```
configure
  [ default ] threshold cp-monitor-5min-loss pct [ clear pct ]
  end
```

```
[ default ] threshold poll cp-monitor-5min-loss interval duration
```

```
configure
  [default] threshold cp-monitor-60min-loss pct [ clear pct ]
  end
```

```
[default] threshold poll cp-monitor-60min-loss interval duration
```

NOTES:

- **default**: Clears the configured thresholds for the Control Plane.
- **clear** *pct* : Clears the configured percentage of packet loss.
- **interval** *duration*: Specifies the amount of time (in seconds) that comprises the polling interval. *duration* must be an integer from 60 through 60000. The default is 300 seconds.
- This command is disabled by default.

Note For supplemental information related to this feature, refer to the *Global Configuration Mode Commands* section of the *Command Line Reference*.

The following alarms/traps are generated when these thresholds are exceeded:

- ThreshControlPlaneMonitor5MinsLoss / ThreshClearControlPlaneMonitor5MinsLoss

- ThreshControlPlaneMonitor60MinsLoss / ThreshControlPlaneMonitor60MinsLoss

See the *SNMP MIB Reference* for more details about these alarms/traps.

Configuring Packet Loss Threshold on Data Plane

Use the following commands to measure percentage packet loss over the corresponding time interval on the Data plane. The threshold alarm and SNMP trap are raised for any card to card connection that exceeds the configured loss percentage over the indicated time period. This command is configured in the Global Configuration mode.

```
configure
   [ default ] threshold dp-monitor-5min-loss pct [ clear pct ]
   end
```

```
[ default ] threshold poll dp-monitor-5min-loss interval duration
```

```
configure
   [default] threshold dp-monitor-60min-loss pct [ clear pct ]
   end
```

```
[ default ] threshold poll dp-monitor-60min-loss interval duration
```

NOTES:

- **default**: Disables the configured thresholds for the Data Plane.
- **clear** *pct* : Clears the configured packet loss.
- **interval** *duration*: Specifies the amount of time (in seconds) that comprises the polling interval. *duration* must be an integer from 60 through 60000. The default is 300 seconds.
- This command is disabled by default.

Note For supplemental information related to this feature, refer to the *Global Configuration Mode Commands* section of the *Command Line Reference*.

The following alarms/traps are generated when these thresholds are exceeded:

- ThreshDataPlaneMonitor5MinsLoss / ThreshClearDataPlaneMonitor5MinsLoss
- ThreshDataPlaneMonitor60MinsLoss / ThreshDataPlaneMonitor60MinsLoss

See the *SNMP MIB Reference* for more details about these alarms/traps.

Monitoring and Troubleshooting

This section provides information regarding CLI commands available in support of monitoring and troubleshooting the feature.

Show Command(s) and/or Outputs

This section provides information regarding show commands and/or their outputs in support of this feature.

show cloud monitor controlplane

This new show command is introduced to display the following output for the most recent Control Plane monitor information.

show cloud monitor controlplane

```
  Cards       15 Second Interval      5 Minute Interval     60 Minute Interval
 Src  Dst    Xmit   Recv  Miss%    Xmit   Recv  Miss%    Xmit   Recv  Miss%
 ---  ---  ------ ------ ------  ------ ------ ------  ------ ------ ------
  01   02      75     75   0.0%    1500   1500   0.0%   18000  18000   0.0%
  01   03      75     75   0.0%    1500   1500   0.0%   18000  17996   0.0%
  01   04      75     75   0.0%    1500   1500   0.0%   18000  18000   0.0%
  01   05      75     75   0.0%    1500   1500   0.0%   18000  18000   0.0%
  01   06      75     75   0.0%    1500   1500   0.0%   18000  18000   0.0%

  02   01      75     75   0.0%    1500   1500   0.0%   18000  18000   0.0%
  02   03      75     75   0.0%    1500   1500   0.0%   18000  17997   0.0%
  02   04      75     75   0.0%    1500   1500   0.0%   18000  18000   0.0%
  02   05      75     75   0.0%    1500   1500   0.0%   18000  18000   0.0%
  02   06      75     75   0.0%    1500   1500   0.0%   18000  18000   0.0%

  03   01      75     75   0.0%    1500   1500   0.0%      -incomplete-
  03   02      75     75   0.0%    1500   1500   0.0%      -incomplete-
  03   04      75     75   0.0%    1500   1500   0.0%      -incomplete-
  03   05      75     75   0.0%    1500   1500   0.0%      -incomplete-
  03   06      75     75   0.0%    1500   1500   0.0%      -incomplete-

  04   01      75     75   0.0%    1500   1500   0.0%   18000  18000   0.0%
  04   02      75     75   0.0%    1500   1500   0.0%   18000  18000   0.0%
  04   03      75     75   0.0%    1500   1500   0.0%   18000  17996   0.0%
  04   05      75     75   0.0%    1500   1500   0.0%   18000  18000   0.0%
  04   06      75     75   0.0%    1500   1500   0.0%   18000  18000   0.0%

  05   01      75     75   0.0%    1500   1500   0.0%   18000  18000   0.0%
  05   02      75     75   0.0%    1500   1500   0.0%   18000  18000   0.0%
  05   03      75     75   0.0%    1500   1500   0.0%   18000  17996   0.0%
  05   04      75     75   0.0%    1500   1500   0.0%   18000  18000   0.0%
  05   06      75     75   0.0%    1500   1500   0.0%   18000  18000   0.0%

  06   01      75     75   0.0%    1500   1500   0.0%   18000  18000   0.0%
  06   02      75     75   0.0%    1500   1500   0.0%   18000  18000   0.0%
  06   03      75     75   0.0%    1500   1500   0.0%   18000  17997   0.0%
  06   04      75     75   0.0%    1500   1500   0.0%   18000  18000   0.0%
  06   05      75     75   0.0%    1500   1500   0.0%   18000  18000   0.0%
```

show cloud monitor dataplane

This new show command is introduced to display the following output for the most recent Data Plane monitor information.

show cloud monitor dataplane

```
  Cards       15 Second Interval      5 Minute Interval     60 Minute Interval
 Src  Dst    Miss    Hit    Pct    Miss    Hit    Pct    Miss    Hit    Pct
 ---  ---  ------ ------ ------  ------ ------ ------  ------ ------ ------
  02   01       0    150   0.0%       0   3000   0.0%       0  36000   0.0%
  03   01       0    150   0.0%       0   3000   0.0%      -incomplete-
  04   01       0    151   0.0%       0   3000   0.0%       0  36000   0.0%
  05   01       0    151   0.0%       0   3000   0.0%       0  36001   0.0%
  06   01       0    150   0.0%       0   3000   0.0%       2  35998   0.0%

  01   02       0    150   0.0%       0   3000   0.0%       0  36000   0.0%
  03   02       0    150   0.0%       0   3000   0.0%      -incomplete-
  04   02       0    150   0.0%       0   3000   0.0%       0  36000   0.0%
  05   02       0    150   0.0%       0   3000   0.0%       0  36000   0.0%
```

```
06   02       0    151   0.0%       0   3001   0.0%       1  35999   0.0%

01   03       0    151   0.0%       0   3000   0.0%      -incomplete-
02   03       0    151   0.0%       0   3000   0.0%      -incomplete-
04   03       0    150   0.0%       0   3000   0.0%      -incomplete-
05   03       0    150   0.0%       0   3000   0.0%      -incomplete-
06   03       0    151   0.0%       0   3000   0.0%      -incomplete-

01   04       0    150   0.0%       0   3001   0.0%       0  36001   0.0%
02   04       0    150   0.0%       0   3000   0.0%       0  36000   0.0%
03   04       0    150   0.0%       0   3000   0.0%      -incomplete-
05   04       1    149   0.7%       1   2999   0.0%       0  36001   0.0%
06   04       0    150   0.0%       0   3000   0.0%       2  35998   0.0%

01   05       1    149   0.7%       1   2999   0.0%       0  36000   0.0%
02   05       0    150   0.0%       0   3000   0.0%       0  36000   0.0%
03   05       0    150   0.0%       0   3000   0.0%      -incomplete-
04   05       0    150   0.0%       1   2999   0.0%       1  35999   0.0%
06   05       0    150   0.0%       0   3000   0.0%       2  35998   0.0%

01   06       0    150   0.0%       0   3001   0.0%       0  36001   0.0%
02   06       0    151   0.0%       0   3000   0.0%       1  35999   0.0%
03   06       0    150   0.0%       0   3001   0.0%      -incomplete-
04   06       0    150   0.0%       0   3000   0.0%       0  36000   0.0%
05   06       0    150   0.0%       0   3000   0.0%       0  36000   0.0%
```

Bulk Statistics

The following statistics are included in support of this feature.

mon-di-net Schema

The following bulk statistics are added in the mon-di-net schema in support of the Monitor the VPC-DI Network feature.

Bulk Statistics	Description
cp-loss-5minave	Indicates the average Control Plane loss in prior 5 minutes.
cp-loss-60minave	Indicates the average Control Plane loss in prior 60 minutes.
dp-loss-5minave	Indicates the average Data Plane loss in prior 5 minutes.
dp-loss-60minave	Indicates the average Data Plane loss in prior 60 minutes.

Configure DI-Network Heartbeat Thresholds

The following steps describe how to configure threshold levels to generate SNMP alarms if the percentage of heartbeats lost exceeds the configured level.

Note The internal High Availability Task (HAT) is alway monitoring the heartbeats across the VMs on the internal DI-Network. This information can be displayed at any time using the **show cloud monitor di-network summary** Exec mode command.

```
configure
  monitoring hat-5min-loss
  threshold hat-hb-5min-loss high_thresh [ clear low_thresh ]
default threshold hat-hb-5min-loss
  [ default ] threshold poll hat-hb-5min-loss interval duration
configure
  monitoring hat-60min-loss
  threshold hat-hb-60min-loss high_thresh [ clear low_thresh ]
default threshold hat-hb-60min-loss
  [ default ] threshold poll hat-hb-5min-loss interval duration
```

Note For supplemental information related to this feature, refer to the *Global Configuration Mode Commands* section of the *Command Line Reference*.

The following alarms/traps are generated when these thresholds are exceeded:

- ThreshHatHb5MinLoss / ThreshClearHatHb5MinLoss.
- ThreshHatHb60MinLoss / ThreshClearHatHb60MinLoss.

See the *SNMP MIB Reference* for more details about these alarms/traps.

Configuration Support for Heartbeat Value

Feature Summary and Revision History

Summary Data

Applicable Product(s) or Functional Area	All
Applicable Platform(s)	• ASR 5500 • VPC - DI
Feature Default	Disabled - Configuration Required
Related Changes in This Release	Not Applicable
Related Documentation	• *ASR 5500 System Administration Guide* • *Command Line Interface Reference*

	• *VPC-DI System Administration Guide* • *Statistics and Counters Reference*

Revision History

Revision Details	Release
In this release, the default heartbeat value between the management and data processing card can be modified to prevent the management card from incorrectly detecting and reporting the packet processing card as failed.	21.8
First introduced.	Pre 21.2

Feature Changes

In certain deployment scenarios, the management card reports the packet processing card as failed when it is unable to detect a heartbeat for about two seconds. This assumed failure is observed when the hearbeat is delayed or lost due to congestion in the internal DI network.

This release addresses this issue.

Previous Behavior: The management card reports the packet processing card as failed due to its inability to detect the heartbeat within the default value of two seconds, thereby causing an unplanned switchover.

New Behavior: To prevent the management card from incorrectly detecting and reporting the packet processing card as failed, the default heartbeat value between the management and data processing card can now be modified.

Customer Impact: Prevents the management card from wrongful reporting of the data processing card and unplanned switchover.

Command Changes

high-availability fault-detection

The above CLI command is enhanced to include the **card hb-loss** *value* keyword, which is used to configure the heartbeat value between the management and packet processing cards. This command is configured in the Global Configuration Mode.

```
configure
  [default] high-availability fault-detection card hb-loss value
  end
```

NOTES:

- **default**: Restores the heartbeat value to the default value of 2 heartbeats.
- **card**: Specifies the packet processing card.
- **hb-loss** *value*: Configures the heartbeat loss value. The default value is 2 heartbeats.
- The heartbeat value between a management to management card is set to the default value of 2 heartbeats.

- This command modifies the heartbeat value only between the management and packet processing cards.
- By default, this CLI is disabled.

Monitoring and Troubleshooting

This section provides information regarding show commands and/or their outputs in support of this feature.

show heartbeat stats hb-loss all

This show command now includes the value for the following new fields for the all packet processing cards.

- Max Bounces
- Total HB Miss
- Total HB Card Failure
  - Card/Cpu
  - Total
  - Age/Intf/Seqno/TimeStamp
  - AFD(oldest first)

show heartbeat stats hb-loss card *card-number*

This show command now includes the value for the following new fields for the specified packet processing card.

- Max Bounces
- Total HB Miss
- Total HB Card Failure
  - Card/Cpu
  - Total
  - Age/Intf/Seqno/TimeStamp
  - AFD(oldest first)

Monitoring the SF

To view NPU statistics for each active and standby SF, use the **show npu utilization table** command. Statistics are reported for the past five seconds, past five minutes and past 15 minutes. Sample output is shown here:

```
[local]swch91# show npu utilization table

 ******** show npu utilization table card 4 *******
             5-Sec Avg:   lcore00|lcore01|lcore02|lcore03|lcore04|lcore05|lcore06|lcore07|
```

```
                  IDLE:          |    99%|       |       |       |       |       |
|
         QUEUE_PORT_RX:          |     0%|       |       |       |       |       |
|
         QUEUE_PORT_TX:          |       |       |       |       |       |       |
|
         QUEUE_VNPU_RX:          |       |       |       |       |       |       |
|
         QUEUE_VNPU_TX:          |       |       |       |       |       |       |
|
          QUEUE_KNI_RX:          |       |       |       |       |       |       |
|
          QUEUE_KNI_TX:          |       |       |       |       |       |       |
|
      QUEUE_THREAD_KNI:          |       |       |       |       |       |       |
|
        QUEUE_MCDMA_RX:          |       |       |       |       |       |       |
|
        QUEUE_MCDMA_TX:          |       |       |       |       |       |       |
|
    QUEUE_THREAD_MCDMA:          |       |       |       |       |       |       |
|
     QUEUE_THREAD_VNPU:          |       |       |       |       |       |       |
|
       QUEUE_CRYPTO_RX:          |       |       |       |       |       |       |
|
      QUEUE_CRYPTO_IPC:          |       |       |       |       |       |       |
|
      QUEUE_THREAD_IPC:          |       |       |       |       |       |       |
|
           MCDMA_FLUSH:          |       |       |       |       |       |       |
|
  QUEUE_THREAD_TYPE_MAX:         |       |       |       |       |       |       |
|
         300-Sec Avg:   lcore00|lcore01|lcore02|lcore03|lcore04|lcore05|lcore06|lcore07|
                  IDLE:          |    99%|       |       |       |       |       |
|
         QUEUE_PORT_RX:          |     0%|       |       |       |       |       |
|
         QUEUE_PORT_TX:          |       |       |       |       |       |       |
|
         QUEUE_VNPU_RX:          |       |       |       |       |       |       |
|
         QUEUE_VNPU_TX:          |       |       |       |       |       |       |
|
          QUEUE_KNI_RX:          |       |       |       |       |       |       |
|
          QUEUE_KNI_TX:          |       |       |       |       |       |       |
|
      QUEUE_THREAD_KNI:          |       |       |       |       |       |       |
|
        QUEUE_MCDMA_RX:          |       |       |       |       |       |       |
|
        QUEUE_MCDMA_TX:          |       |       |       |       |       |       |
|
    QUEUE_THREAD_MCDMA:          |       |       |       |       |       |       |
|
     QUEUE_THREAD_VNPU:          |       |       |       |       |       |       |
|
       QUEUE_CRYPTO_RX:          |       |       |       |       |       |       |
|
      QUEUE_CRYPTO_IPC:          |       |       |       |       |       |       |
|
```

```
        QUEUE_THREAD_IPC:          |       |       |       |       |       |       |
|
            MCDMA_FLUSH:           |       |       |       |       |       |       |
|
  QUEUE_THREAD_TYPE_MAX:           |       |       |       |       |       |       |
|
          900-Sec Avg:   lcore00|lcore01|lcore02|lcore03|lcore04|lcore05|lcore06|lcore07|

                  IDLE:            |    99%|       |       |       |       |       |
|
          QUEUE_PORT_RX:           |     0%|       |       |       |       |       |
|
          QUEUE_PORT_TX:           |       |       |       |       |       |       |
|
          QUEUE_VNPU_RX:           |       |       |       |       |       |       |
|
          QUEUE_VNPU_TX:           |       |       |       |       |       |       |
|
           QUEUE_KNI_RX:           |       |       |       |       |       |       |
|
           QUEUE_KNI_TX:           |       |       |       |       |       |       |
|
        QUEUE_THREAD_KNI:          |       |       |       |       |       |       |
|
         QUEUE_MCDMA_RX:           |       |       |       |       |       |       |
|
         QUEUE_MCDMA_TX:           |       |       |       |       |       |       |
|
      QUEUE_THREAD_MCDMA:          |       |       |       |       |       |       |
|
       QUEUE_THREAD_VNPU:          |       |       |       |       |       |       |
|
        QUEUE_CRYPTO_RX:           |       |       |       |       |       |       |
|
        QUEUE_CRYPTO_IPC:          |       |       |       |       |       |       |
|
        QUEUE_THREAD_IPC:          |       |       |       |       |       |       |
|
            MCDMA_FLUSH:           |       |       |       |       |       |       |
|
  QUEUE_THREAD_TYPE_MAX:           |       |       |       |       |       |       |
|


thread 1 IDLE                           99.32 %
thread 1 QUEUE_KNI_RX                    0.63 %
thread 1 QUEUE_PORT_RX                   0.05 %
-----------------------------------------------

 ******** show npu utilization table card 5 *******
             5-Sec Avg:   lcore00|lcore01|lcore02|lcore03|lcore04|lcore05|lcore06|lcore07|

                  IDLE:            |    99%|       |       |       |       |       |
|
          QUEUE_PORT_RX:           |       |       |       |       |       |       |
|
          QUEUE_PORT_TX:           |       |       |       |       |       |       |
|
          QUEUE_VNPU_RX:           |       |       |       |       |       |       |
|
          QUEUE_VNPU_TX:           |       |       |       |       |       |       |
|
           QUEUE_KNI_RX:           |     0%|       |       |       |       |       |
```

```
|
          QUEUE_KNI_TX:          |       |       |       |       |       |       |
|
      QUEUE_THREAD_KNI:          |       |       |       |       |       |       |
|
        QUEUE_MCDMA_RX:          |       |       |       |       |       |       |
|
        QUEUE_MCDMA_TX:          |       |       |       |       |       |       |
|
    QUEUE_THREAD_MCDMA:          |       |       |       |       |       |       |
|
     QUEUE_THREAD_VNPU:          |       |       |       |       |       |       |
|
       QUEUE_CRYPTO_RX:          |       |       |       |       |       |       |
|
      QUEUE_CRYPTO_IPC:          |       |       |       |       |       |       |
|
      QUEUE_THREAD_IPC:          |       |       |       |       |       |       |
|
           MCDMA_FLUSH:          |       |       |       |       |       |       |
|
  QUEUE_THREAD_TYPE_MAX:         |       |       |       |       |       |       |
|
         300-Sec Avg:   lcore00|lcore01|lcore02|lcore03|lcore04|lcore05|lcore06|lcore07|

                  IDLE:          |    99%|       |       |       |       |       |
|
         QUEUE_PORT_RX:          |       |       |       |       |       |       |
|
         QUEUE_PORT_TX:          |       |       |       |       |       |       |
|
         QUEUE_VNPU_RX:          |       |       |       |       |       |       |
|
         QUEUE_VNPU_TX:          |       |       |       |       |       |       |
|
          QUEUE_KNI_RX:          |     0%|       |       |       |       |       |
|
          QUEUE_KNI_TX:          |       |       |       |       |       |       |
|
      QUEUE_THREAD_KNI:          |       |       |       |       |       |       |
|
        QUEUE_MCDMA_RX:          |       |       |       |       |       |       |
|
        QUEUE_MCDMA_TX:          |       |       |       |       |       |       |
|
    QUEUE_THREAD_MCDMA:          |       |       |       |       |       |       |
|
     QUEUE_THREAD_VNPU:          |       |       |       |       |       |       |
|
       QUEUE_CRYPTO_RX:          |       |       |       |       |       |       |
|
      QUEUE_CRYPTO_IPC:          |       |       |       |       |       |       |
|
      QUEUE_THREAD_IPC:          |       |       |       |       |       |       |
|
           MCDMA_FLUSH:          |       |       |       |       |       |       |
|
  QUEUE_THREAD_TYPE_MAX:         |       |       |       |       |       |       |
|
         900-Sec Avg:   lcore00|lcore01|lcore02|lcore03|lcore04|lcore05|lcore06|lcore07|

                  IDLE:          |    99%|       |       |       |       |       |
|
         QUEUE_PORT_RX:          |       |       |       |       |       |       |
```

```
|
          QUEUE_PORT_TX:          |       |       |       |       |       |       |
|
          QUEUE_VNPU_RX:          |       |       |       |       |       |       |
|
          QUEUE_VNPU_TX:          |       |       |       |       |       |       |
|
           QUEUE_KNI_RX:          |     0%|       |       |       |       |       |
|
           QUEUE_KNI_TX:          |       |       |       |       |       |       |
|
       QUEUE_THREAD_KNI:          |       |       |       |       |       |       |
|
         QUEUE_MCDMA_RX:          |       |       |       |       |       |       |
|
         QUEUE_MCDMA_TX:          |       |       |       |       |       |       |
|
     QUEUE_THREAD_MCDMA:          |       |       |       |       |       |       |
|
      QUEUE_THREAD_VNPU:          |       |       |       |       |       |       |
|
        QUEUE_CRYPTO_RX:          |       |       |       |       |       |       |
|
       QUEUE_CRYPTO_IPC:          |       |       |       |       |       |       |
|
       QUEUE_THREAD_IPC:          |       |       |       |       |       |       |
|
            MCDMA_FLUSH:          |       |       |       |       |       |       |
|
  QUEUE_THREAD_TYPE_MAX:          |       |       |       |       |       |       |
|


thread 1 IDLE                           99.37 %
thread 1 QUEUE_KNI_RX                    0.55 %
thread 1 QUEUE_PORT_RX                   0.08 %
-----------------------------------------------
```

Table 11: show npu utilization table

Field	Description
IDLE	Idle time in each core
QUEUE_PORT_RX	Time spent processing RX port
QUEUE_PORT_TX	Time spent processing TX port
QUEUE_VNPU_RX	Time spent processing RX vNPU
QUEUE_VNPU_TX	Time spent processing TX vNPU
QUEUE_KNI_RX	Time spent processing RX kernal network interface (KNI). The KNI is the path to the kernal from the IFTASK.
QUEUE_KNI_TX	Time spent processing TX KNI
QUEUE_THREAD_KNI	Thread dedicated to KNI processing

Field	Description
QUEUE_MCDMA_RX	Time spent processing RX multi-channel direct memory acess (DMA) [MCDMA]. The MCDMA is the path from the IFTASK to the SESSMGR.
QUEUE_MCDMA_TX	Time spent processing TX MCDMA.
QUEUE_THREAD_MCDMA	Thread dedicated to MCDMA processing
QUEUE_THREAD_VNPU	Thread dedicated to VNPU processing
QUEUE_CRYPTO_RX	Time spent processing IPSec
QUEUE_CRYPTO_IPC	Time spent processing IPSec inter-process communication (IPC)
MCDMA_FLUSH	Time spent flushing out MCDMA packets
QUEUE_THREAD_TYPE_MAX	Not used

Clearing Statistics and Counters

It may be necessary to periodically clear statistics and counters in order to gather new information. The system provides the ability to clear statistics and counters based on their grouping (PPP, MIPHA, MIPFA, etc.).

Statistics and counters can be cleared using the CLI **clear** command. Refer to the *Exec Mode Commands* chapter of the *Command Line Interface Reference* for detailed information on using this command.

CHAPTER 15

Monitor Process Listing

Feature Summary and Revision History

Summary Data

Applicable Product(s) or Functional Area	All
Applicable Platform(s)	• ASR 5500 • VPC-DI • VPC-SI
Feature Default	Enabled - Always-on
Related Changes in This Release	Not applicable
Related Documentation	• *ASR 5500 System Administration Guide* • *Command Line Interface Reference* • *Statistics and Counters Reference* • *VPC-DI System Administration Guide* • *VPC-SI System Administration Guide*

Revision History

Important Revision history details are not provided for features introduced before releases 21.2 and N5.1.

Revision Details	Release
First introduced.	21.11

Feature Description

The Monitor Process Listing feature supports the following functionalities:

- Viewing the running processes to check and detect intrusion.
- Checking the software to detect if it is tamper-proof.
- Enabling security decisions.

The newly introduced CLI command, **show process status**, supports this feature.

Monitoring and Troubleshooting

This section provides information regarding the CLI command available in support of monitoring and troubleshooting the feature.

Show Command(s) and/or Outputs

This section provides information regarding the show command and/or its output in support of this feature.

show process status

The output of this CLI command now includes the following fields in support of this feature:

- card - cpu
  - USER
  - PID
  - PPID
  - STARTED
  - %CPU
  - %MEM
  - COMMAND

Note Only the Security Administrator can run this command.

CHAPTER 16

Bulk Statistics

This chapter provides configuration information for:

Feature Summary and Revision History

Summary Data

Applicable Product(s) or Functional Area	All
Applicable Platform(s)	ASR 5500 VPC-DI VPC-SI
Feature Default	Disabled - Configuration Required
Related Changes in This Release	Not Applicable
Related Documentation	• *ASR 5500 System Administration Guide* • *Command Line Interface Reference* • *VPC-DI System Administration Guide* • *VPC-SI System Administration Guide*

Revision History

Note Revision history details are not provided for features introduced before releases 21.2 and N5.5.

Revision Details	Release
New functionality was added to replace or supplement the configured bulkstats schema with the option of preserving bulkstats configuration parameters. New functionality was added to collect bulkstats samples in the SSD. Refer to the *Collecting Bulk Statistics Samples in SSD* section for more information. The **bulkstat** Global Configuration Mode command added the **config [schema | supplement]** keywords to enable this functionality. Refer to the *Configuring a Separate Bulkstats Config File* section for more information. **show configuration bulkstats brief** command output was expanded to include all bulkstats configuration details except for schema.	21.3
First introduced.	Pre 21.2

Configuring Communication with the Collection Server

Two configuration methods are available for defining how bulk statistics are collected and managed. A "standard" configuration allows the system to automatically assign a number to the bulk statistics file. Optionally, a number can be specified by an administrator in the optional configuration method. Command details and descriptions of keywords and variables for commands in this chapter are located in the *Bulk Statistics Configuration Mode Commands* and *Bulk Statistics File Configuration Mode Commands* chapters in the *Command Line Interface Reference*.

Configuring Standard Settings

The configuration example in this section defines basic operation of the bulk statistics feature. Use the following example configuration to set up the system to communicate with the statistic collection server:

```
configure
  bulkstats mode
     schema name format format_string
     sample-interval time_interval
     transfer-interval xmit_time_interval
     limit mem_limit
     exit
   bulkstats collection
   end
```

Configuring Optional Settings

This section describes optional commands that can be used within the Bulk Statistics Configuration mode. Specifically, you can configure bulk statistic "files" under which to group the bulk statistics. "Files" are used to group bulk statistic schema, delivery options, and receiver configuration. Because multiple "files" can be configured, this functionality provides greater flexibility because it allows you to configure different schemas to go to different receivers.

```
configure
  bulkstats mode
    file number
      receiver ip_address { primary | secondary }
       [ mechanism { { { ftp | sftp } login user_name
       [ encrypted ] password pwd } | tftp } } ] }
      receiver mode { redundant | secondary-on-failure }
      remotefile format naming_convention [ both-receivers | primary-receiver
 | secondary-receiver ]
      header format header_format
      footer format footer_format
      exit
    schema_type schema format format_string
    sample-interval time_interval
    transfer-interval xmit_time_interval
    limit mem_limit
    exit
  bulkstats collection
  end
```

Important In release 20.0 and higher Trusted StarOS builds, FTP is not supported. SFTP is the recommended transfer protocol.

Configuring Bulk Statistic Schemas

In each configuration example described in Configuring Standard Settings, on page 232 and Configuring Optional Settings, on page 233, the following is the primary command used to configure the type of schema and the statistics collected:

```
configure
  bulkstats mode
    schema_type schema format format_string
```

Refer to the *Bulk Statistics Configuration Mode Commands* and *Bulk Statistics File Configuration Mode Commands* chapters in the *Command Line Interface Reference* for more information regarding supported schemas, available statistics, and proper command syntax.

Configuring a Separate Bulkstats Config File

You can configure a separate destination file for storing the bulk statistics sub-mode configuration. Run the **show configuration bulkstats** command to confirm the configuration.

The bulkstats configuration file stores the configuration that was previously stored in the system configuration file under the bulk statistics sub-mode.

The Global Configuration mode **bulkstats config** command creates the separate configuration file in the system configuration.

```
configure
  [no] bulkstats config [ schema | supplement ] url
  end
```

The optional **schema** keyword allows you to replace only the schema using the file provided and preserve the server configuration. The optional **supplement** keyword allows you to supplement the running bulkstats configuration with the contents of the configuration file provided. These keywords only work on existing files.

url specifies the location of the bulkstats configuration file. If the destination file already exists, it is replaced with the new file (except when the **schema** or **supplement** keywords are used). The new file will only be created if you save the configuration after completing changes. The Exec mode **show configuration bulkstats** command displays the URL for the bulkstats configuration mode destination file if it has been configured.

You can copy the bulkstats configuration file from the *url*, edit it and copy it back to /flash. Changes can be applied by using the **no** form of the **bulkstats config** command followed by reconfiguring the **bulkstats config** command.

When the **bulkstats config** command is enabled, StarOS removes the existing bulk statistics sub-mode configuration from the system configuration file. You must save the system configuration to retain the configuration change.

If **no bulkstats config** is used to disable the new destination file after it has been enabled. StarOS does not remove the file. You must save the system configuration to retain the configuration change.

Important After completing changes to the bulk statistics configuration, you must save the system configuration to save the changes. If the **bulkstats config** command is enabled, the bulkstats configuration file will be updated.

Using show bulkstats Commands

There are several Exec mode **show bulkstats** commands that display information about defined parameters.

- **show bulkstats data** – displays criteria contained in the statistics gathering scheme for up to four files. See Viewing Collected Bulk Statistics Data, on page 236.
- **show bulkstats schemas** – displays the scheme used to gather statistics including collection and transmission statistics. See Verifying Your Configuration, on page 235.
- **show bulkstats variables** – displays available bulkstat variables (%*variable*%) by schema type that can be incorporated into a schema format.

In addition, **show configuration bulkstats brief** displays the bulkstats configuration at a global scope, as well as all server configuration. It does not display the schema configuration.

Verifying Your Configuration

After configuring support for bulk statistics on the system, you can check your settings prior to saving them.

Follow the instructions in this section to verify your bulk statistic settings. These instructions assume that you are at the root prompt for the Exec mode.

Check your collection server communication and schema settings by entering the following Exec mode command:

show bulkstats schemas

The following is an example command output:

```
Bulk Statistics Server Configuration:
    Server State:                           Enabled
    File Limit:                             6000 KB
    Sample Interval:                        15 minutes (0D 0H 15M)
    Transfer Interval:                      480 minutes (0D 0H 15M)
    Collection Mode:                        Cumulative
    Receiver Mode:                          Secondary-on-failure
    Local File Storage:                     None
Bulk Statistics Server Statistics:
    Records awaiting transmission:  114
    Bytes awaiting transmission:    8092
    Total records collected:        59926
    Total bytes collected:          4190178
    Total records transmitted:      59812
    Total bytes transmitted:        4188512
    Total records discarded:        0
    Total bytes discarded:          0
    Last collection time required:  2 second(s)
    Last transfer time required:    0 second(s)
    Last successful transfer:       Wednesday December 7 12:14:30 EDT 2011
    Last successful tx recs:        190
    Last successful tx bytes:       13507
    Last attempted transfer:        Wednesday December 7 12:14:30 EDT 2011
File 1
    Remote File Format:     /users/ems/server/data/chicago/bulkstat%date%%time%.txt
    File Header:            "CHI_test %time%"
    File Footer:            ""
Bulkstats Receivers:
    Primary: 192.168.0.100 using FTP with username administrator
    Records awaiting transmission:    0
    Bytes awaiting transmission:      0
    Total records collected:          0
    Total bytes collected:            0
    Total records transmitted:        0
    Total bytes transmitted:          0
    Total records discarded:          0
    Total bytes discarded:            0
    Last transfer time required:      0 second(s)
    No successful data transfers
    No attempted data transfe

File 2 not configured

File 3 not configured

File 4 not configured
```

Saving Your Configuration

Save the configuration as described in the *Verifying and Saving Your Configuration* chapter.

Viewing Collected Bulk Statistics Data

The system provides a mechanism for viewing data that has been collected but has not been transferred. This data is referred to as "pending data".

View pending bulk statistics data per schema by entering the following Exec mode command:

```
show bulkstats data
```

The above command also shows the statistics of remote files, if configured as described in Configuring Optional Settings, on page 233.

Collecting Bulk Statistics Samples in SSD

The output of the show support details (SSD) command is collected and provided to the Technical Assistance Center (TAC) for troubleshooting purposes. Bulkstats information in the SSD enables customers to provide readily available bulk statistics records for analysis, and faster resolution to issues.

By default, the system does not include bulkstats samples in the SSD. This functionality can be enabled using the **bulkstats ssd-samples** command under the Global Configuration Mode.

```
config
    bulkstats ssd-samples { 1 | 2 }
    end
```

If previously enabled, the **{ no | default } bulkstats ssd-samples** command disables collection of bulkstats samples in the SSD archive. Each bulkstats sample contains bulkstats from one transfer history. Currently, a maximum of two bulkstats sample can be included in the SSD archive. The sample files are collected in a temporary storage location at */var/tmp/bulkstats* under the file name *ssd_bulkstats_file<bulkstat_file_number>_sample1.txt* (and *ssd_bulkstats_file<bulkstat_file_number>_sample2.txt*).

While the SSD archive is being created in the temporary storage, the bulk statistics samples might occupy a large amount of the storage space. As a result, the SSD archive creation might fail. During such scenarios, the **no-bulkstats** keyword in the **show support details** command can exculde the bulkstats samples from the SSD archive.

Manually Gathering and Transferring Bulk Statistics

There may be times where it is necessary to gather and transfer bulk statistics outside of the scheduled intervals. The system provides commands that allow you to manually initiate the gathering and transferring of bulk statistics.

To manually initiate the gathering of bulk statistics outside of the configured sampling interval, enter the following Exec mode command:

```
bulkstats force gather
```

To manually initiate the transferring of bulk statistics prior to reaching the of the maximum configured storage limit, enter the following Exec mode command:

```
bulkstats force transfer
```

Clearing Bulk Statistics Counters and Information

It may be necessary to periodically clear counters pertaining to bulk statistics in order to gather new information or to remove bulk statistics information that has already been collected. The following Exec mode command can be used to perform either of these functions:

```
clear bulkstats { counters | data }
```

The **clear bulkstats data** command clears any accumulated data that has not been transferred. This includes any "completed" files that have not been successfully transferred.

Bulkstats Schema Nomenclature

This section describes the nomenclature associated with configuring and viewing bulkstats.

Statistic Types

The following statistic types are defined in the *Statistics and Counters Reference* user document (published prior to Release 20.0) and displayed in the output of the Exec mode **show bulkstats variables** command"

- **Counter:** A counter records incremental data cumulatively and rolls over when the counter limit is reached.
  - All counter statistics are cumulative and reset only by one of the following methods: roll-over when the limit is reached, after a system restart, or after a clear command is performed.
  - The limit depends upon the data type.
- **Gauge:** A gauge statistic indicates a single value; a snapshot representation of a single point in time within a defined time frame. The gauge changes to a new value with each snapshot though a value may repeat from one period to the next. The limit depends upon the data type.
- **Information:** This type of statistic provides information, often intended to differentiate sets of statistics; for example, a VPN name or IP address. The type of information provided depends upon the data type.

The following statistic types are included in the *Statistics and Counters Reference* spreadsheet published after release 20.0 to replace the original user document:

- **Incremental:** An incremental data type records incremental data cumulatively and rolls over when the counter limit is reached.
  - All incremental statistics are cumulative and reset only by one of the following methods: roll-over when the limit is reached, after a system restart, or after a clear command is performed.
  - The limit depends upon the data type.

- **Gauge:** A gauge statistic indicates a single value; a snapshot representation of a single point in time within a defined time frame. The gauge changes to a new value with each snapshot though a value may repeat from one period to the next. The limit depends upon the data type.
- **Primary-key:** This type of statistic provides information, often intended to differentiate sets of statistics; for example, a VPN name or IP address. The type of information provided depends upon the data type.

Data Types

The data type defines the format of the data for the value provided by the statistic. The following data types appear in the *Statistics and Counters Reference* and the output of the Exec mode **show bulkstats variables** command:

- **Int32:** A 32-bit integer; the roll-over to zero limit is 4,294,967,295.
- **Int64:** A 64-bit integer; the roll-over to zero limit is 18,446,744,073,709,551,615.
- **Float:** A numeric value that includes decimal points; for example, 1.345.
- **String:** A series of ASCII alphanumeric characters in a single grouping, usually pre-configured.

Key Variables

Every schema has some variables which are typically referred to as "key variables". These key variables provide index markers to identify to which object the statistics apply. For example, in the card schema the card number (variable %card%) uniquely identifies a card. For an HA service, the keys would be "%vpnname%" plus "%servname%', as the combination uniquely identifies an HA service. So, in a given measurement interval, one row of statistics will be generated per unique key.

There are also a number of common variables shared across schema that identify time, date, place, etc. These common variables are identified in the table below.

Table 12: Common Variables Across Schema

Variables	Description	Statistic Type	Data Type
version-no	Contains complete version information that can be used in the header of the bulkstats file	Information	String
uptime	The total uptime (in seconds) of the system that created the file.	Information	Int32
host	The system hostname that created the file.	Information	String
ipaddr	The default management (local context) IP address in dPv4 dotted-decimal format. An empty string is inserted if no address is available.	Information	String

Variables	Description	Statistic Type	Data Type
date	The UTC date that the collection file was created in YYYYMMDD format where YYYY represents the year, MM represents the month and DD represents the day.	Information	String
date3	The UTC date that the collection file was created in YYMMDD format where YY represents the year, MM represents the month and DD represents the day.	Information	String
time	The UTC time that the collection file was created in HHMMSS format where HH represents the hours, MM represents the minutes, and SS represents the seconds.	Information	String
time2	The UTC time that the collection file was created in HH:MM:SS format where HH represents the hours, MM represents the minutes, and SS represents the seconds.	Information	String
time3	The UTC time that the collection file was created in HH:MM format where HH represents the hours, MM represents the minutes.	Information	String
epochtime	The number of seconds since Jan 1, 1970, 00:00:00 GMT.	Information	In32
schemas	Lists all bulkstat schemas available on this platform.	Information	String
schemas-delta	Lists all bulkstats schemas that have changed the schema list was last output.	Information	String
localdate	The date (adjusted for the local timezone) that the collection file was created in YYYYMMDD format where YYYY represents the year, MM represents the month and DD represents the day.	Information	String

Variables	Description	Statistic Type	Data Type
localdate3	The date that the collection file was created in YYYYMMDD format where YYYY represents the year, MM represents the month and DD represents the day. The date displays in local time, not UTC.	Information	String
localtime	The time (adjusted for the local timezone) that the collection file was created in HHMMSS format where HH represents the hours, MM represents the minutes, and SS represents the seconds.	Information	String
localtime2	The time (adjusted for the local timezone) that the collection file was created in HH:MM:SS format where HH represents the hours, MM represents the minutes, and SS represents the seconds.	Information	String
localtime3	The time that the collection file was created in HH:MM:SS format where HH represents the hours, MM represents the minutes, and SS represents the seconds. The time displays in local time, not UTC.	Information	String
localtz	The local timezone set for this platform.	Information	String
localtzoffset	The offset from UTC/GMT for the local timezone. Format = "+" or "-" HHMM.	Information	String
swbuild	The build number of the StarOS version.	Information	String

Bulk Statistics Event Log Messages

The stat logging facility captures several events that can be useful for diagnosing errors that could occur with either the creation or writing of a bulk statistic data set to a particular location.

The following table displays information pertaining to these events.

Table 13: Logging Events Pertaining to Bulk Statistics

Event	Event ID	Severity	Additional Information
Local File Open Error	31002	Warning	"Unable to open local file *filename* for storing bulkstats data"
Receiver Open Error	31018	Warning	"Unable to open url *filename* for storing bulkstats data"
Receiver Write Error	31019	Warning	"Unable to write to url *filename* while storing bulkstats data"
Receiver Close Error	31020	Warning	"Unable to close url *filename* while storing bulkstats data"

CHAPTER 17

System Logs

This chapter describes how to configure parameters related to the various types of logging and how to viewing their content. It includes the following sections:

Feature Summary and Revision History

Summary Data

Applicable Product(s) or Functional Area	All
Applicable Platform(s)	• ASR 5500 • UGP • VPC-DI • VPC-SI
Feature Default	Enabled
Related Changes in This Release:	Not Applicable

Related Documentation	• *ASR 5500 System Administration Guide* • *Command Line Interface Reference* • *VPC-DI System Administration Guide* • *VPC-SI System Administration Guide*

Revision History

Note Revision history details are not provided for features introduced before releases 21.2 and N5.5.

Revision Details	Release
The Syslog client within StarOS has been updated in this release to support RFC5424 and the syslog messaging standards defined within this standard. StarOS continues to support the previous RFC3164 message formats. In this release, you can also configure multiple syslog server IP addresses with multiple ports. **Note** Release 21.6 supports transport layer messaging with UDP only. TLS and TCP are not supported in this release.	21.6
Two new critical CLI event logs and two new SNMP Traps are added to provide notification if an administrator disables logging entirely for an Event ID or Event ID range, or changes the logging level below default logging level (error level). These event logs and traps are enabled by default in this release, and cannot be disabled. Refer to Global Configuration Mode Filtering, on page 248 for more information. No commands have been added or modified as a result of this feature. The **show snmp trap statistics** command output was expanded to show details in the event that logging events have been disabled or logging level has been changed below the default (error) logging level.	21.3
First introduced.	Pre 21.2

System Log Types

There are five types of logs that can be configured and viewed on the system:

Important Not all Event Logs can be configured on all products. Configurability depends on the hardware platform and licenses in use.

- **Event**: Event logging can be used to determine system status and capture important information pertaining to protocols and tasks in use by the system. This is a global function that will be applied to all contexts, sessions, and processes.

- **Active**: Active logs are operator configurable on a CLI instance-by-CLI instance basis. Active logs configured by an administrative user in one CLI instance cannot be viewed by an administrative user in a different CLI instance. Each active log can be configured with filter and display properties that are independent of those configured globally for the system. Active logs are displayed in real time as events are generated.
- **Trace**: Trace logging can be used to quickly isolate issues that may arise for a particular connected subscriber session. Traces can be taken for a specific call identification (callid) number, IP address, mobile station identification (MSID) number, or username.
- **Monitor**: Monitor logging records all activity associated with a particular session. This functionality is available in order to comply with law enforcement agency requirements for monitoring capabilities of particular subscribers. Monitors can be performed based on a subscriber's MSID or username.
- **Crash**: Crash logging stores useful information pertaining to system software crashes. This information is useful in determining the cause of the crash.

Important Stateful Firewall and NAT supports logging of various messages on screen if logging is enabled for firewall. These logs provide detailed messages at various levels, like critical, error, warning, and debug. Stateful Firewall and NAT attack logs also provide information on the source IP address, destination IP address, protocol, or attack type for any packet dropped due to an attack and are also sent to a syslog server if configured in the system. For more information on logging support for Stateful Firewall and NAT, see the *Logging Support* chapter of *PSF Administration Guide* or *NAT Administration Guide*.

Configuring Event Logging Parameters

The system can be configured to generate logs based on user-defined filters. The filters specify the facilities (system tasks or protocols) that the system is to monitor and severity levels at which to trigger the generation of the event entries.

Event logs are stored in system memory and can be viewed via the CLI. There are two memory buffers that store event logging information. The first buffer stores the active log information. The second buffer stores inactive logging information. The inactive buffer is used as a temporary repository to allow you to view logs without having data be overwritten. Logs are copied to the inactive buffer only through manual intervention.

Each buffer can store up to 50,000 events. Once these buffers reach their capacity, the oldest information is removed to make room for the newest.

To prevent the loss of log data, the system can be configured to transmit logs to a syslog server over a network interface.

Important *For releases after 15.0 MR4,* TACACS+ accounting (CLI event logging) will not be generated for Lawful Intercept users (priv-level 15 and 13).

Configuring Event Log Filters

You can filter the contents of event logs at the Exec mode and Global Configuration mode levels. For additional information, see the *Command Line Interface Reference*.

Exec Mode Filtering

These commands allow you to limit the amount of data contained in logs without changing global logging parameters.

Follow the examples below to filter logs via Exec mode commands.

Active Filtering

logging active [copy runtime filters] [event-verbosity *event_level* **] [pdu-data** *format* **] [pdu-verbosity** *pdu_level* **]**

Notes:

- **copy runtime filters** – Copies the runtime filters and uses that copy to filter the current logging session.
- **event-verbosity** *event_level* – Specifies the level of verboseness to use in logging of events as one of:
  - *min* – Displays minimal information about the event. Information includes event name, facility, event ID, severity level, date, and time.
  - *concise* – Displays detailed information about the event, but does not provide the event source within the system.
  - *full* – Displays detailed information about event, including source information, identifying where within the system the event was generated.
- **pdu-data** *format* – Specifies output format for packet data units when logged as one of:
  - *none* – raw format (unformatted).
  - *hex* – hexadecimal format
  - *hex-ascii* – hexadecimal and ASCII similar to a main-frame dump
- **pdu-verbosity** *pdu_level* – Specifies the level of verboseness to use in logging of packet data units as an integer from 1 through 5, where 5 is the most detailed.

  Save the configuration as described in the *Verifying and Saving Your Configuration* chapter.

Disable or Enable Filtering by Instance(s)

logging filter active facility *facility* **level** *severity_level* **[critical-info | no-critical-info]**

logging filter { disable | enable } facility *facility* **{ all | instance** *instance_number* **}**

Notes:

- **active** – Indicates that only active processes are to have logging options set.

- **disable** – Disables logging for a specific instance or all instances. This keyword is only supported for aaamgr, hamgr and sessmgr facilities.
- **enable** – Enables logging for a specific instance or all instances. This keyword is only supported for aaamgr, hamgr and sessmgr facilities. By default logging is enabled for all instances of aaamgr, hamgr and sessmgr.
- **facility** *facility* and **level** *severity_level* – Configure the logging filter that determines which system facilities should be logged and at what levels. For detailed information, see Specifying Facilities, on page 251 and Event Severities, on page 277.
- **all | instance** *instance_number* – Specifies whether logging will be disabled or enabled for all instances or a specific instance of aaamgr, hamgr or sessmgr. Run the **show session subsystem facility** *facility* command to identify specific instance numbers.

  Note These keywords are only supported with the **disable** and **enable** keywords.

- **level** *severity_level* – Specifies the level of information to be logged from the following list which is ordered from highest to lowest:
  - critical - display critical events
  - error - display error events and all events with a higher severity level
  - warning - display warning events and all events with a higher severity level
  - unusual - display unusual events and all events with a higher severity level
  - info - display info events and all events with a higher severity level
  - trace - display trace events and all events with a higher severity level
  - debug - display all events

  Note This keyword is only supported in conjunction with the **active** keyword.

- **critical-info** – Specifies that events with a category attribute of critical information are to be displayed. Examples of these types of events can be seen at bootup when system processes and tasks are being initiated. This is the default setting.

  no-critical-info – Specifies that events with a category attribute of critical information are not to be displayed.

  Note These keywords are only supported in conjunction with the **active** keyword.

Important To enable logging of a single instance of a facility, you must first disable all instances of the facility (**logging filter disable facility** *facility* **all**) and then enable logging of the specific instance (**logging filter enable facility** *facility* **instance** *instance_number*. To restore default behavior you must re-enable logging of all instances (**logging filter enable facility** *facility* **all**).

You can display the instance numbers for enabled instances per facility using the Exec mode **show instance-logging** command.

Global Configuration Mode Filtering

You can filter the contents of event logs at the Exec mode and Global Configuration mode levels.

Follow the example below to configure run time event logging parameters for the system:

```
configure
   logging filter runtime facility facility level report_level
   logging display { event-verbosity | pdu-data | pdu-verbosity }
   end
```

Notes:

- **facility** *facility* and **level** *severity_level* – Configure the logging filter that determines which system facilities should be logged and at what levels. For detailed information, see Specifying Facilities, on page 251 and Event Severities, on page 277.
- Repeat for every facility that you would like to log.
- *Optional:* Configure event ID restrictions by adding the **logging disable eventid** command. The system provides the ability to restrict the sending of a specific event ID or a range of event IDs to minimize the amount of data logged to that which is most useful. Repeat to disable logging for additional event IDs or event ID ranges.
- If an administrator restricts event logging for an Event ID or Event ID range using the above command (**logging disable eventid**), the system will generate a Critical Event log "cli 30999 critical" as well as an SNMP trap "1361 (DisabledEventIDs)" with the specific Event IDs or Event ID range that was disabled.

  These event logs and traps are enabled by default in this release, and cannot be disabled.
- If an administrator lowers the logging level (using the **logging filter runtime facility *facility* level *report_level*** command below the default level of "error", the system will generate a Critical Event log "cli 30998 critical" as well as an SNMP trap "1362 (LogLevelChanged)" with the specific Event IDs or Event ID range that was disabled.

  These event logs and traps are enabled by default in this release, and cannot be disabled.

The following examples show the CLI output of the traps generated when event logging or logging levels are changed.

```
[local]host# show snmp trap statistics
SNMP Notification Statistics:
  ...
Trap Name                           #Gen #Disc  Disable Last Generated
----------------------------------- ----- -----  ------- --------------------
…
DisabledEventIDs                       1     0        0  2017:05:11:15:35:25
LogLevelChanged                        2     0        0  2017:05:11:15:28:03
```

```
[local]host# show snmp trap history
There are x historical trap records (5000 maximum)

Timestamp                Trap Information
------------------------
```

```
…
Thu May 11 15:28:03 2017 Internal trap notification 1362 (LogLevelChanged) Logging level
of facility resmgr is changed to critical by user #initial-config# context local privilege
 level Security Administrator ttyname /dev/pts/0 address type IPV4 remote ip address 0.0.0.0
…
Thu May 11 15:35:25 2017 Internal trap notification 1361 (DisabledEventIDs) Event IDs from
 100 to 1000 have been disabled by user adminuser context context privilege level security
 administrator ttyname tty address type IPV4 remote ip address 1.2.3.4
…
Mon May 15 10:14:56 2017 Internal trap notification 1362 (LogLevelChanged) Logging level
of facility sitmain is changed to critical by user staradmin context local privilege level
 Security Administrator ttyname /dev/pts/1 address type IPV4 remote ip address 161.44.190.27
```

Save the configuration as described in the *Verifying and Saving Your Configuration* chapter.

Configuring Syslog Servers

Syslog Architecture

System Logging (syslog) is the architecture which produces and sends event information from StarOS over the UDP transport layer to a centralized Event Message Collector. Syslog uses a client-server architecture:

- **Syslog Client:** A set of processes running on StarOS products which operate as the sending device for event messages.
- **Syslog Server:** An external server configured to receive the event messages sent from StarOS products.

StarOS products transport event messages using the Syslog Protocol without expecting acknowledgement of receipt. The system forwards event messages regardless if a Syslog Server is available to receive the messages.

Configuring the System to Sent Event Messages to an External Syslog Server

Information generated by the run time event logging filters can be transmitted to a syslog server for permanent storage.

Important The data transmitted to the Syslog server is meant to be used for informational purposes. Functions such as billing and performance monitoring should not be based on syslogs.

Important Although the system provides the flexibility to configure syslog servers on a context-by-context basis, it is recommended that all servers be configured in the *local* context in order to isolate the log traffic from the network traffic.

Use the following example to configure syslog servers:

```
configure
  context local
    logging syslog ip_address
    end
```

Notes:

- *ip_address* specifies the IP address of a system log server on the network in IPv4 dotted-decimal or IPv6 colon-separated-hexadecimal notation.
- Several optional keywords are available for the **logging syslog** command. Refer to the *Context Configuration Mode Commands* chapter in the *Command Line Interface Reference* for more information.
- Repeat as necessary to configure additional syslog servers. There is no limit to the number of syslog servers that can be configured.

Refer to the **logging** command in the *Command Line Reference, Modes C-D* for more information.

Save the configuration as described in the *Verifying and Saving Your Configuration* chapter.

Configuring Active Logs

Active logs are event logs that are operator configurable on a CLI instance-by-CLI instance basis. Active logs configured by an administrative user in one CLI instance are not displayed to an administrative user in a different CLI instance. Each active log can be configured with filter and display properties that are independent of those configured globally for the system. Active logs are displayed in real time as they are generated.

Active logs are not written to the active memory buffer by default. To write active logs to the active memory buffer execute the following command in the Global Configuration mode:

```
[local]host_name(config)# logging runtime buffer store all-events
```

When active logs are written to the active memory buffer, they are available to all users in all CLI instances.

Use the following example to configure active logging in Global Configuration mode:

```
[local]host_name(config)# logging filter runtime facility facility level report_level
```

Notes:

- Configure the logging filter that determines which system facilities should be logged and at what levels. For detailed information, see Specifying Facilities, on page 251 and Event Severities, on page 277.
- Repeat for every facility that you would like to log.
- *Optional:* Configure event ID restrictions by adding the **logging disable eventid** command. The system provides the ability to restrict the sending of a specific event ID or a range of event IDs to minimize the amount of data logged to that which is most useful. Repeat to disable logging for additional event IDs or event ID ranges.
- A number of keyword options/variables are available for the Exec mode **logging active** command. Refer to the *Exec Mode Commands* chapter in the *Command Line Interface Reference* for more information.

Once all of the necessary information has been gathered, the Active log display can be stopped by entering the following command in the Exec mode:

```
no logging active
```

Specifying Facilities

Important The actual facilities available for logging vary by platform type, StarOS version and installed product licenses.

The following facilities can be configured for logging event data:

- **a10**: A10 interface facility
- **a11**: A11 interface facility
- **a11mgr**: A11 Manager facility
- **aaa-client**: Authentication, Authorization and Accounting (AAA) client facility
- **aaamgr**: AAA manager logging facility
- **aaaproxy**: AAA Proxy facility
- **aal2**: ATM Adaptation Layer 2 (AAL2) protocol logging facility
- **acl-log**: Access Control List (ACL) logging facility
- **acsctrl**: Active Charging Service (ACS) Controller facility
- **acsmgr**: ACS Manager facility
- **afctrl**: Fabric Controller facility [ASR 5500 only]
- **afmgr**: Fabric Manager logging facility [ASR 5500 only]
- **alarmctrl**: Alarm Controller facility
- **alcap**: Access Link Control Application Part (ALCAP) protocol logging facility
- **alcapmgr**: ALCAP manager logging facility
- **all**: All facilities
- **bfd**: Bidirectional Forwarding Detection (BFD) protocol logging facility
- **bgp**: Border Gateway Protocol (BGP) facility
- **bindmux**: IPCF BindMux-Demux Manager logging facility
- **bngmgr**: Broadband Network Gateway (BNG) Demux Manager logging facility
- **bssap+**: Base Station Sub-system Application Part+ protocol facility for the login interface between the SGSN and the MSC/VLR (2.5G and 3G)
- **bssgp**: Base Station Sub-system GPRS Protocol logging facility handles exchange information between the SGSN and the BSS (2.5G only)
- **callhome**: Call Home application logging facility
- **cap**: CAMEL Application Part (CAP) logging facility for protocol used in prepaid applications (2.5G and 3G)

- **cbsmgr**: Cell Broadcasting Service (CBS) logging facility [HNBGW]
- **cdf**: Charging Data Function (CDF) logging facility
- **cfctrl**: Content filtering controller logging facility
- **cfmgr**: Content filtering manager logging facility
- **cgw**: Converged Access Gateway (CGW) logging facility
- **cli**: Command Line Interface (CLI) logging facility
- **cmp**: Certificate Management Protocol (IPSec) logging facility
- **confdmgr**: ConfD Manager proclet (NETCONF) logging facility
- **connectedapps**: SecGW ASR 9000 oneP communication procotol
- **connproxy**: Controller Proxy logging facility
- **credit-control**: Credit Control (CC) facility
- **csp**: Card/Slot/Port controller facility
- **css**: Content Service Selection (CSS) facility
- **css-sig**: CSS RADIUS Signaling facility
- **cx-diameter**: Cx Diameter Messages facility [CSCF <--> HSS]
- **data-mgr**: Data Manager Framework logging facility
- **dcardctrl**: IPSec Daughter Card Controller logging facility
- **dcardmgr**: IPSec Daughter Card Manager logging facility
- **demuxmgr**: Demux Manager API facility
- **dgmbmgr**: Diameter Gmb Application Manager logging facility
- **dhcp**: Dynamic Host Configuration Protocol (DHCP) logging facility
- **dhcpv6**: DHCPv6
- **dhost**: Distributed Host logging facility
- **diabase**: Diabase messages facility
- **diactrl**: Diameter Controller proclet logging facility
- **diameter**: Diameter endpoint logging facility
- **diameter-acct**: Diameter Accounting
- **diameter-auth**: Diameter Authentication
- **diameter-dns**: Diameter DNS subsystem
- **diameter-ecs**: ACS Diameter signaling facility
- **diameter-engine**: Diameter version2 engine logging facility
- **diameter-hdd**: Diameter Horizontal Directional Drilling (HDD) Interface facility

- **diameter-svc**: Diameter Service
- **diamproxy**: DiamProxy logging facility
- **doulosuemgr**: Doulos (IMS-IPSec-Tool) user equipment manager
- **dpath**: IPSec Data Path logging facility
- **drvctrl**: Driver Controller logging facility
- **eap-diameter**: Extensible Authentication Protocol (EAP) IP Sec urity facility
- **eap-ipsec**: Extensible Authentication Protocol (EAP) IPSec facility
- **eap-sta-s6a-s13-s6b-diameter**: EAP/STA/S6A/S13/S6B Diameter messages facility
- **ecs-css**: ACSMGR <-> Session Manager Signalling Interface facility
- **egtpc**: eGTP-C logging facility
- **egtpmgr**: enhanced GPRS Tunneling Protocol (eGTP) manager logging facility
- **egtpu**: eGTP-U logging facility
- **embms**: evolved Multimedia Broadcast Multicast Services Gateway facility
- **embms**: eMBMS Gateway Demux facility
- **epdg**: evolved Packet Data (ePDG) gateway logging facility
- **event-notif**: Event Notification Interface logging facility
- **evlog**: Event log facility
- **famgr**: Foreign Agent manager logging facility
- **firewall**: Firewall logging facility
- **fng**: Femto Network Gateway (FNG) logging facility
- **gbmgr**: SGSN Gb Interface Manager facility
- **gmm**:
  - For 2.5G: Logs the GPRS Mobility Management (GMM) layer (above LLC layer)
  - For 3G: Logs the access application layer (above the RANAP layer)
- **gprs-app**: GPRS Application logging facility
- **gprs-ns**: GPRS Network Service Protocol (layer between SGSN and the BSS) logging facility
- **gq-rx-tx-diameter**: Gq/Rx/Tx Diameter messages facility
- **gss-gcdr**: GTPP Storage Server GCDR facility
- **gtpc**: GTP-C protocol logging facility
- **gtpcmgr**: GTP-C protocol manager logging facility
- **gtpp**: GTP-prime protocol logging facility

- **gtpu**: GTP-U protocol logging facility
- **gtpumgr**: GTP-U Demux manager
- **gx-ty-diameter**: Gx/Ty Diameter messages facility
- **gy-diameter**: Gy Diameter messages facility
- **h248prt**: H.248 port manager facility
- **hamgr**: Home Agent manager logging facility
- **hat**: High Availability Task (HAT) process facility
- **hdctrl**: HD Controller logging facility
- **henbapp**: Home Evolved NodeB (HENB) App facility (Do not use this keyword for HENB-GW in Release 20)
- **henbgw**: HENB-GW facility (Do not use this keyword for HENB-GW in Release 20)
- **henbgw-pws**: HENB-GW Public Warning System logging facility (Do not use this keyword for HENB-GW in Release 20)
- **henbgw-sctp-acs**: HENB-GW access Stream Control Transmission Protocol (SCTP) facility (Do not use this keyword for HENB-GW in Release 20)
- **henbgw-sctp-nw**: HENBGW network SCTP facility (Do not use this keyword for HNB-GW in Release 20)
- **henbgwdemux**: HENB-GW Demux facility (Do not use this keyword for HNB-GW in Release 20)
- **henbgwmgr**: HENB-GW Manager facility (Do not use this keyword for HNB-GW in Release 20)
- **hnb-gw**: HNB-GW (3G Femto GW) logging facility (Do not use this keyword for HNB-GW in Release 20)
- **hnbmgr**: HNB-GW Demux Manager logging facility (Do not use this keyword for HNB-GW in Release 20)
- **hss-peer-service**: Home Subscriber Server (HSS) Peer Service facility
- **iftask**: Internal Forwarder Task (Intel DPDK) used on VPC-SI and VPC-DI platforms
- **igmp**: Internet Group Management Protocol (IGMP)
- **ikev2**: Internet Key Exchange version 2 (IKEv2)
- **ims-authorizatn**: IP Multimedia Subsystem (IMS) Authorization Service facility
- **ims-sh**: HSS Diameter Sh Interface Service facility
- **imsimgr**: SGSN IMSI Manager facility
- **imsue**: IMS User Equipment (IMSUE) facility
- **ip-arp**: IP Address Resolution Protocol facility
- **ip-interface**: IP interface facility
- **ip-route**: IP route facility

- **ipms**: Intelligent Packet Monitoring System (IPMS) logging facility
- **ipne**: IP Network Enabler (IPNE) facility
- **ipsec**: IP Security logging facility
- **ipsecdemux**: IPSec demux logging facility
- **ipsg**: IP Service Gateway interface logging facility
- **ipsgmgr**: IP Services Gateway facility
- **ipsp**: IP Pool Sharing Protocol logging facility
- **kvstore**: Key/Value Store (KVSTORE) Store facility
- **l2tp-control**: Layer 2 Tunneling Protocol (L2TP) control logging facility
- **l2tp-data**: L2TP data logging facility
- **l2tpdemux**: L2TP Demux Manager logging facility
- **l2tpmgr**: L2TP Manager logging facility
- **lagmgr**: Link Aggregation Group (LAG) manager logging facility
- **lcs**: Location Services (LCS) logging facility
- **ldap**: Lightweight Directory Access Protocol (LDAP) messages logging facility
- **li**: Refer to the *Lawful Intercept Configuration Guide* for a description of this command.
- **linkmgr**: SGSN/BSS SS7 Link Manager logging facility (2.5G only)
- **llc**: Logical Link Control (LLC) Protocol logging facility; for SGSN: logs the LLC layer between the GMM and the BSSGP layers for logical links between the MS and the SGSN
- **local-policy**: Local Policy Service facility
- **location-service**: Location Services facility
- **m3ap**: M3 Application Protocol facility
- **m3ua**: M3UA Protocol logging facility
- **magmgr**: Mobile Access Gateway manager logging facility
- **map**: Mobile Application Part (MAP) protocol logging facility
- **megadiammgr**: MegaDiameter Manager (SLF Service) logging facility
- **mme-app**: Mobility Management Entity (MME) Application logging facility
- **mme-embms**: MME evolved Multimedia Broadcast Multicast Service facility
- **mme-misc**: MME miscellaneous logging facility
- **mmedemux**: MME Demux Manager logging facility
- **mmemgr**: MME Manager facility
- **mmgr**: Master Manager logging facility

- **mobile-ip**: Mobile IP processes
- **mobile-ip-data**: Mobile IP data facility
- **mobile-ipv6**: Mobile IPv6 logging facility
- **mpls**: Multiprotocol Label Switching (MPLS) protocol logging facility
- **mrme**: Multi Radio Mobility Entity (MRME) logging facility
- **mseg-app**: Mobile Services Edge Gateway (MSEG) application logging facility (This option is not supported in this release.)
- **mseg-gtpc**: MSEG GTP-C application logging facility (This option is not supported in this release.)
- **mseg-gtpu**: MSEG GTP-U application logging facility (This option is not supported in this release.)
- **msegmgr**: MSEG Demux Manager logging facility (This option is not supported in this release.)
- **mtp2**: Message Transfer Part 2 (MTP2) Service logging facility
- **mtp3**: Message Transfer Part 3 (MTP3) Protocol logging facility
- **multicast-proxy**: Multicast Proxy logging facility
- **nas**: Non-Access Stratum (NAS) protocol logging facility [MME 4G]
- **netwstrg**: Network Storage facility
- **npuctrl**: Network Processor Unit Control facility
- **npumgr**: Network Processor Unit Manager facility
- **npumgr-acl**: NPUMGR ACL logging facility
- **npumgr-drv**: NPUMGR DRV logging facility
- **npumgr-flow**: NPUMGR FLOW logging facility
- **npumgr-fwd**: NPUMGR FWD logging facility
- **npumgr-init**: NPUMGR INIT logging facility
- **npumgr-lc**: NPUMGR LC logging facility
- **npumgr-port**: NPUMGR PORT logging facility
- **npumgr-recovery**: NPUMGR RECOVERY logging facility
- **npumgr-rri**: NPUMGR RRI (Reverse Route Injection) logging facility
- **npumgr-vpn**: NPUMGR VPN logging facility
- **ntfy-intf**: Notification Interface logging facility [Release 12.0 and earlier versions only]
- **ocsp**: Online Certificate Status Protocol logging facility.
- **orbs**: Object Request Broker System logging facility
- **ospf**: OSPF protocol logging facility
- **ospfv3**: OSPFv3 protocol logging facility

- **p2p**: Peer-to-Peer Detection logging facility
- **pagingmgr**: PAGINGMGR logging facility
- **pccmgr**: Intelligent Policy Control Function (IPCF) Policy Charging and Control (PCC) Manager library
- **pdg**: Packet Data Gateway (PDG) logging facility
- **pdgdmgr**: PDG Demux Manager logging facility
- **pdif**: Packet Data Interworking Function (PDIF) logging facility
- **pgw**: Packet Data Network Gateway (PGW) logging facility
- **pmm-app**: Packet Mobility Management (PMM) application logging facility
- **ppp**: Point-To-Point Protocol (PPP) link and packet facilities
- **pppoe**: PPP over Ethernet logging facility
- **proclet-map-frwk**: Proclet mapping framework logging facility
- **push**: VPNMGR CDR push logging facility
- **radius-acct**: RADIUS accounting logging facility
- **radius-auth**: RADIUS authentication logging facility
- **radius-coa**: RADIUS change of authorization and radius disconnect
- **ranap**: Radio Access Network Application Part (RANAP) Protocol facility logging info flow between SGSN and RNS (3G)
- **rct**: Recovery Control Task logging facility
- **rdt**: Redirect Task logging facility
- **resmgr**: Resource Manager logging facility
- **rf-diameter**: Diameter Rf interface messages facility
- **rip**: Routing Information Protocol (RIP) logging facility [RIP is not supported at this time.]
- **rlf**: Rate Limiting Function (RLF) logging facility
- **rohc**: Robust Header Compression (RoHC) facility
- **rsvp**: Reservation Protocol logging facility
- **rua**: RANAP User Adaptation (RUA) [3G Femto GW - RUA messages] logging facility
- **s102**: S102 protocol logging facility
- **s102mgr**: S102Mgr logging facility
- **s1ap**: S1 Application Protocol (S1AP) Protocol logging facility
- **sabp**: Service Area Broadcast Protocol (SABP) logging facility
- **saegw**: System Architecture Evolution (SAE) Gateway facility
- **sbc**: SBc protocol logging facility

- **sccp:** Signalling Connection Control Part (SCCP) Protocol logging (connection-oriented messages between RANAP and TCAP layers).
- **sct**: Shared Configuration Task logging facility
- **sctp**: Stream Control Transmission Protocol (SCTP) Protocol logging facility
- **sef_ecs**: Severely Errored Frames (SEF) APIs printing facility
- **sess-gr**: SM GR facility
- **sessctrl**: Session Controller logging facility
- **sessmgr**: Session Manager logging facility
- **sesstrc**: session trace logging facility
- **sft**: Switch Fabric Task logging facility
- **sgs**: SGs interface protocol logging facility
- **sgsn-app**: SGSN-APP logging various SGSN "glue" interfaces (for example, between PMM, MAP, GPRS-FSM, SMS).
- **sgsn-failures**: SGSN call failures (attach/activate rejects) logging facility (2.5G)
- **sgsn-gtpc**: SGSN GTP-C Protocol logging control messages between the SGSN and the GGSN
- **sgsn-gtpu**: SGSN GTP-U Protocol logging user data messages between the SGSN and GGSN
- **sgsn-mbms-bearer**: SGSN Multimedia Broadcast/Multicast Service (MBMS) Bearer app (SMGR) logging facility
- **sgsn-misc**: Used by stack manager to log binding and removing between layers
- **sgsn-system**: SGSN System Components logging facility (used infrequently)
- **sgsn-test**: SGSN Tests logging facility; used infrequently
- **sgtpcmgr**: SGSN GTP-C Manager logging information exchange through SGTPC and the GGSN
- **sgw**: Serving Gateway facility
- **sh-diameter**: Sh Diameter messages facility
- **sitmain**: System Initialization Task main logging facility
- **slmgr**: Smart Licensing manager logging facility
- **sls**: Service Level Specification (SLS) protocol logging facility
- **sm-app**: SM Protocol logging facility
- **sms**: Short Message Service (SMS) logging messages between the MS and the SMSC
- **sndcp**: Sub Network Dependent Convergence Protocol (SNDCP) logging facility
- **snmp**: SNMP logging facility
- **sprmgr**: IPCF Subscriber Policy Register (SPR) manager logging facility
- **srdb**: Static Rating Database

- **srp**: Service Redundancy Protocol (SRP) logging facility
- **sscfnni**: Service-Specific Coordination Function for Signaling at the Network Node Interface (SSCF-NNI) logging facility
- **sscop**: Service-Specific Connection-Oriented Protocol (SSCOP) logging facility
- **ssh-ipsec**: Secure Shell (SSH) IP Security logging facility
- **ssl**: Secure Socket Layer (SSL) message logging facility
- **stat**: Statistics logging facility
- **supserv**: Supplementary Services logging facility [H.323]
- **system**: System logging facility
- **tacacsplus**: TACACS+ Protocol logging facility
- **tcap**: TCAP Protocol logging facility
- **testctrl**: Test Controller logging facility
- **testmgr**: Test Manager logging facility
- **threshold**: threshold logging facility
- **ttg**: Tunnel Termination Gateway (TTG) logging facility
- **tucl**: TCP/UDP Convergence Layer (TUCL) logging facility
- **udr**: User Data Record (UDR) facility (used with the Charging Service)
- **user-data**: User data logging facility
- **user-l3tunnel**: User Layer 3 tunnel logging facility
- **usertcp-stack**: User TCP Stack
- **vim**: Voice Instant Messaging (VIM) logging facility
- **vinfo**: VINFO logging facility
- **vmgctrl**: Virtual Media Gateway (VMG) controller facility
- **vmgctrl**: VMG Content Manager facility
- **vpn**: Virtual Private Network logging facility
- **vpp**: Vector Packet Processing (VPP) logging facility
- **wimax-data**: WiMAX DATA
- **wimax-r6**: WiMAX R6
- **wsg**: Wireless Security Gateway (ASR 9000 Security Gateway)
- **x2gw-app**: X2GW (X2 proxy Gateway, eNodeB) application logging facility
- **x2gw-demux**: X2GW demux task logging facility

Configuring Trace Logging

Trace logging is useful for quickly resolving issues for specific sessions that are currently active. They are temporary filters that are generated based on a qualifier that is independent of the global event log filter configured using the **logging filter** command in the Exec mode. Like event logs, however, the information generated by the logs is stored in the active memory buffer.

All debug level events associated with the selected call are stored.

Important Trace logs impact session processing. They should be implemented for debug purposes only.

Use the following example to configure trace logs in the Exec mode:

```
[local]host_name# logging trace { callid call_id | ipaddr ip_address | msid ms_id
 | username username }
```

Once all of the necessary information has been gathered, the trace log can be deleted by entering the following command:

```
[local]host_name# no logging trace { callid call_id | ipaddr ip_address | msid
ms_id | username username }
```

Configuring Monitor Logs

Monitor logging records all activity associated with all of a particular subscriber's sessions. This functionality is available in compliance with law enforcement agency requirements for monitoring capabilities of particular subscribers.

Monitors can be performed based on a subscriber's MSID or username, and are only intended to be used for finite periods of time as dictated by the law enforcement agency. Therefore, they should be terminated immediately after the required monitoring period.

This section provides instructions for enabling and disabling monitor logs.

Enabling Monitor Logs

Use the following example to configure monitor log targets:

```
configure
  logging monitor { ip_addr | ipv6_addr | msid id | username name }
  end
```

Repeat to configure additional monitor log targets.

Disabling Monitor Logs

Use the following example to disable monitor logs:

```
configure
  no logging monitor { ip_addr | ipv6_addr | msid id | username name }
  end
```

Viewing Logging Configuration and Statistics

Logging configuration and statistics can be verified by entering the following command from the Exec mode:

```
[local]host_name# show logging [ active | verbose ]
```

When no keyword is specified, the global filter configuration is displayed as well as information about any other type of logging that is enabled.

The following table provides information and descriptions of the statistics that are displayed when the **verbose** keyword is used.

Table 14: Logging Configuration and Statistics Commands

Field	Description
General Logging Statistics	
Total events received	Displays the total number of events generated by the system.
Number of applications receiving events	Displays the number of applications receiving the events.
Logging Source Statistics	
Event sequence ids by process	Displays a list of system processes that have generated events and the reference identification number of the event that was generated.
Msg backlog stat with total cnt	Displays the number of event messages that have been back logged in comparison to the total number of events generated.
LS L2 filter drop rate	Displays the percentage of logging source (LS) layer 2 (L2) event drops.
Abnormal Log Source Statistics	Displays abnormal logging source (LS) statistics, if any.
Runtime Logging Buffer Statistics	
Active buffer	Displays the number of events currently logged in the active memory buffer and a timestamp for the oldest and most recent entries in the buffer.
Inactive buffer	Displays the number of events currently logged in the inactive memory buffer.

Viewing Event Logs Using the CLI

Event logs generated by the system can be viewed in one of the following ways:

- **From the syslog server:** If the system is configured to send logs to a syslog server, the logs can be viewed directly on the syslog server.
- **From the system CLI:** Logs stored in the system memory buffers can be viewed directly from the CLI.
- **From the console port:** By default, the system automatically displays events over the console interface to a terminal provided that there is no CLI session active.

This section provides instructions for viewing event logs using the CLI. These instructions assume that you are at the root prompt for the Exec mode.

Step 1 Copy the active log memory buffer to the inactive log memory buffer.

When the active log memory buffer is copied to the inactive log memory buffer existing information in the inactive log memory buffer is deleted.

Both active and inactive event log memory buffers can be viewed using the CLI in Exec mode. However, it is preferable to view the inactive log in order to prevent any data from being over-written. The information from the active log buffer can be copied to the inactive log buffer by entering the following command:

```
[local]host_name# logs checkpoint
```

Step 2 View the logs by entering the following command:

```
[local]host_name# show logs
```

A number of optional keywords/variables are available for the **show logs** command. Refer to the *Exec Mode Show Commands* chapter in the *Command Line Interface Reference* for more information.

Configuring and Viewing Crash Logs

In the unlikely even of a software crash, the system stores information that could be useful in determining the reason for the crash. This information can be maintained in system memory or it can be transferred and stored on a network server.

The system supports the generation of the following two types of logs:

- **Crash log:** Crash logs record all possible information pertaining to a software crash (full core dump). Due to their size, they can not be stored in system memory. Therefore, these logs are only generated if the system is configured with a Universal Resource Locator (URL) pointing to a local device or a network server where the log can be stored.
- **Abridged crash log:** Crash event records are automatically generated when a software crash occurs and are stored in flash memory on management cards. The abridged crash log contains a list crash event records along with associated dump files. This log allows you to view event records and dump files via CLI commands.

Crash Logging Architecture

The crash log is a persistent repository of crash event information. Each event is numbered and contains text associated with a CPU (minicore), NPU or kernel crash. The logged events are recorded into fixed length records and stored in /flash/crashlog2.

Whenever a crash occurs, the following crash information is stored:

1. The event record is stored in /flash/crashlog2 file (the crash log).
2. The associated minicore, NPU or kernel dump file is stored in the /flash/crsh2 directory.
3. A full core dump is stored in a user configured directory.

Important The crashlog2 file along with associated minicore, NPU and kernel dumps are automatically synchronized across redundant management cards (SMC, MIO/UMIO). Full core dumps are not synchronized across management cards.

The following behaviors apply to the crash logging process.

- When a crash event arrives on an active management card, the event record is stored in its crashlog2 file along with the minicore, NPU, or kernel dump file in /flash/crsh2. The crash event and dump file are also automatically stored in the same locations on the standby management card.
- When a crash log entry is deleted via CLI command, it is deleted on both the active and standby management cards.
- When a management card is added or replaced, active and standby cards will automatically synchronize crash logs and dump files.
- When a crash event is received and the crash log file is full, the oldest entry in the crash log and its related dump file will be replaced with the latest arrived event and dump file on both management cards. Information for a maximum of 120 crash events can be stored on management cards.
- Duplicate crash events bump the count of hits in the existing record and update the new record with the old crash record. Additions to the count use the timestamp for the first time the event happened.

Configuring Software Crash Log Destinations

The system can be configured to store software crash log information to any of the following locations:

- On the ASR 5500:
  - **Flash memory:** Installed on the active MIO/UMIO [abridged crash log and associated dump files only]
  - **USB memory stick:** Installed in the USB slot on the active MIO/UMIO
- On VPC
  - **Flash memory:** Accessible by the virtual machine
  - **USB memory stick:** Installed in the USB slot of the platform (USB slot has been enabled via the hypervisor)

- **Network Server:** Any workstation or server on the network that the system can access using the Trivial File Transfer Protocol (TFTP), the File Transfer Protocol (FTP), the Secure File Transfer Protocol (SFTP), or the Hyper-Text Transfer Protocol (HTTP); this is recommended for large network deployments in which multiple systems require the same configuration

Important In release 20.0 and higher Trusted StarOS builds, FTP is not supported.

Crash log files (full core dumps) are written with unique names as they occur to the specified location. The name format is *crash-card-cpu-time-core*. Where *card* is the card slot, *cpu* is the number of the CPU on the card, and *time* is the Portable Operating System Interface (POSIX) timestamp in hexadecimal notation.

Use the following example to configure a software crash log destination in the Global Configuration mode:

```
configure
  crash enable [ encrypted ] url crash_url
  end
```

Notes:

- Refer to the *Global Configuration Mode Commands* chapter in the *Command Line Interface Reference* for more information on this command.
- Repeat to configure additional software crash log destinations. There is no limit to the number of destinations that can be configured.

Save the configuration as described in the *Verifying and Saving Your Configuration* chapter.

Viewing Abridged Crash Log Information Using the CLI

You can view abridged crash information that is stored as a set of event records in flash memory on management cards (**/flash/crashlog2**). Each crash event record has an associated dump file (minicore, NPU or kernel) that can also be displayed (**/flash/crsh2**)

Follow the instructions in this section to view software crash events that have occurred on the system. These instructions assume that you are at the root prompt for the Exec mode.

Step 1 View a list of software crash events by entering the following Exec mode command:

```
[local]host_name# show crash { all | list | number crash_num }
```

Notes:

- Run **show crash list** to obtain the number for a specific crash event.
- Run **show crash number** *crash_num* to display the output for the target crash event.

The resulting output may not be the same for all platforms:

Information about similar crash events is suppressed in the output of this command.

Step 2 View the dump file associated with a specific crash event.

The information contained in the dump file helps identify and diagnose any internal or external factors causing the software to crash.

- Crash # – unique number assigned by StarOS when logging the crash event

- SW Version – StarOS build release in format: RR.n(bbbbb)
- Similar Crash Count – number of similar crashes
- Time of first crash – timestamp when first crash occurred in format: YYYY-MMM-DD+hh:mm:ss
- Failure message – text of event message
- Function – code identifier
- Process – where the crash occurred (Card, CPU, PID, etc.)
- Crash time – timestamp for when the crash occurred in the format: YYYY-MMM-DD+hh:mm:ss time zone
- Recent errno – text of most recent error number.
- Stack – memory stack information
- Last Bounce – information about the messaging received prior to the crash
- Registers – memory register contents
- Current inbound message – hexadecimal information for the current inbound message
- Address Map
- Recent heap activity (oldest first)
- Recent events (oldest first)
- Profile depth

The informational content of each crash log entry varies based on the type of crash and the StarOS release.

Reducing Excessive Event Logging

Event logging (evlogd) is a shared medium that captures event messages sent by StarOS facilities. When one or more facilities continuously and overwhelmingly keep sending a high volume of event messages, the remaining non-offender facilities are impacted. This scenario degrades system performance, especially as the number of facilities generating logs increases.

Rate-control of event message logging is handled in the Log Source path. Essentially, every second a counter is set to zero and is incremented for each log event that is sent to evlogd. If the count reaches a threshold before the second is up, the event is sent, queued or dropped (if the evlogd messenger queue is full).

When any facility exceeds the upper threshold set with this command for the rate of message logging and remains in the same state for prolonged interval, StarOS notifies the user via an SNMP trap or alarm.

A new threshold command allows a user to specify the percentage of facility event queue full. When this threshold is exceeded, an SNMP trap and alarm are generated that specifies the offending facility.

The formats for the SNMP traps associated with this command are as follows:

- **ThreshLSLogsVolume**

```
<timestamp> Internal trap notification <trap_id> (ThreshLSLogsVolume) threshold
<upper_percent>%
measured value <actual_percent>% for facility <facility_name> instance <instance_id>
```

- **ThreshClearLSLogsVolume**

```
<timestamp> Internal trap notification <trap_id> (ThreshClearLSLogsVolume) threshold
<upper_percent>%
measured value <actual_percent>% for facility <facility_name> instance <instance_id>
```

If a trigger condition occurs within the polling interval, the alert or alarm will not be generated until the end of the polling interval.

Both traps can be enabled or suppressed via the Global Configuration mode **snmp trap** command.

Configuring Log Source Thresholds

There are three Global Configuration mode commands associated with configuring and implementing Log Source thresholds.

1. **threshold ls-logs-volume** – sets the parameters for the upper and lower thresholds for generating and clearing traps/alarms respectively.
2. **threshold poll ls-logs-volume interval** – establishes the polling interval for this threshold.
3. **threshold monitoring ls-logs-volume** – turns monitoring of this threshold on and off.

Use the following example to configure syslog servers:

```
configure
  [ default ] threshold ls-logs-volume upper_percent [ clear lower_percent ]
  [ default ] threshold poll ls-logs-volume interval duration
  [ no ] threshold monitoring ls-logs-volume
  end
```

Notes:

- *upper_percent* and *lower_percent* are expressed as integers from 0 to 100. Default value for *upper_percent* is 90%. If *lower_percent* is not specified, the default clear value is *upper_percent*.
- **threshold poll ls-logs-volume interval** sets the polling interval in seconds. The default interval is 300 seconds (5 minutes).
- **threshold monitoring ls-logs-volume** enables or disables this feature.

You can verify the configuration of this threshold by running the Exec mode **show threshold** command.

Save the configuration as described in the *Verifying and Saving Your Configuration* chapter.

Checkpointing Logs

Checkpointing identifies logged data as previously viewed or marked. Checkpointing allows you to only display log information since the last checkpoint.

Individual logs may have up to 50,000 events in the active log. Checkpointing the logs results in at most 50,000 events being in the inactive log files. This gives a maximum of 100,000 events in total which are available for each facility logged.

You check point log data via the Exec mode logs checkpoint command to set the log contents to a well-known point prior to special activities taking place. This command may also be a part of periodic regular maintenance to manage log data.

Checkpointing logs moves the current log data to the inactive logs. Only the most recently check pointed data is retained in the inactive logs. A subsequent check pointing of the logs results in the prior check pointed inactive log data being cleared and replaced with the newly check pointed data. Checkpointed log data is not available for viewing.

Important Checkpointing logs should be done periodically to prevent the log files becoming full. Logs which have 50,000 events logged will discard the oldest events first as new events are logged.

Important An Inspector-level administrative user cannot execute this command.

Saving Log Files

Log files can be saved to a file in a local or remote location specified by a URL. Use the following Exec mode command to save log files:

save logs { *url* } [**active**]] [**inactive**] [**callid** *call_id*] [**event-verbosity** *evt_verboseness*] [**facility** *facility*] [**level** *severity_level*] [**pdu-data** *pdu_format*] [**pdu-verbosity** *pdu_verboseness*] [**since** *from_date_time* [**until** *to_date_time*]] [| { **grep** *grep_options* | **more** }]

For detailed information on the **save logs** command, see the *Exec Mode Commands* chapter in the *Command Line Interface Reference*.

Event ID Overview

Important The use of event IDs depends on the platform type and the licenses running on the platform.

Identification numbers (IDs) are used to reference events as they occur when logging is enabled on the system. As described previously, logs are collected on a per facility basis. Each facility possesses its own range of event IDs as indicated in the following table.

Table 15: System Facilities and Event ID Ranges

Facility	Description	Event ID Range
a10	A10 Protocol Facility	28000-28999
a11	A11 Protocol Facility	29000-29999
a11mgr	A11 Manager Facility	9000-9999
aaa-client	AAA Client Facility	6000-6999
aaamgr	AAA Manager Facility	36000-36999
aaaproxy	AAA Proxy Facility	64000-64999
aal2	AAL2 Protocol Facility	173200-173299
acl-log	IP Access Control List (ACL) Facility	21000-21999

Facility	Description	Event ID Range
acsctrl	Active Charging Service Controller (ACSCtrl) Facility	90000-90999
acsmgr	Active Charging Service Manager (ACSMgr) Facility	91000-91999
afctrl	Ares Fabric Controller (ASR 5500 only)	186000-186999
afmgr	Ares Fabric Manager (ASR 5500 only)	187000-187999
alarmctrl	Alarm Controller Facility	65000-65999
alcap	Access Link Control Application Part (ALCAP) Protocol Facility	160900-161399
alcapmgr	ALCAP Manager Facility	160500-160899
asf	ASF Facility	73000-73999
asfprt	ASFPRT Facility	59000-59999
asngwmgr	Access Service Network (ASN) Gateway Manager Facility	100000-100499
asnpcmgr	ASN Paging/Location-Registry Manager Facility	100500-100999
bcmcs	Broadcast/Multicast Service (BCMCS) Facility	109000-109999
bfd	Bidirectional Forwarding Detection (BFD) Protocol Facility	170500-170999
bgp	Border Gateway Protocol (BGP) Facility	85000-85999
bindmux	BindMux Manager Facility [Intelligent Policy Control Function (IPCF)]	158200-158999
bngmgr	Broadband Network Gateway (BNG) Manager Facility	182000-182999
bssap	Base Station System Application Part+ (BSSAP+) Service Facilities	131000-131199
bssgp	Base Station System GPRS Protocol (BSSGP) Facility	115050-115099
callhome	Call Home Facility	173600-173999
cap	CAMEL Application Part (CAP) Facility	87900-88099
chatconf	CHATCONF Facility	74000-74999

Facility	Description	Event ID Range
cli	Command Line Interface (CLI) Facility	30000-30999
connproxy	Connection Proxy Facility	190000-190999
crdt-ctl	Credit Control Facility	127000-127999
csg	Closed Subscriber Groups (CSG) Facility	188000-188999
csg-acl	CSG Access Control List (ACL) Facility	189000-189999
csp	Card/Slot/Port (CSP) Facility	7000-7999
css	Content Steering Service (CSS) Facility [ESC]	77000-77499
css-sig	Content Service Selection (CSS) RADIUS Signaling Facility	77500-77599
cx-diameter	Cx Diameter Message Facility	92840-92849
dcardctrl	Daughter Card Controller Facility	62000-62999
dcardmgr	Daughter Card Manager Facility	57000-57999
demuxmgr	Demux Manager Facility	110000-110999
dgmbmgr	Diameter Gmb (DGMB) Application Manager Facility	126000-126999
dhcp	DHCP Facility	53000-53999
dhcpv6	DHCPv6 Protocol Facility	123000-123999
dhost	Distributed Host Manager Facility	83000-83999
diameter	Diameter Endpoint Facility	92000-92599
diabase	Diabase Message Facility	92800-92809
diameter-acct	Diameter Accounting Protocol Facility	112000-112999
diameter-auth	Diameter Authentication Protocol Facility	111000-111999
diameter-dns	Diameter DNS Subsystem Facility	92600-92699
diameter-ecs	ECS Diameter Signaling Facility	81990-81999
diameter-hdd	Diameter Horizontal Directional Drilling (HDD) Interface Facility	92700-92799
diameter-svc	Diameter Service Facility	121200-121999
diamproxy	Diameter Proxy Facility	119000-119999
dpath	Data Path for IPSec Facility	54000-54999

Facility	Description	Event ID Range
drvctrl	Driver Controller Facility	39000-39999
ds3mgr	DS3 and DS3/E Line Card Manager Facility (part of NPU Manager Controller Facility)	40000-40999
eap-diameter	Extensible Authentication Protocol (EAP) Diameter Facility	92870-92879
eap-ipsec	EAP IPSec Facility	118000-118999
ecs-css	ACS Session Manager (ACSMgr) Signalling Interface Facility	97000-97099
edr	Event Data Record (EDR) Facility	80000-80999
egtpc	eGTP-C Facility	141000-141999
egtpmgr	eGTP Manager Facility	143000-143999
egtpu	eGTP-U Facility	142000-142999
epdg	Evolved Packet Data Gateway (ePDG) Facility	178000-178999
evlog	Event Log Facility	2000-2999
famgr	Foreign Agent (FA) Manager Facility	33000-33999
firewall	Firewall Facility	96000-96999
fng	Femto Network Gateway (FNG) Facility	149000-149999
gbrmgr	Gb-Manager Facility	201900-202699
gcdr	GGSN-Charging Data Record (G-CDR) Facility	66000-66999
gmm	GPRS Mobility Management (GMM) Facility	88100-88299
gprs-app	General Packet Radio Service (GPRS) Application Facility	115100-115399
gprs-ns	GPRS-NS Protocol Facility	115000-115049
gq-rx-tx-diameter	Gq/Rx/Tx Diameter Messages Facility	92830-92839
gss-gcdr	GTPP Storage Server GCDR Facility	98000-98099
gtpc	GTPC Protocol Facility	47000-47999
gtpcmgr	GTPC Signaling Demultiplexer Manager Facility	46000-46999
gtpp	GTP-PRIME Protocol Facility	52000-52999

Facility	Description	Event ID Range
gtpu	GTPU Protocol Facility	45000-45999
gtpumgr	GTPU Manager Facility	157200-157999
gx-ty-diameter	Gx/Ty Diameter Messages Facility	92820-92829
gy-diameter	Gy Diameter Messages Facility	92810-92819
h248prt	H.248 Protocol Facility	42000-42999
hamgr	Home Agent (HA) Manager Facility	34000-34999
hat	High Availability Task (HAT) Facility	3000-3999
hdctrl	Hard Disk (HD) Controller Facility	132000-132999
hddshare	HDD Share Facility	184000-184999
henb-gw	Home eNodeB-GW Facility	195000-195999
henbapp	Home eNodeB Application Facility	196000-196999
henbgwdemux	Home eNodeB-GW Demux Facility	194000-194999
henbgwmgr	Home eNodeB-GW Manager Facility	193000, 193999
hnb-gw	Home NodeB (HNB) Gateway Facility	151000-151999
hnbmgr	HNB Manager Facility	158000-158199
hss-peer-service	Home Subscriber Server (HSS) Facility [MME]	138000-138999
igmp	Internet Group Management Protocol (IGMP) Facility	113000-113999
ikev2	IKEv2 Facility	122000-122999
ims-authorizatn	IMS Authorization Service Library Facility	98100-98999
ims-sh	IMS SH Library Facility	124000-124999
imsimgr	International Mobile Subscriber Identity (IMSI) Manager Facility	114000-114999
imsue	IMS User Equipment (IMSUE) Facility	144000-145999
ip-arp	IP Address Resolution Protocol (ARP) Facility	19000-19999
ip-interface	IP Interface Facility	18000-18999
ip-route	IP Route Facility	20000-20999

Facility	Description	Event ID Range
ipms	Intelligent Packet Monitoring System (IPMS) Facility	134000-134999
ipne	IP Network Enabler (IPNE) Facility	192000-192999
ipsec	IPSec Protocol Facility	55000-56998
ipsg	IP Services Gateway (IPSG) Facility	128000-128999
ipsgmgr	IPSG Manager (IPSGMgr) Facility	99000-99999
ipsp	IP Pool Sharing Protocol (IPSP) Facility	68000-68999
kvstore	Key/Value Store (KVSTORE) Facility	125000-125999
l2tp-control	L2TP Control PDU Protocol Facility	50000-50999
l2tp-data	L2TP Data PDU Protocol Facility	49000-49999
l2tpdemux	L2TP Demux Facility	63000-63999
l2tpmgr	L2TP Manager Facility	48000-48999
lagmgr	Link Aggregation Group (LAG) Manager Facility	179000-179999
ldap	Lightweight Directory Access Protocol (LDAP) Request Facility	160000-160499
li	Lawful Intercept (LI) Log Facility	69000-69999
linkmgr	Link Manager Facility	89500-89999
llc	Logical Link-Control (LLC) Layer Facility (GPRS)	115700-115799
local-policy	Local Policy Configuration Facility	161400-162399
m3ap	M3 Application Protocol (M3AP) Facility	211500-211999
m3ua	MTP Level 3 (M3UA) Protocol Facility [SIGTRAN]	87500-87699
magmgr	Mobile Access Gateway (MAG) Manager Facility	137500-137999
map	Mobile Application Part (MAP) Protocol Facility [SS7]	87100-87299
megadiammgr	MegaDiameter Manager Facility	121000-121199
mme-app	Mobility Management Entity (MME) Application Facility	147000-147999

Facility	Description	Event ID Range
mme-embms	MME evolved Multimedia Broadcast Multicast Service (eMBMS) Facility	212000-212499
mme-misc	MME Miscellaneous Facility	155800-156199
mmedemux	MME Demux Manager Facility	154000-154999
mmemgr	MME Manager Facility	137000-137499
mmgr	Master Manager (MMGR) Facility	86000-86399
mobile-ip	Mobile IP (MIP) Protocol Facility	26000-26999
mobile-ip-data	MIP Tunneled Data Facility	27000-27999
mobile-ipv6	Mobile IPv6 Facility	129000-129999
mpls	Multiprotocol Label Switching (MPLS) Facility	163500-163999
mseg-app	Mobile Services Edge Gateway (MSEG) Application Facility Not supported in this release.	172300-172999
mseg-gtpc	MSEG GTPC Application Facility Not supported in this release.	172000-172199
mseg-gtpu	MSEG GTPU Application Facility Not supported in this release.	172200-172299
msegmgr	MSEG Manager Facility Not supported in this release.	171000-171999
mtp2	Message Transfer Part 2 (MTP2) Service Facility [SS7]	116900-116999
mtp3	Message Transfer Part 3 (MTP3) Service Facility [SS7]	115600-115699
multicast-proxy	Multicast Proxy Facility	94000-94999
nas	Network Access Signaling (NAS) Facility	153000-153999
netwstrg	Network Storage Facility	78000-78999
npuctrl	Network Processing Unit (NPU) Control Facility	16000-16999
npudrv	NPU Driver Facility	191000-191999
npumgr	NPU Manager (NPUMGR) Facility	17000-17999
npumgr-acl	NPUMGR ACL Facility	169000-169999
npumgr-drv	NPUMGR Driver Facility	185000-185999

Facility	Description	Event ID Range
npumgr-flow	NPUMGR Flow Facility	167000-167999
npumgr-fwd	NPUMGR Forwarding Facility	168000-168999
npumgr-init	NPUMGR Initialization Facility	164000-164999
npumgr-lc	NPUMGR LC Facility	180000-180999
npumgr-port	NPUMGR Port Facility	166000-166999
npumgr-recovery	NPUMGR Recovery Facility	165000-165999
npumgr-vpn	NPUMGR VPN Facility	181000-181999
npusim	NPUSIM Facility	176000-176999
ntfy-intf	Event Notification Interface Facility	170000-170499
orbs	Object Request Broker (ORB) System Facility	15000-15999
ospf	Open Shortest Path First (OSPF) Protocol Facility	38000-38999
ospfv3	OSPFv3 Protocol Facility [IPv6]	150000-150999
p2p	Peer-to-Peer (P2P) Facility	146000-146999
pccmgr	Policy Charging and Control (PCC) Manager Facility	159000-159499
pdg	Packet Data Gateway (PDG) Facility	152010-152999
pdgdmgr	PDG TCP Demux Manager (pdgdmgr) Facility (this is a customer-specific facility)	162400-162999
pdif	Packet Data Interworking Function (PDIF) Facility	120000-120999
pgw	Packet Data Network Gateway (PGW) Facility	139000-139999
pmm-app	Packet Mobility Management (PMM) Application Facility [SGSN]	89200-89499
ppp	Point-To-Point Protocol (PPP) Facility	25000-25999
pppoe	Point-to-Point Protocol over Ethernet (PPPoE) Facility	183000-183999
ptt	PTT Facility	76000-76999
push	PUSH (VPNMgr CDR Push) Facility	133000-133999

Facility	Description	Event ID Range
radius-acct	RADIUS Accounting Protocol Facility	24000-24999
radius-auth	RADIUS Authentication Protocol Facility	23000-23999
radius-coa	RADIUS Change of Authorization (CoA) and Disconnect Facility	70000-70999
ranap	Radio Access Network Application Part (RANAP) Facility	87700-87899
rct	Recovery Control Task (RCT) Facility	13000-13999
rdt	Redirector Task (RDT) Facility	67000-67999
resmgr	Resource Manager (RM) Facility	14000-14999
rf-diameter	Rf Diameter Messages Facility	92860-92869
rip	Routing Information Protocol (RIP) Facility	35000-35999
rohc	Robust Header Compression (ROHC) Protocol Facility	103000-103999
rsvp	RSVP Protocol Facility	93000-93999
rua	RANAP User Adaptation (RUA) Protocol Facility	152000-152009
s1ap	S1 Application Protocol (S1AP) Facility	155200-155799
saegw	System Architecture Evolution Gateway Facility	191000-191999
sccp	Signalling Connection Control Part (SCCP) Protocol Facility [SS7]	86700-86899
sct	Shared Configuration Task (SCT) Facility	32000-32099
sctp	Stream Control Transmission Protocol (SCTP) Protocol Facility	87300-87499
sess-gr	SESS-GR Facility	77600-77999
sessctrl	Session Controller Facility	8000-8999
sessmgr	Session Manager Facility	10000-12999
sesstrc	Session Trace Facility	155000-155199
sft	Switch Fabric Task (SFT) Facility	58000-58999
sgs	SGs Interface Protocol Facility [MME]	173000-173199

Facility	Description	Event ID Range
sgsn-app	SGSN Application Interface Facility	115900-115999
sgsn-failures	SGSN Call Failures Facility	89100-89199
sgsn-gtpc	SGSN GTP-C Protocol Facility	116000-116599
sgsn-gtpu	SGSN GTP-U Protocol Facility	86900-87099
sgsn-mbms-bearer	SGSN MBMS Bearer Application (SMGR) Facility	116600-116799
sgsn-misc	SGSN Miscellaneous Facility	88800-89099
sgsn-system	SGSN System Components Facility	86400-86499
sgsn-test	SGSN Tests Facility	88700-88799
sgsn2	SGSN2 Facility	114000-117999
sgtpcmgr	SGSN GTP-C (SGTPC) Manager Facility	117000-117999
sgw	Serving Gateway (SGW) Facility	140000-140999
sh-diameter	Sh Diameter Messages Facility	92850-92859
sipcdprt	SIPCDPRT Facility	95000-95999
sitmain	System Initiation Task (SIT) Main Facility	4000-4999
sm-app	Short Message Service (SMS) Facility	88300-88499
sms	SMS Service Facility	116800-116899
sndcp	Sub Network Dependent Convergence Protocol (SNDCP) Facility	115800-115899
snmp	Simple Network Management Protocol (SNMP) Facility	22000-22999
sprmgr	Subscriber Policy Register (SPR) Manager Facility	159500-159999
srdb	Static Rating Database Facility	102000-102999
srp	Service Redundancy Protocol (SRP) Facility	84000-84999
sscfnni	SSCFNNI Protocol Facility [ATM]	115500-115599
sscop	SSCOP Protocol Facility [ATM]	115400-115499
ssh-ipsec	SSH IP Security Facility	56999-56999
ssl	SSL Facility (this is a customer-specific facility)	156200-157199

Facility	Description	Event ID Range
stat	Statistics Facility	31000-31999
system	System Facility	1000-1999
tacacs+	TACACS+ Protocol Facility	37000-37999
taclcp	TACLCP Facility	44000-44999
tcap	Transaction Capabilities Application Part (TCAP) Protocol Logging Facility [SS7]	86500-86699
testctrl	Test Controller Facility	174000-174999
testmgr	Test Manager Facility	175000-175999
threshold	Threshold Facility	61000-61999
ttg	Tunnel Termination Gateway (TTG) Facility	130000-130999
tucl	TCP/UDP Convergence Layer (TUCL) Facility [SS7]	88500-88699
udr	User Data Record (UDR) Facility	79000-79999
user-data	User-Data Facility	51000-51999
user-l3tunnel	User L3 Tunnel Facility	75000-75999
usertcp-stack	User TCP Stack Facility	173300-173499
vim	Voice Instant Message (VIM) Facility	60000, 60999
vinfo	VINFO Facility	82000, 82999
vmgctrl	Virtual Media Gateway (VMG) Controller Facility	41000, 41999
vmgctxmgr	VMG Context Manager Facility	43000, 43999
vpn	Virtual Private Network (VPN) Facility	5000-5999
wimax-data	WiMAX DATA Facility	104900-104999
wimax-r6	WiMAX R6 Protocol (Signaling) Facility	104000-104899

Event Severities

The system provides the flexibility to configure the level of information that is displayed when logging is enabled. The following levels are supported:

- **critical:** Logs only those events indicating a serious error has occurred that is causing the system tor a system component to cease functioning. This is the highest severity level.

- **error:** Logs events that indicate an error has occurred that is causing the system or a system component to operate in a degraded state. This level also logs events with a higher severity level.
- **warning:** Logs events that may indicate a potential problem. This level also logs events with a higher severity level.
- **unusual:** Logs events that are very unusual and may need to be investigated. This level also logs events with a higher severity level.
- **info:** Logs informational events and events with a higher severity level.
- **trace:** Logs events useful for tracing and events with a higher severity level.
- **debug:** Logs all events regardless of the severity.

Each of the above levels correspond to the "severity" level of the event ID. Therefore, only those event IDs with a "severity" level equal to the logging level are displayed.

Understanding Event ID Information in Logged Output

This section explains the event information that is displayed when logging is enabled.

The following displays a sample output for an event that was logged.

```
2011-Dec-11+5:18:41.993 [cli 30005 info] [8/0/609 cli:8000609 _commands_cli.c:1290] [software
 internal system] CLI session ended for Security Administrator admin on device /dev/pts/2
```

The following table describes the elements of contained in the sample output.

Table 16: Event Element Descriptions

Element	Description
2011-Dec-11+5:18:41.993	Date/Timestamp indicating when the event was generated
[cli 30005 info]	Information about the event including: • The facility the event belongs to • The event ID • The event's severity level In this example, the event belongs to the CLI facility, has an ID of 3005, and a severity level of "info".
[8/0/609 cli:8000609 _commands_cli.c:1290]	Information about the specific CLI instance.
[software internal system]	Indicates that the event was generated because of system operation.
CLI session ended for Security Administrator admin on device /dev/pts/2	The event's details. Event details may, or may not include variables that are specific to the occurrence of the event.

CHAPTER 18

Troubleshooting

This chapter provides information and instructions for using the system command line interface (CLI) for troubleshooting any issues that may arise during system operation.

Verifying Network Connectivity

There are multiple commands supported by the system to verify and/or troubleshoot network connectivity. Note that network connectivity can only be tested once system interfaces and ports have been configured and bound.

The commands specified in this section should be issued on a context-by-context basis. Contexts act like virtual private networks (VPNs) that operate independently of other contexts. Ports, interfaces, and routes configured in one context cannot be tested from another context without additional configuration.

To switch between contexts enter the following command at the root prompt for the Exec mode:

```
[local]host_name# context context_name
```

context_name is the name of the context to which you wish to switch. The following prompt appears:

```
[context_name]host_name#
```

Using the ping or ping6 Command

The **ping** or **ping6** command verifies the system's ability to communicate with a remote node in the network by passing data packets between and measuring the response. This command is useful in verifying network routing and if a remote node is able to respond at the IP layer.

For VPC-DI deployments, use the Exec mode command **system ping** to verify the internal network (DI-network) connectivity between the VPC-DI VMs. Refer to the *Command Line Interface Reference* for more details.

Syntax

The **ping** command has the following syntax:

ping *host_ipv4_address* **[count** *num_packets* **] [flood] [pattern** *packet_pattern* **] [size** *octet_count* **] [src {** *src_host_name* **|** *src_host_ipv4_address* **}] [vrf** *vrf_nam* **]**

ping6 *host_ipv6_address* **[count** *num_packets* **] [flood][pattern** *packet_pattern* **] [size** *octet_count* **] [src {** *src_host_name* **|** *src_host_ipv6_address* **}] [vrf** *vrf_nam* **]**

For complete information on the above commands, see the *Exec Mode Commands* chapter of the *Command Line Interface Reference*.

The following displays a sample of a successful **ping** (IPV4) response.

```
PING 192.168.250.1 (192.168.250.1): 56 data bytes
64 bytes from 192.168.250.1: icmp_seq=0 ttl=255 time=0.4 ms
64 bytes from 192.168.250.1: icmp_seq=1 ttl=255 time=0.2 ms
64 bytes from 192.168.250.1: icmp_seq=2 ttl=255 time=0.2 ms
64 bytes from 192.168.250.1: icmp_seq=3 ttl=255 time=0.2 ms
64 bytes from 192.168.250.1: icmp_seq=4 ttl=255 time=0.2 ms
--- 192.168.250.1 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 0.2/0.2/0.4 ms
```

Troubleshooting

If no response is received from the target follow these troubleshooting procedures:

- Verify that the correct IP address was entered.
- Attempt to ping a different device on the same network. If the ping was successful then it is likely that your system configuration is correct. Verify that the device you are attempting to ping is powered and functioning properly.
- Verify the port is operational.
- Verify that the configuration of the ports and interfaces within the context are correct.
- If the configuration is correct and you have access to the device that you're attempting to ping, ping the system from that device.
- If there is still no response, it is likely that the packets are getting discarded by a network device. Use the **traceroute** or **traceroute6** and **show ip static-route** commands discussed in this chapter to further troubleshoot the issue.

Using the traceroute or traceroute6 Command

The **traceroute** or **traceroute6** command collects information on the route data will take to a specified host. This is a useful troubleshooting command that can be used to identify the source of significant packet delays or packet loss on the network. This command can also be used to identify bottle necks in the routing of data over the network.

traceroute – IPv4

The **traceroute** command has the following syntax:

traceroute { *host_name* | *host_ipv4_address* } [**count** *packets*] [**df**] [**maxttl** *max_ttl*] [**minttl** *min_ttl*] [**port** *port_number*] [**size** *octet_count*] [**src** { *src_host_name* | *src_host_ipv4_address* }] [**timeout** *seconds*] [**vrf** *vrf_nam*]

For complete information on the above command, see the *Exec Mode Commands* chapter of the *Command Line Interface Reference*.

The following displays a sample output.

```
traceroute to 192.168.250.1 (192.168.250.1), 30 hops max, 40 byte packets
 1  192.168.250.1 (192.168.250.1)  0.446 ms  0.235 ms  0.178 ms
```

traceroute6 – IPv6

The **traceroute6** command has the following syntax:

traceroute6 { *host_name* | *host_ipv6_address* } [**count** *packets*] [**maxttl** *max_ttl*] [**port** *port_number*] [**size** *octet_count*] [**src** { *src_host_name* | *src_host_ipv6_address* }] [**timeout** *seconds*] [**vrf** *vrf_nam*]

For complete information on the above commands, see the *Exec Mode Commands* chapter of the *Command Line Interface Reference*.

The following displays a sample output.

```
traceroute6 to 2001:4A2B::1f3F (2001:4A2B::1f3F), 30 hops max, 40 byte packets
 1  2001:4A2B::1f3F (2001:4A2B::1f3F)  0.446 ms  0.235 ms 0.178 ms
```

Viewing IP Routes

The system provides a mechanism for viewing route information to a specific node or for an entire context. This information can be used to verify network connectivity and to ensure the efficiency of the network connection. The command has the following syntax:

show ip route [*route_ip_address*]
show ipv6 route [*route_ipv6_address*]]

For complete information on the above commands, see the *Exec Mode show Commands* chapter of the *Command Line Interface Reference*.

If no keywords are specified, all IP routes within the context's routing table are displayed.

The following displays a sample of this command's output showing a context IPv4 routing table.

```
"*" indicates the Best or Used route.
  Destination     Nexthop     Protocol    Prec    Cost    Interface
*0.0.0.0/0        10.0.4.1    static      0       0       SPIO1
*10.0.4.0/24      0.0.0.0     kernel      0       0       SPIO1
*10.0.4.0/32      0.0.0.0     kernel      0       0       SPIO1
*10.0.4.3/32      0.0.0.0     kernel      0       0       SPIO1
*10.0.4.255/32    0.0.0.0     kernel      0       0       SPIO1
```

Viewing the Address Resolution Protocol Table

The system provides a mechanism for viewing Address Resolution Protocol (ARP) table information to a specific node or for an entire context. This information can be used to verify that when the system sends an ARP packet, it receives valid responses from other network nodes.

```
[local]host_name# show ip arp [ arp_ip_address ]
```

arp_ip_address specifies a specific network node for which to display ARP information. The address can be entered in IPv4 dotted-decimal or IPv6 colon-separated-hexadecimal notation. If this keyword is not specified, all entries within the context's ARP table are displayed.

Important Restarting the VPN Manager removes all interfaces from the kernel which in turn removes all ARP entries. However, the NPU still retains all of the ARP entries so that there is no traffic disruption. From a user point of view, **show ip arp** is broken since this command gathers information from the kernel and not the NPU.

The following displays a sample of this command's output showing a context's ARP table.

```
Flags codes:
C - Completed, M - Permanent, P - Published, ! - Not answered
T - has requested trailers
Address         Link Type    Link Address          Flags    Mask Interface
10.0.4.240      ether        00:05:47:02:20:20     C        MIO1
10.0.4.7        ether        00:05:47:02:03:36     C        MIO1
10.0.4.1        ether        00:01:30:F2:7F:00     C        MIO1
```

Using the System Diagnostic Utilities

The system provides protocol monitor and test utilities that are useful when troubleshooting or verifying configurations. The information generated by these utilities can help identify the root cause of a software or network configuration issue.

This section describes how to use these utilities.

Important Only an administrator with Operator or higher privilege can run the diagnostic utilities described in this section.

Using the Monitor Utility

For troubleshooting purposes, the system provides a protocol monitoring utility. This tool displays protocol information for a particular subscriber session or for every session being processed.

Caution The monitor tool may cause session processing delays and/or data loss. Therefore, it should be used only when troubleshooting.

Using the Protocol Monitor

The protocol monitor displays information for every session that is currently being processed. Depending on the number of protocols monitored, and the number of sessions in progress, a significant amount of data is generated. It is highly recommended that logging be enabled on your terminal client in order to capture all of the information that is generated.

Refer also to Packet Capture (PCAP) Trace, on page 289 to enable PCAP functionality for the **monitor protocol** and **monitor subscriber** commands.

Follow the instructions below to invoke and configure the protocol monitoring tool.

Step 1 Invoke the protocol monitor from the Exec mode by entering the **monitor protocol** command.

```
[local]host_name# monitor protocol
```

An output listing all the currently available protocols, each with an assigned number, is displayed.

Step 2 Choose the protocol that you wish to monitor by entering the associated number at the *Select*: prompt. A right arrow (>) appears next to the protocol you selected.

Step 3 Repeat *step 2* as needed to choose multiple protocols.

Step 4 Press **B** to begin the protocol monitor.

```
WARNING!!! You have selected options that can DISRUPT USER SERVICE
Existing CALLS MAY BE DROPPED and/or new CALLS MAY FAIL!!!
(Under heavy call load, some debugging output may not be displayed)
Proceed? - Select (Y)es or (N)o
```

Step 5 Enter **Y** to proceed with the monitor or **N** to go back to the previous menu.

```
C - Control Events    (ON )
D - Data Events               (ON )
E - EventID Info              (ON )
H - Display ethernet          (ON )
I - Inbound Events            (ON )
O - Outbound Events           (ON )
S - Sender Info               (OFF)
T - Timestamps                (ON )
X - PDU Hexdump               (OFF)
A - PDU Hex/Ascii             (OFF)
+/- Verbosity Level           (    1)
L - Limit Context             (OFF)
M - Match Newcalls            (ON )
R - RADIUS Dict               (no-override)
G - GTPP Dict                 (no-override)
Y - Multi-Call Trace          ((OFF))
      (Q)uit,      <ESC> Prev Menu,      <SPACE> Pause,      <ENTER> Re-Display Options
```

Step 6 Configure the amount of information that is displayed by the monitor. To enable or disable options, enter the letter associated with that option (C, D, E, etc.). To increase or decrease the verbosity, use the plus (+) or minus (-) keys.

The current state, ON (enabled) or OFF (disabled), is shown to the right of each option.

Step 7 Press the **Enter** key to refresh the screen and begin monitoring.

The monitor remains active until disabled. To quit the protocol monitor and return to the prompt, press **q**.

Using the Protocol Monitor for a Specific Subscriber

The protocol monitor can be used to display information for a specific subscriber session that is currently being processed. Depending on the number of protocols monitored, and the number of sessions in progress, a significant amount of data is generated. It is highly recommended that logging be enabled on your terminal client in order to capture all of the information that is generated.

Follow the instructions in this section to invoke and configure the protocol monitoring tool for a specific subscriber session.

Step 1 To invoke the session-specific protocol monitor from the Exec mode enter the **monitor subscriber** command.

```
[local]host_name# monitor subscriber { callid | imei | imsi | ipaddr | ipv6addr |
msid | msisdn | next-call | pcf | peer-fa | peer-lac | sgsn-address | type |
username }
```

Step 2 Specify the method the monitor should use by entering the appropriate keyword.

Step 3 Select other options and/or enter the appropriate information for the selected keyword.

If no session matching the specified criteria was being processed when the monitor was invoked, a screen of available monitoring options appears.

Step 4 Configure the amount of information that is displayed by the monitor. To enable or disable options, enter the letter or 2-digit number associated with that option (C, D, E, 11, 12, etc.). To increase or decrease the verbosity, use the plus (+) or minus (-) keys.

The current state, ON (enabled) or OFF (disabled), is shown to the right of each option.

Option **Y** for performing multi-call traces is only supported for use with the GGSN.

Step 5 Repeat *step 6* as needed to enable or disable multiple protocols.

Step 6 Press **Enter** to refresh the screen and begin monitoring.

The following displays a portion of a sample of the monitor's output for a subscriber named *user2@aaa*. The default protocols were monitored.

```
-----------------------------------------------------------------------------
Incoming Call:
-----------------------------------------------------------------------------
    MSID: 0000012345 Callid: 002dc6c2
    Username: user2@aaa SessionType: unknown
    Status: Active Service Name: xxx1
    Src Context: source Dest Context:
-----------------------------------------------------------------------------

<<<<OUTBOUND 10:02:35:415 Eventid:25001(0)
PPP Tx PDU (9)
PAP 9: Auth-Ack(1), Msg=

<<<<OUTBOUND 10:02:35:416 Eventid:25001(0)
PPP Tx PDU (14)
IPCP 14: Conf-Req(1), IP-Addr=192.168.250.70

<<<<OUTBOUND 10:02:35:416 Eventid:25001(0)
PPP Tx PDU (27)
CCP 27: Conf-Req(1), MPPC, Stac-LZS, Deflate, MVRCA

INBOUND>>>>> 10:02:35:517 Eventid:25000(0)
PPP Rx PDU (30)
```

```
IPCP 30: Conf-Req(1), IP-Comp VJ-Comp, IP-Addr=0.0.0.0, Pri-DNS=0.0.0.0,
Sec-DNS=0.0.0.0

<<<<OUTBOUND 10:02:35:517 Eventid:25001(0)
PPP Tx PDU (26)
IPCP 26: Conf-Rej(1), IP-Comp VJ-Comp, Pri-DNS=0.0.0.0, Sec-DNS=0.0.0.0

INBOUND>>>>> 10:02:35:517 Eventid:25000(0)
PPP Rx PDU (12)
IPCP 12: Conf-Ack(1), IP-Addr=192.168.250.70

INBOUND>>>>> 10:02:35:518 Eventid:25000(0)
PPP Rx PDU (31)
LCP 31: Prot-Rej(1), Rejected-Protocol=CCP (0x80fd)

INBOUND>>>>> 10:02:35:518 Eventid:25000(0)
PPP Rx PDU (12)
IPCP 12: Conf-Req(2), IP-Addr=0.0.0.0

<<<<OUTBOUND 10:02:35:518 Eventid:25001(0)
PPP Tx PDU (14)
IPCP 14: Conf-Nak(2), IP-Addr=192.168.250.87

INBOUND>>>>> 10:02:35:519 Eventid:25000(0)
PPP Rx PDU (12)
IPCP 12: Conf-Req(3), IP-Addr=192.168.250.87
```

The monitor remains active until disabled. To quit the protocol monitor and return to the prompt, press **q**.

Generating an SSD

An SSD is an instance of the output when the Exec mode **show support details** command is run. It displays a comprehensive list of system information that is useful for troubleshooting purposes. In most cases, the output of this command is requested by the Technical Assistance Center (TAC).

An SSD output .tar file can redirected to a local or remote location (URL).

The .tar file includes:

- **support_summary** - An ASCII text file that contains the support detail information.
- **information.minicores.tar** - A .tar file that contains any minicore files found on the system. Minicore files contain memory core dumps that are captured during some events. These core dumps provide specific memory locations and other information about the event. This information is useful to the technical support team in identifying where and when an event occurred along with its probably cause.

The **show support details** command includes information that is not otherwise accessible to users but that is helpful in the swift resolution of issues by TAC.

Important Platforms with large configuration files can take up to 30 minutes to complete an SSD. Executing the **show support details** command consumes system resources and may reduce traffic throughput.

For releases prior to 20.0, an operator could initiate another SSD while an SSD was already running. In a large configuration, the SSD request often timed out while waiting for the first one to complete. The operator

was not aware of the failure until the entire timeout expired. An operator could have more than one SSD running simultaneously.

For release 20.0 and higher, if an SSD is in progress when the operator enters the **show support details** command, StarOS responds with a warning message stating that an SSD is already in progress and the user should try again later. The operator is restricted to running only one SSD instance at a time.

There are optional keywords to the **show support details** command that can target the SSD to only report specific type of information. These keywords can reduce the amount of time required to generate the SSD/

For additional information about the **show support details** command, see the *Exec Mode show Commands (Q-S)* chapter in the *Command Line Interface Reference*.

Configuring and Using the Support Data Collector

The task of collecting the support data is performed by a background CLI task called the record collector. The administrator configures the Support Data Collector (SDC) via the CLI with the commands to be executed on a periodic basis. The record collector always runs in the background and checks if there are records to be collected.

When it is time to collect support data, the scheduler executes the configured sequence of CLI commands and stores the results in a gunzipped (.gz) file on the hard-disk. This file is called an SDR (Support Data Record), and represents a snapshot of the overall state of the system at that time.

Technical Assistance Center (TAC) personnel and local administrators can review the SDRs on-line or by transferring them off the system. They may also wish to investigate the collector state information.

Refer to the *Support Data Collector* chapter for a complete description of SDC functionality.

Hypervisor Forced Reboot

The hypervisor is required to provide a virtual watchdog device. If StarOS stops servicing this watchdog device because of an underlying host OS failure, the hypervisor must forcefully reboot the VM.

In the virtualized chassis that is a VPC-DI instance, there is no way for the CF to remotely force a reboot of another VM resulting from a host failure. Under normal conditions, the CF can remotely reboot another VM by sending a message to StarOS running on that VM. However, under host failure conditions this may not work and the hypervisor watchdog must function as a fall-back mechanism.

Under KVM, a virtual watchdog device can be provided using the "--watchdog i6300esb" command line arguments.

VMware provides a proprietary watchdog mechanism refer to VMware documentation for additional information.

Table 17: Failure Conditions and Reboot Methods

Failure Condition	Reboot Method	Recovery	Notes
card reboot *x*	CF tells remote StarOS to do local reboot.		

Failure Condition	Reboot Method	Recovery	Notes
Critical task failure	Hypervisor watchdog	Hypervisor reboots VM	StarOS HATCPU task stops servicing the watchdog.
Kernel hang/crash	Kernel reboot or Hypervisor watchdog	Hypervisor reboots VM	
Host failure	Hypervisor HA	CF tells remote StarOS to do local reboot	Reference: a VMware HA cluster.

Manually Switching to a Standby CF

You can manually force the active CF "card" to switch to the standby CF in the redundant pair.

The following Exec mode CLI switches the active CF to the standby CF:

`card switch to` *`slot`*

Notes:

- This command only works for redundant CF VMs in a VPC-DI instance.
- You must be logged into the Active CF to initiate the switchover.
- *slot* can be 1 or 2.

The switchover process proceeds as follows:

1. Active CF transitions to Standby mode.
2. Standby CF transitions to Active mode.
3. A reload is initiated on the formerly Active CF (now the Standby CF).

CHAPTER 19

Packet Capture (PCAP) Trace

Feature Information

Summary Data

Applicable Product(s) or Functional Area	• ePDG • IPSec • MME • SaMOG
Applicable Platform(s)	ASR 5500 vPC-SI vPC-DI
Feature Default	Disabled
Related Changes in This Release	Not Applicable
Related Documentation	• *ASR 5000 System Administration Guide* • *ASR 5500 System Administration Guide* • *Command Line Interface Reference Guide* • *ePDG Administration Guide* • *IPSec Reference Guide* • *SaMOG Administration Guide* • *VPC-SI System Administration Guide*

Revision History

Important Revision history details are not provided for features introduced before release 21.2.

Revision Details	Release
PCAP Tracing support for MME S1-AP interface is added in this release.	21.4
First introduced.	21.2

Feature Description

This feature enables the output of the **monitor subscriber** and **monitor protocol** commands to be captured using the packet capture (PCAP) functionality. The output can be stored in a text file in a hard disk, and later transferred to an external server through SFTP using a PUSH or PULL method. The text file can then be converted to a pcap file using external tools such as text2pcap, or imported directly as PCAP using packet analyzer tools such as wireshark.

PCAP trace and hexdump file collection can be enabled or disabled under the **monitor protocol** and **monitor subscriber** commands. For more information, refer *Enabling or Disabling Hexdump* section of this chapter.

Note For VPC-DI deployments, a separate function is available to perform packet captures on specific cards (VMs) and card interfaces on the internal DI-network. Refer to the Exec mode command **system packet-dump** command in the *Command Line Interface Reference* for more information.

Configuring PCAP Trace

Enabling Multiple Instances of CDRMOD

Use the following configuration to enable multiple instances of CDRMOD (one per packet processing card):

```
config
   cdr-multi-mode
   end
```

Notes:

- Although hexdump record generation is supported on both single-mode and multi-mode, it is recommended to enable the CDR multi-mode.

Important After you configure the **cdr-multi-mode** CLI command, you must save the configuration and then reload the chassis for the command to take effect. For information on saving the configuration file and reloading the chassis, refer to the *System Administration Guide* for your deployment.

- Use the **default cdr-multi-mode** command to configure this command with its default setting.

Important After you configure the **default cdr-multi-mode** CLI command, you must save the configuration and then reload the chassis for the command to take effect. For information on saving the configuration file and reloading the chassis, refer to the *System Administration Guide* for your deployment.

- **Default**: Single CDRMOD mode

Configuring the Hexdump Module

Use the following configuration to specify the handling characteristics of the hexdump files:

```
config
   context context_name
      hexdump-module
         hexdump { purge { storage-limit megabytes | time-limit seconds } [
 max-files max_records ] | push-interval interval | push-trigger
space-usage-percent trigger_percent | remove-file-after-transfer |
transfer-mode { pull [ module-only ] | push primary { encrypted-url | url
 } url [ secondary { encrypted-secondary-url | secondary-url } secondary_url
 ] [ via local-context ] [ max-files files ] [ max-tasks max_tasks ] [
module-only ] } | use-harddisk }
         end
```

Notes:

- Use the **default hexdump [purge | push-interval | push-trigger [space-usage-percent] | remove-file-after-transfer | transfer-mode [module-only] | use-harddisk] +** command to configure the keywords to its the default setting.
  - **purge**: Not enabled
  - **push-interval**: 60 seconds
  - **push-trigger**: 80 percent
  - **remove-file-after-transfer**: Disabled
  - **transfer mode**: PUSH
  - **use-harddisk**: Disabled
- Use the **no hexdump [purge | remove-file-after-transfer | use-harddisk] +** command to disable the configured hexdump file storage and processing.

- **purge**: Disables the deleting of record files on the hard disk based on a storage limit or a time limit.
- **remove-file-after-transfer**: Retains a copy of the file even after it has been pushed or pulled to another server.
- **use-harddisk**: Disables data storage on the system's hard disk.

- Use the **purge { storage-limit** *megabytes* **| time-limit** *seconds* **} [max-files** *max_records* **]** keywords to configure parameters for deleting hexdump records from the hard drive. This command is not enabled by default.
  - **storage-limit** *megabytes*: Specifies that hexdump records are to be deleted from the hard drive upon reaching a storage limit defined in megabytes.

    bytes must be an integer from 10 through 143360.
  - **time-limit** *seconds*: Specifies that hexdump records are to be deleted from the hard drive upon reaching a time limit defined in seconds.

    seconds must be an integer from 600 through 2592000.
  - **max-files** *max_records*: Specifies the maximum number of files to purge. If configured to 0, all records will be purged until the limit is reached.

    max_records must be an integer that is of value 0, or from 1000 through 10000.

- Use the **push-interval** *interval*keyword to specify the transfer interval (in seconds) when hexdump files will be pushed to an external file server.
  - *interval* must be an integer from 30 through 3600.
  - **Default**: 60

- Use the **push-trigger space-usage-percent** *trigger_percent* to specify the disk space utilization percentage threshold at which an automatic push is triggered and files are transferred to the external server.
  - *trigger_percent* must be an integer from 10 through 80.
  - **Default**: 80

- Use the **remove-file-after-transfer** keyword to specify that the system must delete hexdump files after they have been transferred to the external file server.

  Default: Disabled.

  Important This keyword must be enabled for hexdump records.

- Use the **transfer-mode { pull [module-only] | push primary { encrypted-url | url }** *url* **[secondary { encrypted-secondary-url | secondary-url }** *secondary_url* **] [via local-context] [max-files** *files* **] [max-tasks** *max_tasks* **] [module-only] }** keywords to specify the transfer mode to be used when transferring hexdump files to an external file server
  - **pull**: Specifies that the destination server (external storage) will pull the hexdump files.
  - **push**: Specifies that the system will push hexdump files to the destination server. This is the default mode.

- **primary encrypted-url** *url*: Specifies the primary URL location to which the system pushes the files in encrypted format.

  url must be an alphanumeric string of 1 through 8192 characters.

- **primary url** *url*: Specifies the primary URL location to which the system pushes the hexdump files.

  url must be an alphanumeric string of 1 through 1024 characters in the format: *//user:password@host:[port]/direct*.

- **secondary encrypted-secondary-url** *secondary_url*: Specifies the secondary URL location to which the system pushes the files in encrypted format.

  secondary_url must be an alphanumeric string of 1 through 8192 characters.

- **secondary secondary-url** *secondary_url*: Specifies the secondary URL location to which the system pushes the hexdump files.

  secondary_url must be an alphanumeric string of 1 through 1024 characters in the format: *//user:password@host:[port]/direct*.

- **via local-context**: Specifies that the local context, and, subsequently, the SPIO management ports, will be used to pull or push hexdump files.

- **max-files** *files*: Specifies the maximum number of files that can be transferred per push.

  files must be an integer from 4 to 4000.

- **max-tasks** *max_tasks*: Specifies the maximum number of files per push.

  max_tasks must be an integer from 4 through 8.

- **module-only**: Specifies that the transfer of hexdump records is to be applied only to the module type for which the configuration was originally created. If this option is not enabled, the transfer will occur for all record types.

- Use the **use-harddisk** keyword to specify that the hard disk drive on the SMC is to be used to store hexdump records.

  Default: Disabled.

Important This keyword must be enabled for hexdump records.

Configuring the Hexdump File Parameters

Use the following configuration to specify the format of the hexdump files:

```
config
   context context_name
      hexdump-module
         file [ compression { gzip | none } | current-prefix prefix |
delete-timeout seconds | directory directory_name | exclude-checksum-record |
 field-separator { hyphen | omit | underscore } | headers | name file_name
 | reset-indicator | rotation { num-records number | tariff-time minute
```

```
minutes hour hours | time seconds | volume bytes } | sequence-number { length
 length | omit | padded | padded-six-length | unpadded } | storage-limit
limit | time-stamp { expanded-format | rotated-format | unix-format } |
trailing-text string | trap-on-file-delete | xor-final-record ] +
          end
```

Notes:

- Use the **default file [compression | current-prefix | delete-timeout | directory | field-separator | headers | name | reset-indicator | rotation { num-records | tariff-time | time | volume } | sequence-number | storage-limit | time-stamp | trailing-text | trap-on-file-delete] +** command to configure the default setting for the specified keyword(s).
- Use the **compression { gzip | none }** keyword to specify the compressions of hexdump files.
  - **gzip**: Enables GNU zip compression of the hexdump file at approximately 10:1 ratio.
  - **none**: Disables Gzip compression.
- Use the **current-prefix** *prefix* keyword to specify a string to add at the beginning of the hexdump file that is currently being used to store records.
  - *prefix* must be an alphanumeric string of 1 through 31 characters.
  - **Default**: curr
- Use the **delete-timeout** *seconds* keyword to specify a time period, in seconds, after which the hexdump files are deleted. By default, files are never deleted.
  - *seconds* must be an integer from 3600 through 31536000.
  - **Default**: Disabled
- Use the **directory** *directory_name* keyword to specify a subdirectory in the default directory in which to store hexdump files.
  - *directory_name* must be an alphanumeric string of 0 through 191 characters.
  - **Default**: */records/hexdump*
- Use the **exclude-checksum-record** keyword to exclude the final record containing #CHECKSUM followed by the 32-bit Cyclic Redundancy Check (CRC) of all preceding records from the hexdump file.

  Default: Disabled (a checksum record is included in the hexdump file header)
- Use the **field-separator { hyphen | omit | underscore }** to specify the type of separators between two fields of a hexdump file name:
  - **hyphen**: Specifies the field separator as a "-" (hyphen) symbol between two fields.
  - **omit**: Omits the field separator between two fields.
  - **underscore**: Specifies the field separator as an "_" (underscore) symbol between two fields.
- Use the **headers** keyword to include a file header summarizing the record layout.
- Use the **name** *file_name* to specify a string to be used as the base file name for hexdump files.

file_name must be an alphanumeric string from 1 through 31 characters.

- Use the **reset-indicator** to specify the inclusion of the reset indicator counter (value from 0 through 255) in the hexdump file name.

  The counter is incremented whenever any of the following conditions occur:

  - A peer chassis has taken over in compliance with Interchassis Session Recovery (ICSR).
  - The sequence number (see **sequence-number** keyword) has rolled over to zero.

- Use the **rotation { num-records** *number* **| tariff-time minute** *minutes* **hour** *hours* **| time** *seconds* **| volume** *bytes* **}** keyword to specify when to close a hexdump file and create a new one.

  - **num-records** *number*: Specifies the maximum number of records that should be added to a hexdump file. When the number of records in the file reaches this value, the file is complete.

    number must be an integer from 100 through 10240. **Default**: 1024

  - **tariff-time minute** *minutes* **hour** *hours*: Specifies to close the current hexdump file and create a new one based on the tariff time (in minutes and hours).

    minutes must be an integer from 0 through 59.

    hours must be an integer from 0 through 23.

  - **time** *seconds*: Specifies the period of time to wait (in seconds) before closing the current hexdump file and creating a new one.

    seconds must be an integer from 30 through 86400. **Default**: 3600

Important It is recommended to set the rotation time to 30 seconds.

  - **volume** *bytes*: Specifies the maximum size of the hexdump file (in bytes) before closing it and creating a new one.

    bytes must be an integer from 51200 through 62914560. Note that a higher setting may improve the compression ratio when the compression keyword is set to gzip. **Default**: 102400

- Use the **sequence-number { length** *length* **| omit | padded | padded-six-length | unpadded }** keyword to exclude or include the sequence number with a specified format in the file name.

  - **length** *length*: Includes the sequence number with the specified length.

    length must be the file sequence number length with preceding zeroes in the file name, and must be an integer from 1 through 9.

  - **omit**: Excludes the sequence number from the file name.
  - **padded**: Includes the padded sequence number with preceding zeros in the file name. This is the default setting.
  - **padded-six-length**: Includes the padded sequence number with six preceding zeros in the file name.
  - **unpadded**: Includes the unpadded sequence number in the file name.

- Use the **storage-limit** *limit* keyword to set the storage limit. Files will be deleted when the specified amount of space (in bytes) is reached.

  limit must be an integer from 10485760 through 268435456.

- Use the **time-stamp { expanded-format | rotated-format | unix-format }** keyword to specify the format of the file creation timestamp to be included in the file name.
  - **expanded-format**: Specifies the UTC (Universal Time Coordinated) MMDDYYYYHHMMSS format.
  - **rotated-format**: Specifies the time stamp format to YYYYMMDDHHMMSS format.
  - **unix-format**: Specifies the UNIX format of x.y, where x is the number of seconds since 1/1/1970 and y is the fractional portion of the current second that has elapsed.

- Use the **trailing-text** *string* keyword to specify the inclusion of an arbitrary text string in the file name as an alphanumeric string of 1 through 30 characters.

  string must be an alphanumeric string from 1 through 30 characters.

- Use the **trap-on-file-delete** keyword to instruct the system to send an SNMP notification (trap) when a hexdump file is deleted due to lack of space.

  Default: Disabled

- Use the **xor-final-record** keyword to insert an exclusive OR (XOR) checksum (instead of a CRC checksum) into the hexdump file header, if the exclude-checksum-record is left at its default setting.

  Default: Disabled

- The + symbol indicates that more than one of the previous keywords can be entered within a single command.

Enabling or Disabling Hexdump

Hexdump captures can be enabled for protocols in the **monitor subscriber** and **monitor protocol** commands in the Exec Mode. Subscriber information for PCAP trace can be specified using the filters in the **monitor subscriber** command. For protocols and filters supported for a specific product, refer the respective product Administration and Reference guides.

When the **monitor subscriber** or **monitor protocol** command is running, use the **U** or **V** option to enable hexdump capturing:

- **U - Mon Display (ON)**: Use this option to display message captures on the terminal.
  - **Default**: ON
  - When this option is turned off, monitoring will still run in the background.

- **V - PCAP Hexdump (NONE)**: Use this option to enable or disable capturing hexdump packets globally.
  - **Default**: None
  - **V - PCAP Hexdump (ON)**: Hexdump capture is enabled with the prompt:

    Warning :Turning ON/OFF will impact other cli logging terminals, You will interupt others already using hexdump.

- **V - PCAP Hexdump (OFF)**: Hexdump capture is disabled (paused).

Enabling PCAP Trace for MME

This section describes how to enable PCAP trace for MME S1-AP interface and SGsAP interface.

- Under monitor protocol (monpro), enable S1-AP and SGS, or SCTP protocol option along with V - PCAP Hexdump (ON), to capture all S1-AP messages in PCAP hexdump.
- Monitor subscriber (monsub) supports PCAP tracing on S1-AP and SGS filter options.
- When S1-AP or SGS filter option is selected in monpro/monsub, PCAP Hexdump will have dummy SCTP header. The following fields are set as dummy in the SCTP header:
  - Verification tag
  - Checksum
  - Chunk flags
  - Transmission Sequence Numbers (TSN)
  - Stream identifier
  - Stream sequence number
- When the SCTP protocol option is selected in monpro, PCAP hexdump will have the original SCTP header.

Monitoring and Troubleshooting PCAP Trace

Show Command(s) and/or Outputs

The show command(s) in this section are available in support of PCAP trace.

show cdr statistics

The following fields are available in the output of the **show cdr statistics** command in support of this feature:

```
EDR-UDR file Statistics:
------------------------
CDRMOD Instance Id: 2
 Hexdump-module Record Specific Statistics:
  Hexdump-module files rotated:                      0
  Hexdump-module files rotated due to volume limit:  0
  Hexdump-module files rotated due to time limit:    0
  Hexdump-module files rotated due to tariff-time:   0
  Hexdump-module files rotated due to records limit: 0
  Hexdump-module file rotation failures:             0
  Hexdump-module files deleted:                      0
  Hexdump-module records deleted:                    0
  Hexdump-module records received:                   0
  Current open Hexdump-module files:                 0
  Time of last Hexdump-module file deletion:         0
```

Table 18: show cdr statistics Command Output Descriptions

Field	Description
EDR-UDR file Statistics:	
CDRMOD Instance Id	Indicates the CDRMOD instance id for which the statistics are collected.
Hexdump-module Record Specific Statistics:	
Hexdump-module files rotated	Total number of times a hexdump file was closed and a new hexdump file was created.
Hexdump-module files rotated due to volume limit	Total number of times a hexdump file was closed and a new hexdump file was created since the volume limit was reached.
Hexdump-module files rotated due to time limit	Total number of times a hexdump file was closed and a new hexdump file was created since the time limit was reached.
Hexdump-module files rotated due to tariff-time	Total number of times a hexdump file was closed and a new hexdump file was created since the tariff time was reached.
Hexdump-module files rotated due to records limit	Total number of times a hexdump file was closed and a new hexdump file was created since the records limit was reached.
Hexdump-module file rotation failures	Total number of times hexdump file rotation failed.
Hexdump-module files deleted	Total number of times hexdump files were deleted.
Hexdump-module records deleted	Total number of times hexdump records were deleted.
Hexdump-module records received	Total number of times hexdump records were received.
Current open Hexdump-module files	Total number of hexdump files currently open.
Time of last Hexdump-module file deletion	Time of the last deleted hexdump file.

show { hexdump-module | cdr } file-space-usage

The following fields are available in the output of the **show { hexdump-module | cdr } file-space-usage** command in support of this feature:

```
CDRMOD Instance Id: 2
   Hexdump-module File Storage LIMIT              : 33554432 bytes
   Hexdump-module File Storage USAGE              : 196608 bytes
   Percentage of Hexdump-module file store usage  : 0.585938
```

*Table 19: **show { hexdump-module | cdr } file-space-usage** Command Output Descriptions*

Field	Description
CDRMOD Instance Id	Indicates the CDRMOD instance id for which the statistics are collected.
Hexdump-module File Storage LIMIT	Indicates the maximum storage space (in bytes) that can be used for hexdump files.
Hexdump-module File Storage USAGE	Indicates the total storage space (in bytes) used for hexdump files.
Percentage of Hexdump-module file store usage	Indicates the total percentage of storage used for hexdump files.

show hexdump-module statistics

The following fields are available in the output of the **show hexdump-module statistics** command in support of this feature.

```
Hexdump-module-Record file Statistics:
------------------------
CDRMOD Instance Id: 2
  Hexdump-module files rotated:                      0
  Hexdump-module files rotated due to volume limit:  0
  Hexdump-module files rotated due to time limit:    0
  Hexdump-module files rotated due to tariff-time:   0
  Hexdump-module files rotated due to records limit: 0
  Hexdump-module file rotation failures:             0
  Hexdump-module files deleted:                      0
  Hexdump-module records deleted:                    0
  Hexdump-module records received:                   0
  Current open Hexdump-module files:                 0
  Time of last Hexdump-module file deletion:         0

Hexdump-module PUSH Statistics:
-----------------------------------
  Successful File Transfers   : 0
  Failed File Transfers       : 0
  Num of times PUSH initiated : 0
  Num of times PUSH Failed    : 0
  Num of times PUSH cancelled
            due to HD failure : 0
  Num of periodic PUSH        : 0
  Num of manual PUSH          : 0
  Current status of PUSH      : Not Running
  Last completed PUSH time    : N/A

Primary Server Statistics:
  Successful File Transfers   : 0
  Failed File Transfers       : 0
  Num of times PUSH initiated : 0
  Num of times PUSH Failed    : 0
  Num of periodic PUSH        : 0
  Num of manual PUSH          : 0
  Current status of PUSH      : Not Running
  Last completed PUSH time    : N/A

Secondary Server Statistics:
```

```
Successful File Transfers   : 0
Failed File Transfers       : 0
Num of times PUSH initiated : 0
Num of times PUSH Failed    : 0
Num of periodic PUSH        : 0
Num of manual PUSH          : 0
Current status of PUSH      : Not Running
Last completed PUSH time    : N/A
```

Important Use the **clear hexdump-module statistics** command under the Exec Mode to clear and reset the hexdump module statistics.

*Table 20: **show hexdump-module statistics** Command Output Descriptions*

Field	Description
Hexdump-module-Record file Statistics:	
CDRMOD Instance Id	Indicates the CDRMOD instance id for which the statistics are collected.
Hexdump-module files rotated	Total number of times a hexdump file was closed and a new hexdump file was created.
Hexdump-module files rotated due to volume limit	Total number of times a hexdump file was closed and a new hexdump file was created since the volume limit was reached.
Hexdump-module files rotated due to time limit	Total number of times a hexdump file was closed and a new hexdump file was created since the time limit was reached.
Hexdump-module files rotated due to tariff-time	Total number of times a hexdump file was closed and a new hexdump file was created since the tariff time was reached.
Hexdump-module files rotated due to records limit	Total number of times a hexdump file was closed and a new hexdump file was created since the records limit was reached.
Hexdump-module file rotation failures	Total number of times hexdump file rotation failed.
Hexdump-module files deleted	Total number of times hexdump files were deleted.
Hexdump-module records deleted	Total number of times hexdump records were deleted.
Hexdump-module records received	Total number of times hexdump records were received.
Current open Hexdump-module files	Total number of hexdump files currently open.
Time of last Hexdump-module file deletion	Time of the last deleted hexdump file.
Hexdump-module PUSH Statistics:	
Successful File Transfers	Total number of hexdump files that were successfully transferred.
Failed File Transfers	Total number of hexdump files that failed to transfer.

Field	Description
Num of times PUSH initiated	Total number of times the PUSH operation was initiated.
Num of times PUSH Failed	Total number of times PUSH operation failed.
Num of times PUSH cancelled due to HD failure	Total number of times PUSH operation failed due to hard disk failure.
Num of periodic PUSH	Total number of periodic times PUSH operation was performed.
Num of manual PUSH	Total number of times the PUSH operation was performed manually.
Current status of PUSH	Indicates if the PUSH operation is currently running.
Last completed PUSH time	Indicates the time when the last PUSH operation was completed.
Primary Server Statistics:	
Successful File Transfers	Total number of hexdump files successfully transferred to the primary storage server.
Failed File Transfers	Total number of hexdump files that failed transfer to the primary storage server.
Num of times PUSH initiated	Total number of times PUSH operation was initiated to transfer hexdump files to the primary storage server.
Num of times PUSH Failed	Total number of times PUSH operation failed to transfer hexdump files to the primary storage server.
Num of periodic PUSH	Total number of periodic times PUSH operation was performed to the primary storage server.
Num of manual PUSH	Total number of times the PUSH operation to the primary storage server was performed manually.
Current status of PUSH	Indicates if the PUSH operation to the primary storage server is currently running.
Last completed PUSH time	Indicates the time when the last PUSH operation to the primary storage server was completed.
Secondary Server Statistics:	
Successful File Transfers	Total number of hexdump files successfully transferred to the secondary storage server.
Failed File Transfers	Total number of hexdump files that failed transfer to the secondary storage server.
Num of times PUSH initiated	Total number of times PUSH operation was initiated to transfer hexdump files to the secondary storage server.

Field	Description
Num of times PUSH Failed	Total number of times PUSH operation failed to transfer hexdump files to the secondary storage server.
Num of periodic PUSH	Total number of periodic times PUSH operation was performed to the secondary storage server.
Num of manual PUSH	Total number of times the PUSH operation to the secondary storage server was performed manually.
Current status of PUSH	Indicates if the PUSH operation to the secondary storage server is currently running.
Last completed PUSH time	Indicates the time when the last PUSH operation to the secondary storage server was completed.

CHAPTER 20

System Recovery

This chapter describes how to recover a system after it has failed to complete a reboot following a power off cycle or interruption of the normal boot sequence following a **reload** command.

Caution This system recovery process interrupts subscriber service by dropping any existing flows and preventing traffic from being processed during the boot interval. It should only be initiated as an emergency measure.

This chapter includes the following sections:

Prerequisites

Recovery from a failed reboot requires that you have access to the VPC-SI or VPC-DI CF VM via a hypervisor console, and have an uncorrupted copy of the StarOS .bin and .iso image files accessible to the hypervisor.

Console Access

The boot recovery sequence can only be executed via the hypervisor console.

Boot Image

The SYSLINUX bootloader allows you to specify the priority of the boot image from which you would like to boot the system. If a VPC VM failed to reload following a software update, you can initiate a boot from a previously stored image.

The system recovery process will prompt you to enter the path name for the location of the StarOS boot image from which the system will boot. By default the boot command will timeout and attempt to reload the highest priority image from flash memory using the default configuration file.

The StarOS software is delivered as a single binary file (**.bin** file extension) and is loaded as a single instance for the entire system.

- *For StarOS releases prior to 16.1*, the image filename is identified by its release version and corresponding build number. Format = **production.***build_number***.***platform***.bin**.
- *For StarOS release 16.1 onwards*, the image filename is identified by its platform type and release number. Format = *platform-release_number***.bin**.

Multiple boot priorities are provided, each of which consist of a boot image (.bin) and configuration file. The lowest boot priority will be automatically booted on each boot. However, on startup a different priority can be manually booted by entering its number at the SYSLINUX "boot:" prompt.

Note VPC VMs do **not** support booting from the network; they can only be booted from the local vHDD.

Refer to the *Configuring the Boot Stack* section in the *Software Management Operations* chapter for additional information on boot stack entries and prioritization.

Accessing the boot CLI

To access the boot CLI you must interrupt an in-progress reload (reboot) sequence.

Caution This system recovery process interrupts subscriber service by dropping any existing flows and preventing traffic from being processed during the boot interval. It should only be initiated as an emergency measure.

Initiate a Reboot

A reload is initiated by restarting the VM via the hypervisor GUI. This will automatically bring up the SYSLINUX bootloader.

The boot sequence displays messages on the console as it steps through its processes.

At the *boot:* prompt, type the priority number of the desired boot file.

Interrupt the Boot Sequence

When the "Booting priority" message line appears (and not before), press CTRL+C to break out of the boot process as shown in the example below:

```
Booting priority 8
   image : /flash/image_filename.bin
   config: /flash/system.cfg
Entry at 0x000000000cba45e0
```

Press CTRL+C at this point in the sequence.

A message similar to the following appears after the boot process has been interrupted:

```
*******9/0 Ctrl-C Pressed-------------------------------------------------------
Failed.
   aborted by user
8/0:boot>
```

Enter CLI Mode

With the boot prompt displayed, enter **cli** to access the boot recovery CLI. The CLI prompt changes as shown below:

```
8/0:boot>cli
8/0:cli>
```

boot Command Syntax

The boot recovery command has the following syntax:

boot [-show | -priority=* | -config=* | -noconfig] { bootfile_URL }

The options for this command include:

- **-show**: displays the current boot configuration
- **-priority=***: selects the desired boot stack priority (*)
- **-config=***: enters the desired configuration filename (*), if not the default file
- **-noconfig**: boots using no configuration file

bootfile_URL is the URL for the location of the StarOS boot image file. It specifies the path and file name of the StarOS .bin file from which the system will be booted.

The URL may refer to a local file (flash) or an external file on a memory device attached to the management card. The URL must be entered in the following format:

{ /flash | /pcmcia1 | /usb1 }/*filename*

Booting from a Selected Image

You will issue a **boot** command via the boot CLI to initiate the system recovery process.

Boot Using No Configuration FIle

This procedure boots the system using the specified boot image without also loading a configuration file. A sample command string appears below:

8/0:cli>boot -noconfig /flash/*image_filename***.bin**

The boot sequence ends with a prompt to enter the Quick Setup Wizard for creating a configuration file.

```
Launching StarOS
Starting program at 0x0000000000100000
Starent Networks ASR5500 Intelligent Mobile Gateway
management_card is starting up.............................
Starting software image_version_number...
No configuration found, press enter to continue.
1. Do you wish to continue with the Quick Setup Wizard[yes/no]:
```

You can exit the Quick Setup Wizard by entering **no** in response to the above prompt. Load a desired configuration file using the Exec mode **configure** command followed by the URL for the configuration file as shown in the example below:

[local]*host_name*# **configure /flash/system.cfg**

Boot Using A Specified Configuration File

This procedure boots the system using the specified boot image and configuration file. A sample command string appears below:

8/0:cli>boot -config=/flash/system.cfg /flash/*image_filename***.bin**

The boot sequence ends with the appearance of the CLI prompt.

[local]*host_name*#

Confirm that the desired configuration has loaded by running the Exec mode **show configuration** command.

Recovering from a Bad Startup Configuration File

In the event the startup configuration file is corrupt or unusable (such as, an empty configuration file, one with no administrators or invalid passwords), the VPC VM can be recovered as follows:

- Reboot the VM.
- At the SYSLINUX "boot:" prompt type *priority_number* **config=** where *priority_number* is a boot priority with a known good .bin file.

The VM will then boot with that priority's .bin file but with no startup configuration

CHAPTER 21

Access Control Lists

This chapter describes system support for access control lists and explains how they are configured. The product administration guides provide examples and procedures for configuration of basic services on the system. You should select the configuration example that best meets your service model before using the procedures described below.

Important You do not require a license to configure ACLs. However, the number of ACLs configured may impact performance significantly.

Important Not all commands and keywords/variables may be available. Availability depends on the platform type.

This chapter contains the following sections:

Overview

IP access lists, commonly known as access control lists (ACLs), control the flow of packets into and out of the system. They are configured on a per-context basis and consist of "rules" (ACL rules) or filters that control the action taken on packets that match the filter criteria. Once configured, an ACL can be applied to any of the following:

- An individual interface
- All traffic facilitated by a context (known as a policy ACL)
- An individual subscriber
- All subscriber sessions facilitated by a specific context

Separate ACLs may be created for IPv4 and IPv6 access routes.

Understanding ACLs

This section discusses the two main aspects to ACLs on the system:

- Rule(s), on page 308
- Rule Order, on page 309

Important Refer to *ACL Configuration Mode Commands* and the *IPv6 ACL Configuration Mode Commands* chapter in the *Command Line Interface Reference* for the full command syntax.

Rule(s)

A single ACL consists of one or more ACL rules. Each rule is a filter configured to take a specific action when packets matching specific criteria. Up to 256 rules can be configured per ACL.

Important Configured ACLs consisting of no rules imply a "deny any" rule. The **deny** action and **any** criteria are discussed later in this section. This is the default behavior for an empty ACL.

Each rule specifies the action to take when a packet matches the specifies criteria. This section discusses the rule actions and criteria supported by the system.

Actions

ACLs specify that one of the following actions can be taken on a packet that matches the specified criteria:

- **Permit**: The packet is accepted and processed.
- **Deny**: The packet is rejected.
- **Redirect**: The packet is forwarded to the specified next-hop address through a specific system interface or to the specified context for processing.

Important Redirect rules are ignored for ACLs applied to specific subscribers or all subscribers facilitated by a specific context, or APN for UMTS subscribers.

Criteria

Each ACL consists of one or more rules specifying the criteria that packets will be compared against.

The following criteria are supported:

- **Any**: Filters all packets
- **Host**: Filters packets based on the source host IP address
- **ICMP**: Filters Internet Control Message Protocol (ICMP) packets

- **IP**: Filters Internet Protocol (IP) packets
- **Source IP Address**: Filter packets based on one or more source IP addresses
- **TCP**: Filters Transport Control Protocol (TCP) packets
- **UDP**: Filters User Datagram Protocol (UDP) packets

Each of the above criteria are described in detail in the sections that follow.

Important The following sections contain basic ACL rule syntax information. Refer to the *ACL Configuration Mode Commands* and *IPv6 ACL Configuration Mode Commands* chapters in the *Command Line Interface Reference* for the full command syntax.

- **Any**: The rule applies to all packets.
- **Host**: The rule applies to a specific host as determined by its IP address.
- **ICMP**: The rule applies to specific Internet Control Message Protocol (ICMP) packets, Types, or Codes. ICMP type and code definitions can be found at *www.iana.org* (RFC 3232).
- **IP**: The rule applies to specific Internet Protocol (IP) packets or fragments.
- **IP Packet Size Identification Algorithm**: The rule applies to specific Internet Protocol (IP) packets identification for fragmentation during forwarding.

  This configuration is related to the "IP Identification field" assignment algorithm used by the system, when subscriber packets are being encapsulated (such as Mobile IP and other tunneling encapsulation). Within the system, subscriber packet encapsulation is done in a distributed way and a 16-bit IP identification space is divided and distributed to each entity which does the encapsulation, so that unique IP identification value can be assigned for IP headers during encapsulation.

  Since this distributed IP Identification space is small, a non-zero unique identification will be assigned only for those packets which may potentially be fragmented during forwarding (since the IP identification field is only used for reassembly of the fragmented packet). The total size of the IP packet is used to determine the possibility of that packet getting fragmented.
- **Source IP Address**: The rule applies to specific packets originating from a specific source address or a group of source addresses.
- **TCP**: The rule applies to any Transport Control Protocol (TCP) traffic and could be filtered on any combination of source/destination IP addresses, a specific port number, or a group of port numbers. TCP port numbers definitions can be found at *www.iana.org*
- **UDP**: The rule applies to any User Datagram Protocol (UDP) traffic and could be filtered on any combination of source/destination IP addresses, a specific port number, or a group of port numbers. UDP port numbers definitions can be found at *www.iana.org*.

Rule Order

A single ACL can consist of multiple rules. Each packet is compared against each of the ACL rules, in the order in which they were entered, until a match is found. Once a match is identified, all subsequent rules are ignored.

Additional rules can be added to an existing ACL and properly ordered using either of the following options:

- Before
- After

Using these placement options requires the specification of an existing rule in the ACL and the configuration of the new rule as demonstrated by the following flow:

```
[ before | after ] { existing_rule }
```

Configuring ACLs on the System

This section describes how to configure ACLs.

Important This section provides the minimum instruction set for configuring access control list on the system. For more information on commands that configure additional parameters and options, refer to the *ACL Configuration Mode Commands* and *IPv6 ACL Configuration Mode Commands* chapters in the *Command Line Interface Reference*.

To configure the system to provide an access control list facility to subscribers:

Step 1 Create the access control list by following the example configuration in Creating ACLs, on page 310

Step 2 Specify the rules and criteria for action in the ACL list by following the example configuration in Configuring Action and Criteria for Subscriber Traffic, on page 311

Step 3 *Optional*. The system provides an "undefined" ACL that acts as a default filter for all packets into the context. The default action is to "permit all". Modify the default configuration for "unidentified" ACLs for by following the example configuration in Configuring an Undefined ACL, on page 311

Step 4 Verify your ACL configuration by following the steps in Verifying the ACL Configuration, on page 311

Step 5 Save your configuration to flash memory, an external memory device, and/or a network location using the Exec mode **save configuration** command. For additional information refer to the *Verifying and Saving Your Configuration* chapter.

Creating ACLs

To create an ACL, enter the following command sequence from the Exec mode of the system CLI:

```
configure
  context acl_ctxt_name [ -noconfirm ]
    { ip | ipv6 } access-list acl_list_name
    end
```

Notes:

- The maximum number of ACLs that can be configured per context is limited by the amount of available memory in the VPN Manager software task. Typically, the maximum is less than 200.

Configuring Action and Criteria for Subscriber Traffic

To create rules to deny/permit the subscriber traffic and apply the rules after or before action, enter the following command sequence from the Exec mode of the system CLI:

```
configure
  context acl_ctxt_name [ -noconfirm ]
    { ip | ipv6 } access-list acl_list_name
      deny { ip_address | any | host | icmp | ip | log | tcp | udp }
      permit { ip_address | any | host | icmp | ip | log | tcp | udp }
      after { deny | permit | readdress | redirect }
      before { deny | permit | readdress | redirect }
      end
```

Notes:

Caution The system does not apply a "deny any" rule, unless it is specified in the ACL. This behavior can be changed by adding a "deny any" rule at the end of the ACL.

- The maximum number of rules that can be configured per ACL varies depending on how the ACL is to be used. For more information, refer to the *Engineering Rules* chapter.
- Use the information provided in the Actions and Criteria to configure the rules that comprise the ACL. For more information, refer to the *ACL Configuration Mode Commands* and *IPv6 ACL Configuration Mode Commands* chapters in the *Command Line Interface Reference*.

Configuring an Undefined ACL

As discussed previously the system uses an "undefined" ACL mechanism for filtering the packet(s) in the event that an ACL that has been applied is not present. This scenario is likely the result of a mis-configuration such as the ACL name being mis-typed during the configuration process.

For these scenarios, the system provides an "undefined" ACL that acts as a default filter for all packets into the context. The default action is to "permit all".

To modify the default behavior for unidentified ACLs, use the following configuration:

```
configure
  context acl_ctxt_name [-noconfirm]
    access-list undefined { deny-all | permit-all }
    end
```

Notes:

- Context name is the name of the context containing the "undefined" ACL to be modified. For more information, refer to the *Context Configuration Mode Commands* chapter in the *Command Line Interface Reference*.

Verifying the ACL Configuration

To verify the ACL configuration, enter the Exec mode **show { ip | ipv6 } access-list** command.

The following is a sample output of this command. In this example, an ACL named *acl_1* was configured.

```
ip access list acl_1
   deny host 10.2.3.4
   deny ip any host 10.2.3.4
   permit any 10.2.4.4
1 ip access-lists are configured.
```

Applying IP ACLs

Once an ACL is configured, it must be applied to take effect.

Important All ACLs should be configured and verified according to the instructions in the Configuring ACLs on the System, on page 310 prior to beginning these procedures. The procedures described below also assume that the subscribers have been previously configured.

As discussed earlier, you can apply an ACL to any of the following:

- Applying an ACL to an Individual Interface, on page 314
- Applying an ACL to All Traffic Within a Context, on page 315 (known as a policy ACL)
- Applying an ACL to an Individual Subscriber, on page 317
- Applying a Single ACL to Multiple Subscribers, on page 321
- Applying a Single ACL to Multiple Subscribers, on page 321 (for 3GPP subscribers only)

Important ACLs must be configured in the same context in which the subscribers and/or interfaces to which they are to be applied. Similarly, ACLs to be applied to a context must be configured in that context.

If ACLs are applied at multiple levels within a single context (such as an ACL is applied to an interface within the context and another ACL is applied to the entire context), they will be processed as shown in the following figure and table.

Figure 6: ACL Processing Order

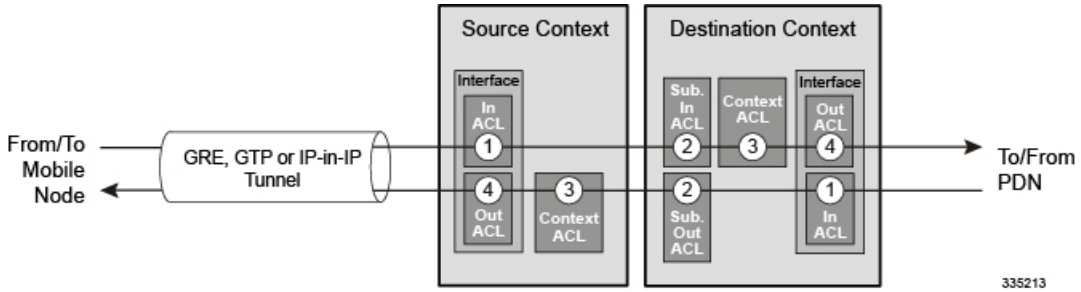


Table 21: ACL Processing Order Descriptions

Packet coming from the mobile node to the packet data network (left to right)	
Order	Description

1	An inbound ACL configured for the receiving interface in the Source Context is applied to the tunneled data (such as the outer IP header). The packet is then forwarded to the Destination Context.
2	An inbound ACL configured for the subscriber (either the specific subscriber or for any subscriber facilitated by the context) is applied.
3	A context ACL (policy ACL) configured in the Destination Context is applied prior to forwarding.
4	An outbound ACL configured on the interface in the Destination Context through which the packet is being forwarded, is applied.
Packet coming from the packet data network to the mobile node (right to left)	
Order	Description
1	An inbound ACL configured for the receiving interface configured in the Destination Context is applied.
2	An outbound ACL configured for the subscriber (either the specific subscriber or for any subscriber facilitated by the context) is applied. The packet is then forwarded to the Source Context.
3	A context ACL (policy ACL) configured in the Source Context is applied prior to forwarding.
4	An outbound ACL configured on the interface in the Source Context through which the packet is being forwarded, is applied to the tunneled data (such as the outer IP header).

In the event that an IP ACL is applied that has not been configured (for example, the name of the applied ACL was configured incorrectly), the system uses an "undefined" ACL mechanism for filtering the packet(s).

This section provides information and instructions for applying ACLs and for configuring an "undefined" ACL.

Applying the ACL to an Interface

To apply the ACL to an interface, use the following configuration:

```
configure
  context acl_ctxt_name [ -noconfirm ]
    interface interface_name
      { ip | ipv6 } access-group acl_list_name { in | out } [ preference ]
      end
```

Notes:

- The context name is the name of the ACL context containing the interface to which the ACL is to be applied.
- The ACL to be applied must be configured in the context specified by this command.
- Up to 16 ACLs can be applied to a group provided that the number of rules configured within the ACL(s) does not exceed the 128-rule limit for the interface.

Applying an ACL to an Individual Interface

This section provides information and instructions for applying one or more ACLs to an individual interface configured on the system.

Important This section provides the minimum instruction set for applying the ACL list to an interface on the system. For more information on commands that configure additional parameters and options, refer to the *Ethernet Interface Configuration Mode Commands* chapter in the *Command Line Interface Reference*.

To configure the system to provide ACL facility to subscribers:

Step 1 Apply the configured access control list by following the example configuration in Applying the ACL to an Interface, on page 313

Step 2 Verify that ACL is applied properly on interface by following the steps in Verifying the ACL Configuration on an Interface, on page 314

Step 3 Save your configuration to flash memory, an external memory device, and/or a network location using the Exec mode **save configuration** command. For additional information refer to the *Verifying and Saving Your Configuration* chapter.

Verifying the ACL Configuration on an Interface

This section describes how to verify the ACL configuration.

In the Exec Mode, enter the following command:

```
[local]host_name# show configuration context context_name
```

context_name is the name of the context containing the interface to which the ACL(s) was/were applied.

The output of this command displays the configuration of the entire context. Examine the output for the commands pertaining to interface configuration. The commands display the ACL(s) applied using this procedure.

```
configure
   context context_name
      ip access-list acl_name
         deny host ip_address
         deny ip any host ip_address
         exit
      ip access-group access_group_name
      service-redundancy-protocol
      exit
      interface interface_name
         ip address ip_address/mask
         exit
```

```
subscriber default
exit
aaa group default
exit
gtpp group default
end
```

Applying the ACL to a Context

To apply the ACLs to a context, use the following configuration:

```
configure
  context acl_ctxt_name [ -noconfirm ]
    { ip | ipv6 } access-group acl_list_name [ in | out ] [ preference ]
    end
```

Notes:

- The context name is the name of the ACL context containing the interface to which the ACL is to be applied.
- The context-level ACL is applied to outgoing packets. This applies to incoming packets also if the flow match criteria fails and forwarded again.

  The **in** and **out** keywords are deprecated and are only present for backward compatibility.

  Context ACL will be applied in the following cases:

  - Outgoing packets to an external source.
  - Incoming packets that fail flow match and are forwarded again. In this case, the context ACL applies first and only if it passes are packets forwarded.

    During forwarding, if an ACL rule is added with a destination address as a loopback address, the context ACL is also applied. This is because StarOS handles packets destined to the kernel by going through a forwarding lookup for them. To apply ACL rules to incoming packets, the interface ACL must be used instead of the context ACL.
- The ACL to be applied must be configured in the context specified by this command.
- Up to 16 ACLs can be applied to a group provided that the number of rules configured within the ACL(s) does not exceed the 256-rule limit for the interface.

Applying an ACL to All Traffic Within a Context

This section provides information and instructions for applying one or more ACLs to a context configured within a specific context on the system. The applied ACLs, known as policy ACLs, contain rules that apply to all traffic facilitated by the context.

Important This section provides the minimum instruction set for applying the ACL list to all traffic within a context. For more information on commands that configure additional parameters and options, refer to the *Context Configuration Mode Commands* chapter in the *Command Line Interface Reference*.

To configure the system to provide access control list facility to subscribers:

Step 1 Apply the configured ACL as described in Applying the ACL to a Context, on page 315

Step 2 Verify that ACL is applied properly on interface as described in Verifying the ACL Configuration in a Context, on page 316

Step 3 Save your configuration to flash memory, an external memory device, and/or a network location using the Exec mode **save configuration** command. For additional information refer to the *Verifying and Saving Your Configuration* chapter.

Verifying the ACL Configuration in a Context

To verify the ACL configuration:

Verify that your ACL lists were applied properly by entering the following command in Exec Mode:

```
[local]host_name# show configuration context context_name
```

context_name is the name of the context to which the ACL(s) was/were applied.

The output of this command displays the configuration of the entire context. Examine the output for the commands pertaining to interface configuration. The commands display the ACL(s) applied using this procedure.

```
configure
   context context_name
      ip access-list acl_name
         deny host ip_address
         deny ip any host ip_address
         exit
      ip access-group access_group_name
      service-redundancy-protocol
      exit
      interface interface_name
         ip address ip_address/mask
         exit
      subscriber default
      exit
      aaa group default
      exit
      gtpp group default
               end
```

Applying an ACL to a RADIUS-based Subscriber

IP ACLs are applied to subscribers via attributes in their profile. The subscriber profile could be configured locally on the system or remotely on a RADIUS server.

To apply an ACL to a RADIUS-based subscriber, use the **Filter-Id** attribute.

For more details on this attribute, if you are using StarOS 12.3 or an earlier release, refer to the *AAA and GTPP Interface Administration and Reference*. If you are using StarOS 14.0 or a later release, refer to the *AAA Interface Administration and Reference*.

This section provides information and instructions for applying an ACL to an individual subscriber whose profile is configured locally on the system.

Important This section provides the minimum instruction set for applying the ACL list to all traffic within a context. For more information on commands that configure additional parameters and options, refer to the *Subscriber Configuration Mode Commands* chapter in the *Command Line Interface Reference*.

To configure the system to provide access control list facility to subscribers:

Step 1 Apply the configured access control list by following the example configuration in Applying an ACL to an Individual Subscriber, on page 317

Step 2 Verify that ACL is applied properly on interface by following the steps in Verifying the ACL Configuration to an Individual Subscriber, on page 317

Step 3 Save your configuration to flash memory, an external memory device, and/or a network location using the Exec mode **save configuration** command. For additional information refer to the *Verifying and Saving Your Configuration* chapter.

Applying an ACL to an Individual Subscriber

To apply the ACL to an individual subscriber, use the following configuration:

```
configure
   context acl_ctxt_name [ -noconfirm ]
      subscriber name subs_name
         { ip | ipv6 } access-group acl_list_name [ in | out ]
         end
```

Notes:

- The context name is the name of the ACL context containing the interface to which the ACL is to be applied.
- If neither the **in** nor the **out** keyword is specified, the ACL will be applied to all inbound and outbound packets.
- The ACL to be applied must be configured in the context specified by this command.
- Up to eight ACLs can be applied to a group provided that the number of rules configured within the ACL(s) does not exceed the 128-rule limit for the interface.

Verifying the ACL Configuration to an Individual Subscriber

These instructions are used to verify the ACL configuration.

Verify that your ACL lists were applied properly by entering the following command in Exec Mode:

```
[local]host_name# show configuration context context_name
```

context_name is the name of the context containing the subscriber *subs1* to which the ACL(s) was/were applied.

The output of this command displays the configuration of the entire context. Examine the output for the commands pertaining to interface configuration. The commands display the ACL(s) applied using this procedure.

```
configure
   context context_name
      ip access-list acl_name
         deny host ip_address
         deny ip any host ip_address
         exit
      ip access-group access_group_name
      service-redundancy-protocol
      exit
      interface interface
         ip address ip_address/mask
         exit
      subscriber default
      exit
      subscriber name subscriber_name
         ip access-group access_group_name in
         ip access-group access_group_name out
         exit
      aaa group default
      exit
      gtpp group default
      exit
      content-filtering server-group cfsg_name
         response-timeout response_timeout
         connection retry-timeout retry_timeout
               end
```

Applying an ACL to the Subscriber Named default

This section provides information and instructions for applying an ACL to the subscriber named *default*.

Important This section provides the minimum instruction set for applying the ACL list to all traffic within a context. For more information on commands that configure additional parameters and options, refer to *Subscriber Configuration Mode Commands* in the *Command Line Interface Reference*.

To configure the system to provide access control list facility to subscribers:

Step 1 Apply the configured access control list by following the example configuration in Applying an ACL to the Subscriber Named default, on page 318

Step 2 Verify that ACL is applied properly on interface by following the steps in Verifying the ACL Configuration to the Subscriber Named default, on page 319

Step 3 Save your configuration to flash memory, an external memory device, and/or a network location using the Exec mode **save configuration** command. For additional information refer to the *Verifying and Saving Your Configuration* chapter.

Applying an ACL to the Subscriber Named default

To apply the ACL to the subscriber named *default*, use the following configuration:

```
configure
   context acl_ctxt_name [ -noconfirm ]
      subscriber name subs_name
         { ip | ipv6 } access-group acl_list_name [ in | out ]
         end
```

Notes:

- The context name is the name of the ACL context containing the interface to which the ACL is to be applied.
- If neither the **in** nor the **out** keyword is specified, the ACL will be applied to all inbound and outbound packets.
- The ACL to be applied must be configured in the context specified by this command.
- Up to 16 ACLs can be applied to a group provided that the number of rules configured within the ACL(s) does not exceed the 256-rule limit for the interface.

Verifying the ACL Configuration to the Subscriber Named default

These instructions are used to verify the ACL configuration.

Verify that your ACL lists were applied properly by entering the following command in Exec Mode:

```
[local]host_name# show configuration context context_name
```

context_name is the name of the context containing the subscriber default to which the ACL(s) was/were applied.

The output of this command displays the configuration of the entire context. Examine the output for the commands pertaining to interface configuration. The commands display the ACL(s) applied using this procedure.

```
configure
   context context_name
      ip access-list acl_name
         deny host ip_address
         deny ip any host ip_address
         exit
      ip access-group access_group_name
      service-redundancy-protocol
      exit
      interface interface
         ip address ip_address/mask
         exit
      subscriber name default
         ip access-group access_group_name in
         ip access-group access_group_name out
         exit
      aaa group default
      exit
      gtpp group default
      exit
      content-filtering server-group cfsg_name
         response-timeout response_timeout
         connection retry-timeout retry_timeout
               end
```

Applying an ACL to Service-specified Default Subscriber

This section provides information and instructions for applying an ACL to the subscriber to be used as the "default" profile by various system services.

Important This section provides the minimum instruction set for applying the ACL list to all traffic within a context. For more information on commands that configure additional parameters and options, refer to the *Subscriber Configuration Mode Commands* chapter in the *Command Line Interface Reference*.

To configure the system to provide access control list facility to subscribers:

Step 1 Apply the configured access control list by following the example configuration in Applying an ACL to the Subscriber Named default, on page 318.

Step 2 Verify that the ACL is applied properly on interface by following the steps in Verifying the ACL Configuration to Service-specified Default Subscriber, on page 320.

Step 3 Save your configuration to flash memory, an external memory device, and/or a network location using the Exec mode **save configuration** command. For additional information refer to the *Verifying and Saving Your Configuration* chapter.

Applying an ACL to Service-specified Default Subscriber

To apply the ACL to a service-specified Default subscriber, use the following configuration:

```
configure
  context acl_ctxt_name [ -noconfirm ]
    { pdsn-service | fa-service | ha-service } service_name
      default subscriber svc_default_subs_name
      exit
    subscriber name svc_default_subs_name
      { ip | ipv6 } access-group acl_list_name [ in | out ]
      end
```

Notes:

- The context name is the name of the ACL context containing the interface to which the ACL is to be applied.
- If neither the **in** nor the **out** keyword is specified, the ACL will be applied to all inbound and outbound packets.
- The ACL to be applied must be configured in the context specified by this command.
- Up to eight ACLs can be applied to a group provided that the number of rules configured within the ACL(s) does not exceed the 128-rule limit for the interface.

Verifying the ACL Configuration to Service-specified Default Subscriber

To verify the ACL configuration.

Verify that your ACL lists were applied properly by entering the following command in Exec Mode:

```
[local]host_name# show configuration context context_name
```

context_name is the name of the context containing the service with the default subscriber to which the ACL(s) was/were applied.

The output of this command displays the configuration of the entire context. Examine the output for the commands pertaining to interface configuration. The commands display the ACL(s) applied using this procedure.

```
configure
   context context_name
      ip access-list acl_name
         deny host ip_address
         deny ip any host ip_address
         exit
      ip access-group access_group_name
      interface interface
         ip address ip_address/mask
         exit
      subscriber default
      exit
      subscriber name subscriber_name
         ip access-group access_group_name in
         ip access-group access_group_name out
         exit
      pdsn-service service_name
         default subscriber subscriber_name
               end
```

Applying a Single ACL to Multiple Subscribers

As mentioned in the previous section, IP ACLs are applied to subscribers via attributes in their profile. The subscriber profile could be configured locally on the system or remotely on a RADIUS server.

The system provides for the configuration of subscriber functions that serve as default values when specific attributes are not contained in the individual subscriber's profile. The following table describes these functions.

Table 22: Functions Used to Provide "Default" Subscriber Attributes

Function	Description
Subscriber named *default*	Within each context, the system creates a subscriber called *default*. The profile for the subscriber named *default* provides a configuration template of attribute values for subscribers authenticated in that context. Any subscriber attributes that are not included in a RADIUS-based subscriber profile is configured according to the values for those attributes as defined for the subscriber named *default*. **NOTE:** The profile for the subscriber named *default* is not used to provide missing information for subscribers configured locally.

Function	Description
default subscriber	This command allows multiple services to draw "default" subscriber information from multiple profiles.

When configured properly, the functions described in the table above could be used to apply an ACL to:

- All subscribers facilitated within a specific context by applying the ACL to the profile of the subscriber named *default*.
- All subscribers facilitated by specific services by applying the ACL to a subscriber profile and then using the **default subscriber** command to configure the service to use that subscriber as the "default" profile.

Applying an ACL to Multiple Subscriber via APNs

To apply the ACL to multiple subscribers via APN, use the following configuration:

```
configure
  context dest_context_name [-noconfirm]
    apn apn_name
      { ip | ipv6 } access-group acl_list_name [ in | out ]
      end
```

Notes:

- The ACL to be applied must be in the destination context of the APN (which can be different from the context where the APN is configured).
- If neither the **in** nor the **out** keyword is specified, the ACL will be applied to all inbound and outbound packets.
- This command supports only one ACL. However, the ACL can have up to 256 rules.
- Four access-groups can be applied for each APN, for example:

  ip access-group acl_list_name_1 in

  ip access-group acl_list_name_2 out

  ipv6 access-group acl_list_name_3 in

  ipv6 access-group acl_list_name_4 out

Applying an ACL to Multiple Subscriber via APNs

If IP ACLs are applied to subscribers via attributes in their profile, the subscriber profile could be configured locally on the system or remotely on a RADIUS server.

To reduce configuration time, ACLs can alternatively be applied to APN templates for GGSN subscribers. When configured, any subscriber packets facilitated by the APN template would then have the associated ACL applied.

This section provides information and instructions for applying an ACL to an APN template.

Important This section provides the minimum instruction set for applying the ACL list to all traffic within a context. For more information on commands that configure additional parameters and options, refer to the *Subscriber Configuration Mode Commands* chapter in the *Command Line Interface Reference*.

To configure the system to provide access control list facility to subscribers:

Step 1 Apply the configured access control list by following the example configuration in Applying an ACL to Multiple Subscriber via APNs, on page 322.

Step 2 Verify that ACL is applied properly on interface by following the steps in Verifying the ACL Configuration to APNs, on page 323.

Step 3 Save your configuration to flash memory, an external memory device, and/or a network location using the Exec mode **save configuration** command. For additional information refer to the *Verifying and Saving Your Configuration* chapter.

Verifying the ACL Configuration to APNs

To verify the ACL configuration:

Verify that your ACL lists were applied properly by entering the following command in Exec Mode:

show configuration context *context_name*

context_name is the name of the context containing the APN *apn1* having *default* subscriber to which the ACL(s) was/were applied.

The output of this command displays the configuration of the entire context. Examine the output for the commands pertaining to interface configuration. The commands display the ACL(s) applied using this procedure.

```
configure
   context context_name
      ip access-list acl_name
         deny host ip_address
         deny ip any host ip_address
         exit
      ip access-group access_group_name
      interface interface
         ip address ip_address/mask
         exit
      subscriber default
      exit
      apn apn_name
         ip access-group access_group_name in
         ip access-group access_group_name out
               end
```

CHAPTER 22

Congestion Control

This chapter describes the Congestion Control feature. It covers the following topics:

Overview

Congestion Control monitors the system for conditions that could potentially degrade performance when the system is under heavy load. Typically, these conditions are temporary (for example, high CPU or memory utilization) and are quickly resolved. However, continuous or large numbers of these conditions within a specific time interval may impact the system's ability to service subscriber sessions. Congestion control helps identify such conditions and invokes policies for addressing the situation.

Congestion control operation is based on configuring the following:

- **Congestion Condition Thresholds:** Thresholds dictate the conditions for which congestion control is enabled and establishes limits for defining the state of the system (congested or clear). These thresholds function in a way similar to operation thresholds that are configured for the system as described in the *Thresholding Configuration Guide*. The primary difference is that when congestion thresholds are reached, a service congestion policy and an SNMP trap (starCongestion) are generated.

  A threshold tolerance dictates the percentage under the configured threshold that must be reached in order for the condition to be cleared. An SNMP trap, starCongestionClear, is then triggered.

  - **Port Utilization Thresholds**: If you set a port utilization threshold, when the average utilization of all ports in the system reaches the specified threshold, congestion control is enabled.
  - **Port-specific Thresholds**: If you set port-specific thresholds, when any individual port-specific threshold is reached, congestion control is enabled system-wide.

- **Service Congestion Policies:** Congestion policies are configurable for each service. These policies dictate how services respond when the system detects that a congestion condition threshold has been crossed.

Important This section provides the minimum instruction set for configuring congestion control. Commands that configure additional interface or port properties are provided in *Subscriber Configuration Mode* in the *Command Line Interface Reference*. Always refer to the Administration Guides for all of the licensed products running on this platform for additional configuration information with respect to congestion control. Congestion control functionality varies based on product and StarOS version.

For the MME three levels of congestion control thresholds are supported – critical, major and minor. By default only the critical threshold is supported for other products. SNMP traps also support major and minor congestion control thresholds. A set of **congestion-action-profile** commands allows an operator to establish additional actions to be taken for specific thresholds and threshold levels.

Configuring Congestion Control

To configure Congestion Control functionality:

Step 1 Configure congestion control thresholds as described in Configuring the Congestion Control Threshold, on page 326

Step 2 Configure service congestion policies as described in Configuring Service Congestion Policies, on page 327

Step 3 Enable redirect overload policies as described in Enabling Congestion Control Redirect Overload Policy, on page 328

Step 4 Configure disconnecting subscribers based on call or inactivity time as described in Disconnecting Subscribers Based on Call or Inactivity Time, on page 328

Step 5 Save your configuration as described in the *Verifying and Saving Your Configuration* chapter.

Configuring the Congestion Control Threshold

To configure congestion control threshold, apply the following example configuration in the Global Configuration mode of the CLI:

```
configure
   congestion-control threshold max-sessions-per-service-utilization percent

   congestion-control threshold tolerance percent
   end
```

Notes:

- There are numerous threshold parameters. See *Global Configuration Mode Commands* in the *Command Line Interface Reference* for more information.
- The tolerance is the percentage under a configured threshold that dictates the point at which the condition is cleared.
- Multiple levels of congestion thresholds – critical, major and minor – a re supported for various types of congestion control thresholds. If a threshold level is not specified, the default is critical. Currently, major and minor thresholds are only supported for the MME. The **congestion-action-profile** command under **lte-policy** defines the action to be taken when thresholds are exceeded. See *Global Configuration*

Mode Commands, *LTE Policy Configuration Mode Commands* and *Congestion Action Profile Configuration Mode Commands* in the *Command Line Interface Reference* for more information.

- Repeat this configuration as needed for additional thresholds.

Configuring Service Congestion Policies

To create a congestion control policy, apply the following example configuration in the Global Configuration mode of the CLI:

```
configure
      congestion-control policy service action { drop | none | redirect |
reject }
      end
```

Notes:

- When the redirect action occurs for PDSN services, the PDSN responds to the PCF with a reply code of 136, "unknown PDSN address" along with the IP address of an alternate PDSN.
- **redirect** is not available for PDIF. The default action for PDIF is "none."
- When the redirect action occurs for HA services, the system responds to the FA with a reply code of 136, "unknown home agent address".
- **redirect** cannot be used in conjunction with GGSN services.
- **redirect** is not available for the Local Mobility Anchor (LMA) service.
- When setting the action to **reject**, the reply code is 130, "insufficient resources".
- For the GGSN, the reply code is 199, "no resources available".
- For the SaMOG, MME, **redirect** is not available.
- For the MME, create action profiles for optional major and minor thresholds using the **congestion-action-profile** command under **lte-policy** in the Global Configuration mode.
- For the MME, you can specify *service* as **critical**, **major** or **minor** to set a policy and associate an action-profile for the respective threshold. See *Global Configuration Mode Commands* in the *Command Line Interface Reference* for more information.

Configuring Overload Reporting on the MME

When an overload condition is detected on an MME and the report-overload keyword is enabled in the **congestion-control policy** command, the system reports the condition to a specified percentage of eNodeBs and proceeds to take the configured action on incoming sessions. To create a congestion control policy with overload reporting, apply the following example configuration:

```
configure
   congestion-control policy mme-service action report-overload
reject-new-sessions enodeb-percentage percentage
   end
```

Notes:

- Other overload actions include **permit-emergency-sessions** and **reject-non-emergency-sessions**.

Enabling Congestion Control Redirect Overload Policy

To create a congestion control policy and configure a redirect overload policy for the service, apply the following example configuration:

```
configure
   congestion-control
      context context_name
        {service_configuration_mode}
             policy overload redirect address
             end
```

Notes:

- *Optional:* If the congestion control policy action was configured to **redirect**, then a redirect overload policy must be configured for the service(s) that are affected.
- There are several service configuration modes that you can configure. See the *Command Line Interface Reference* for a complete list of modes.
- You can set various options for redirection. See the *Command Line Interface Reference* for more information.
- Repeat this configuration example to configure overload policies for additional services configured in the same context.

Verify the Service Overload Policies

To verify that the service overload policies were properly configured enter the following command in the Exec Mode:

```
[local]host_name# show service_type name service_name
```

This command lists the entire service configuration. Verify that the information displayed for the "Overload Policy" is accurate.

Repeat this configuration example to configure additional services in other contexts.

Verify the Congestion Control Configuration

Verify MME Congestion Action Profiles

To verify MME multilevel congestion action profiles, run the following Exec mode command:

```
[local]host_name# show lte-policy congestion-action-profile { name profile_name
 | summary }
```

Disconnecting Subscribers Based on Call or Inactivity Time

During periods of heavy system load, it may be necessary to disconnect subscribers in order to maintain an acceptable level of system performance. You can establish thresholds to select subscribers to disconnect based on the length of time that a call has been connected or inactive.

To enable overload disconnect for the currently selected subscriber, use the following configuration example:

```
configure
   context context_name
      subscriber name subscriber_name
         default overload-disconnect threshold inactivity-time dur_thresh
         default overload-disconnect threshold connect-time dur_thresh
         end
```

To disable the overload disconnect feature for this subscriber, use the following configuration example:

```
configure
  context context_name
    subscriber subscriber_name
    no overload-disconnect { [threshold inactivity-time] | [threshold
connect-time] }
    end
```

CHAPTER 23

Routing

This chapter provides information on configuring an enhanced, or extended, service. The product administration guides provide examples and procedures for configuring basic services on the system. You should select the configuration example that best meets your service model, and configure the required elements for that model before using the procedures described below.

This chapter includes the following sections:

Routing Policies

This section describes how to configure the elements needed to define routing policies. Routing policies modify and redirect routes to and from the system to satisfy specific network deployment requirements.

Use the following building blocks to configure routing policies:

- **Route Access Lists** – The basic building block of a routing policy. Route access lists filter routes based on a range of IP addresses.
- **IP Prefix Lists** – A more advanced element of a routing policy. An IP Prefix list filters routes based on IP prefixes.
- **AS Path Access Lists** – A basic building block used for Border Gateway Protocol (BGP) routing. These lists filter Autonomous System (AS) paths.
- **Route Maps** – Route-maps provide detailed control over routes during route selection or route advertisement by a routing protocol, and in route redistribution between routing protocols. For this level of control you use IP Prefix Lists, Route Access Lists and AS Path Access Lists to specify IP addresses, address ranges, and Autonomous System paths.

Creating IP Prefix Lists

Use the following configuration example to create IP Prefix Lists:

```
config
   context context_name
      ip prefix-list name list_name { deny | permit } network_address/net_mask
```

Notes:

- Set the IP prefix list to deny, permit or match any prefix.
- IPv4 dotted-decimal and IPv6 colon-separated-hexadecimal addresses are supported.
- Save your configuration as described in the *Verifying and Saving Your Configuration* chapter.

Creating Route Access Lists

Use the following procedure to create a Route Access List:

```
config
   context context_name
     route-access-list { extended identifier } { deny | permit } [ ip
address ip_address ]
     route-access-list named list_name { deny | permit } { ip_address/mask |
any } [ exact-match ]
route-access-list
standard identifier { permit | deny ) { ip_address
wildcard_mask | any |network_address }
```

Notes:

- A maximum of 64 access lists are supported per context.
- A maximum of 16 entries can defined for each route-access-list.
- Save your configuration as described in the *Verifying and Saving Your Configuration* chapter.

Creating AS Path Access Lists

Use the following procedure to create an AS Path Access List:

```
config
   context context_name
     ip as-path access-list list_name [ { deny | permit } reg_expr ]
```

Notes:

- Save your configuration as described in the *Verifying and Saving Your Configuration* chapter.

Creating Route Maps

Use the following configuration example to create a Route Map:

```
config
   context context_name
      route-map map_name { deny | permit } seq_number
```

Notes:

- Use the **match** and **set** commands in Route Map Configuration mode to configure the route map. Refer to the *Command Line Interface Reference* for more information on these commands.
- Save your configuration as described in the *Verifying and Saving Your Configuration* chapter.

Sample Configuration

The example below shows a configuration that creates two route access lists, applies them to a route map, and uses that route map for a BGP router neighbor.

```
config
   context isp1
      route-access-list named RACLin1a permit 88.151.1.0/30
      route-access-list named RACLin1b permit 88.151.1.4/30
      route-access-list named RACLany permit any
      route-map RMnet1 deny 100
         match ip address route-access-list RACLin1a
         #exit
         route-map RMnet1 deny 200
         match ip address route-access-list RACLin1b
         #exit
      route-map RMnet1 permit 1000
         match ip address route-access-list RACLany
         #exit
      router bgp 1
         neighbor 152.20.1.99 as-path 101
         neighbor 152.20.1.99 route-map RMnet1 in
```

Static Routing

The system supports static network route configuration on a per context basis. Define network routes by specifying the:

- IP address and mask for the route
- Name of the interface in the current context that the route must use
- Next hop IP address

Important On the VPC-DI, static routes with IPv6 prefix lengths less than /12 and between the range of /64 and /128 are not supported.

Adding Static Routes to a Context

To add static routes to a context configuration, you must know the names of the interfaces that are configured in the current context. Use the **show ip interface** command to list the interfaces in the current context (Exec mode).

Information for all interfaces configured in the current context is displayed as shown in the following example.

```
[local]host_name# show ip interface
Intf Name: Egress 1
Description:
IP State: Up (Bound to slot/port untagged ifIndex 402718721)
IP Address: 192.168.231.5
Subnet Mask: 255.255.255.0
Bcast Address: 192.168.231.255
MTU: 1500
Resoln Type: ARP          ARP timeout: 3600 secs
L3 monitor LC-port switchover: Disabled
Number of Secondary Addresses: 0
Total interface count: 1
```

The first line of information for each interface lists the interface name for the current context as shown in the example output. In this example, there is one interface with the name *Egress 1*.

```
config
   context context_name
      ip route { ip_address [ ip_mask ] | ip_addr_mask_combo } { next-hop
next_hop_address | egress_name [ precedence precedence [ cost cost ]
```

Notes:

- You can configure a maximum of 1,200 static routes per context. Save your configuration as described in the *Verifying and Saving Your Configuration* chapter.

Deleting Static Routes From a Context

Use the following configuration example to remove static routes from a context's configuration:

```
config
   context context_name
       no ip route { ip_address ip_mask | ip_addr_mask_combo } next_hop_address
egress_name [ precedence precedence ] [ cost cost ]
```

Notes

- Save your configuration as described in the *Verifying and Saving Your Configuration* chapter.

OSPF Routing

This section gives an overview of Open Shortest Path First (OSPF) routing and its implementation in the system. It also describes how to enable the base OSPF functionality and lists the commands that are available for more complex configurations.

You must purchase and install a license key before you can use this feature. Contact your Cisco account representative for more information on licenses.

Important During system task recovery, it is possible for a dynamically-learned forwarding entry to incorrectly remain in the system forwarding table if that forwarding entry has been removed from the dynamic routing protocol during the recovery.

Important On the VPC-DI, OSPF routes with IPv6 prefix lengths less than /12 and between the range of /64 and /128 are not supported.

OSPF Version 2 Overview

OSPF is a link-state routing protocol that employs an interior gateway protocol (IGP) to route IP packets using the shortest path first based solely on the destination IP address in the IP packet header. OSPF routed IP packets are not encapsulated in any additional protocol headers as they transit the network.

An Autonomous System (AS), or Domain, is defined as a group of networks within a common routing infrastructure.

OSPF is a dynamic routing protocol that quickly detects topological changes in the AS (such as router interface failures) and calculates new loop-free routes after a period of convergence. This period of convergence is short and involves a minimum of routing traffic.

In a link-state routing protocol, each router maintains a database, referred to as the link-state database, that describes the Autonomous System's topology. Each participating router has an identical database. Each entry in this database is a particular router's local state (for example, the router's usable interfaces and reachable neighbors). The router distributes its local state throughout the AS by flooding.

All routers run the same algorithm in parallel. From the link-state database, each router constructs a tree of shortest paths with itself as root to each destination in the AS. Externally derived routing information appears on the tree as leaves. The cost of a route is described by a single dimensionless metric.

OSPF allows sets of networks to be grouped together. Such a grouping is called an area. The topology of this area is hidden from the rest of the AS, which enables a significant reduction in routing traffic. Also, routing within the area is determined only by the area's own topology, lending the area protection from bad routing data. An area is a generalization of an IP subnetted network.

OSPF enables the flexible configuration of IP subnets so that each route distributed by OSPF has a destination and mask. Two different subnets of the same IP network number may have different sizes (that is, different masks). This is commonly referred to as variable-length subnetting. A packet is routed to the best (longest or most specific) match. Host routes are considered to be subnets whose masks are "all ones" (0xffffffff).

OSPF traffic can be authenticated or non-authenticated, or can use no authentication, simple/clear text passwords, or MD5-based passwords. This means that only trusted routers can participate in the AS routing. You can specify a variety of authentication schemes and, in fact, you can configure separate authentication schemes for each IP subnet.

Externally derived routing data (for example, routes learned from an exterior protocol such as BGP) is advertised throughout the AS. This externally derived data is kept separate from the OSPF ink state data.

Each external route can also be tagged by the advertising router, enabling the passing of additional information between routers on the boundary of the AS.

OSPF uses a link-state algorithm to build and calculate the shortest path to all known destinations.

Basic OSPFv2 Configuration

This section describes how to implement basic OSPF routing.

Enabling OSPF Routing For a Specific Context

Use the following configuration example to enable OSPF Routing for a specific context:

```
config
  context context_name
    router ospf
    end
```

Notes:

- Save your configuration as described in the *Verifying and Saving Your Configuration* chapter.

Enabling OSPF Over a Specific Interface

After you enable OSPF, specify the networks on which it will run. Use the following command to enable OSPF:

```
network network_ip_address/network_mask area { area_id | area_ip_address }
```

Important The default cost for OSPF on the system is 10. To change the cost, refer to the **ip ospf cost** command in the *Ethernet Interface Configuration Mode Commands* chapter of the *Command Line Interface Reference*.

Notes:

- Save your configuration as described in the *Verifying and Saving Your Configuration* chapter.

Redistributing Routes Into OSPF (Optional)

Redistributing routes into OSPF means any routes from another protocol that meet specified a specified criterion, such as route type, metric, or rule within a route-map, are redistributed using the OSPFv2 protocol to all OSPF areas. This is an optional configuration.

```
config
  context context_name
    router ospf
      redistribute { connected  | static }
      end
```

Notes:

- Save your configuration as described in the *Verifying and Saving Your Configuration* chapter.

Confirming OSPF Configuration Parameters

To confirm the OSPF router configuration, use the following command and look for the section labeled **router ospf** in the screen output:

```
show config context ctxt_name [ verbose ]
```

OSPFv3 Routing

This section gives an overview of Open Shortest Path First Version 3 (OSPFv3) routing and its implementation in the system. It also describes how to enable the base OSPFv3 functionality and lists the commands that are available for more complex configurations.

Important On the VPC-DI, OSPFv3 routes with IPv6 prefix lengths less than /12 and between the range of /64 and /128 are not supported.

OSPFv3 Overview

Much of OSPF version 3 is the same as OSPF version 2. OSPFv3 expands on OSPF version 2 to provide support for IPv6 routing prefixes and the larger size of IPv6 addresses. OSPFv3 dynamically learns and advertises (redistributes) IPv6 routes within an OSPFv3 routing domain

In OSPFv3, a routing process does not need to be explicitly created. Enabling OSPFv3 on an interface will cause a routing process and its associated configuration to be created.

Basic OSPFv3 Configuration

This section describes how to implement basic OSPF routing.

Enabling OSPFv3 Routing For a Specific Context

Use the following configuration example to enable OSPF Routing for a specific context:

```
config
  context context_name
    router ospfv3
    end
```

Notes:

- Save your configuration as described in the *Verifying and Saving Your Configuration* chapter.

Enabling OSPFv6 Over a Specific Interface

After you enable OSPFv3 specify the area in which it will run. Use the following command to enable OSPFv3:

```
area { area_id | area_ip_address } [ default-cost dflt-cost ] [ stub stub-area ]
[ virtual-link vl-neighbor-ipv4address ]
```

Important The default cost for OSPFv3 on the system is 10. To change the cost, refer to the **ipv6 ospf cost** command in the *Ethernet Interface Configuration Mode Commands* chapter of the *Command Line Interface Reference*.

Notes:

- Save your configuration as described in the *Verifying and Saving Your Configuration* chapter.

Redistributing Routes Into OSPFv3 (Optional)

Redistributing routes into OSPFv3 means any routes from another protocol that meet specified a specified criterion, such as route type, metric, or rule within a route-map, are redistributed using the OSPFv3 protocol to all OSPF areas. This is an optional configuration.

```
config
  context context_name
    router ospf3
      redistribute { connected | static }
      end
```

Notes:

- Save your configuration as described in the *Verifying and Saving Your Configuration* chapter.

Confirming OSPFv3 Configuration Parameters

To confirm the OSPF router configuration, use the following command and look for the section labeled **router ipv6 ospf** in the screen output:

```
show config context ctxt_name [ verbose ]
```

Equal Cost Multiple Path (ECMP)

The system supports ECMP for routing protocols. ECMP distributes traffic across multiple routes that have the same cost to lessen the burden on any one route.

ECMP can be used in conjunction with most routing protocols, since it is a per-hop decision that is limited to a single router. It potentially offers substantial increases in bandwidth by load-balancing traffic over multiple paths

The following command configures the maximum number of equal cost paths that can be submitted by a routing protocol:

```
config
  context context_name
    ip routing maximum-paths [ max_num ]
```

Notes:

- *max_num* is an integer from 1 through 10 (*releases prior to 18.2*) or 1 through 32 (*release 18.2+*).
- Save your configuration as described in the *Verifying and Saving Your Configuration* chapter.

BGP-4 Routing

The Border Gateway Protocol 4 (BGP-4) routing protocol is supported through a BGP router process that is implemented at the context level.

Border Gateway Protocol (BGP) is an inter-AS routing protocol. An Autonomous System (AS) is a set of routers under a single technical administration that use an interior gateway protocol and common metrics to route packets within the AS. The set of routers uses an exterior gateway protocol to route packets to other autonomous systems.

BGP runs over TCP. This eliminates the need for the BGP protocol to implement explicit update fragmentation, retransmission, acknowledgement, and sequencing information. Any authentication scheme used by TCP may be used in addition to BGP's own authentication mechanisms.

BGP routers exchange network reachability information with other BGP routers. This information builds a picture of AS connectivity from which routes are filtered and AS-level policy decisions are enforced.

BGP-4 provides classless inter-domain routing. This includes support for advertising an IP prefix and eliminates the concept of network class within BGP. BGP-4 also allows the aggregation of routes, including the aggregation of AS paths.

Important On the VPC-DI, BGP routes with IPv6 prefix lengths less than /12 and between the range of /64 and /128 are not supported.

Overview of BGP Support

Mobile devices communicate to the Internet through Home Agents (HAs). HAs assign IP addresses to the mobile node from a configured pool of addresses. These addresses are also advertised to Internet routers through an IP routing protocol to ensure dynamic routing. The BGP-4 protocol is used as a monitoring mechanism between an HA and Internet router with routing to support Interchassis Session Recovery (ICSR). (Refer to *Interchassis Session Recovery* for more information.)

The objective of BGP-4 protocol support is to satisfy routing requirements and monitor communications with Internet routers. BGP-4 may trigger an active to standby switchover to keep subscriber services from being interrupted.

The following BGP-4 features are supported:

- Exterior Border Gateway Protocol (EBGP) multi-hop
- Route Filtering for inbound and outbound routes
- Route redistribution and route-maps
- Support for BGP communities and extended communities in route maps
- Local preference for IPv4 and IPv6 (IBGP peers)

IP pool routes and loopback routes are advertised in the BGP domain in the following ways:

- Through BGP Configuration Mode **redistribution** commands, all or some of the connected routes are redistributed into the BGP domain. (IP pool and loopback routes are present in the IP routing table as connected routes.) The **network routemap** command provides the flexibility to change many BGP attributes.
- Through the BGP Configuration Mode **network** commands, connected routes are explicitly configured for advertisement into the BGP domain. The **network routemap** command provides the flexibility to change many BGP attributes. Refer to the *BGP Configuration Mode Commands* chapter of the *Command Line Interface Reference* for details on these commands.

Important If a BGP task restarts because of a processing card failure, a migration, a crash, or the removal of a processing card, all peering session and route information is lost.

Configuring BGP

This section describes how to configure and enable basic BGP routing support in the system.

```
config
   context context_name
      router bgp AS_number
         neighbor ip_address remote-as AS_num
```

Notes:

- A maximum of 64 BGP peers are supported per context.
- Save your configuration as described in the *Verifying and Saving Your Configuration* chapter.

Redistributing Routes Into BGP (Optional)

Redistributing routes into BGP simply means that any routes from another protocol that meet a specified criterion, such as a route type, or a rule within a route-map, are redistributed through the BGP protocol to all BGP areas. This is an optional configuration.

```
config
   context context_name
      router bgp as_number
         redistribute bgp { bgp | connected | static } [ metric  metric_value
 ] [ metric-type { 1 | 2 } ] [ route-map  route_map_name ]
```

Notes:

- The redistribution options are connected, ospf, rip, or static. Refer to the *Border Gateway Protocol Configuration Mode Commands* chapter of the *Command Line Interface Reference* for details on the **redistribute** command.
- A maximum of 64 route-maps are supported per context.
- Save your configuration as described in the *Verifying and Saving Your Configuration* chapter.

BGP Communities and Extended Communities

Route filtering based on a BGP community or extended community (route target) is configured via CLI Route Map Configuration mode commands.

BGP Communities

Configuring a BGP Community

A BGP community is a group of destinations that share some common attribute. Each destination can belong to multiple communities. Autonomous system administrators define to which communities a destination belongs.

You configure a BGP community via a Context Configuration mode command.

```
config
  context context_name
      ip community-list { named named_list | standard  identifier } { deny |
permit } { internet | local-AS | no-advertise | no-export | value
AS-community_number AS-community_number AS-community_number ...}
       { internet | local-AS | no-advertise | no-export | value
AS-community_number AS-community_number AS-community_number ...}
       { internet | local-AS | no-advertise | no-export | value
AS-community_number AS-community_number AS-community_number ...}
```

You can permit or deny the following BGP community destinations.

- **internet** – Advertise this route to the internet community, and any router that belongs to it.
- **local-AS** – Use in confederation scenarios to prevent sending packets outside the local autonomous system (AS).
- **no-advertise** – Do not advertise this route to any BGP peer, internal or external.
- **no-export** – Do not advertise to external BGP (eBGP) peers. Keep this route within an AS.
- **value** *AS-community_number* – Specifies a community string in AS:NN format, where AS = 2-byte AS-community hexadecimal number and NN = 2-byte hexadecimal number (1 to 11 characters).

You can enter multiple destinations and AS community numbers for each community. For additional information, see the *Command Line Interface Reference*.

Multiple community-list entries can be attached to a community-list by adding multiple permit or deny clauses for various community strings. Up to 64 community-lists can be configured in a context.

Setting the Community Attribute

You set the BGP community attribute via a **set community** command in a route map.

```
config
  context context_name
      route-map map_name { deny | permit } sequence_number
         set community [additive]{ internet | local-AS | no-advertise |
no-export | none | value AS-community_number AS-community_number AS-community_number
 ...}
         { internet | local-AS | no-advertise | no-export | none | value
 AS-community_number AS-community_number AS-community_number ... }
         { internet | local-AS | no-advertise | no-export | none | value
 AS-community_number AS-community_number AS-community_number ... }
```

The **additive** option allows you to enter multiple destinations and AS community numbers. For additional information, see the *Command Line Interface Reference*.

Filtering via a BGP Community

To filter routes based on a BGP community, you configure a **match** clause in a route map. The command sequence follows below.

```
config
   context context_name
       route-map map_name { deny | permit } sequence_number
          match community { named  named_list | standard  identifier }
```

BGP Extended Communities

Configuring a BGP Extended Community (Route Target)

A BGP extended community defines a route target. MPLS VPNs use a 64-bit Extended Community attribute called a Route Target (RT). An RT enables distribution of reachability information to the correct information table.

You configure a BGP extended community via a Context Configuration mode command.

```
config
   context context_name
       ip extcommunity-list { named named_list | standard  identifier } { deny
 | permit } rt rt_number rt_number rt_number ...
```

rt_number specifies a Route Target as a string in AS:NN format, where AS = 2-byte AS-community hexadecimal number and NN = 2-byte hexadecimal number (1 to 11 characters). You can add multiple route numbers to an IP extcommunity list.

Multiple extended community-list entries can be attached to an extended community-list by adding multiple permit or deny clauses for various extended community strings. Up to 64 extended community-lists can be configured in a context.

Setting the Extended Community Attribute

You set the BGP extended community attribute via a **set extcommunity** command in a route map.

```
config
   context context_name
       route-map map_name { deny | permit } sequence_number
        set extcommunity rt rt_number rt_number rt_number ...
```

rt_number specifies a Route Target as a string in AS:NN format, where AS = 2-byte AS-community hexadecimal number and NN = 2-byte hexadecimal number (1 to 11 characters). You can add multiple route numbers to an IP extcommunity list.

Filtering via a BGP Extended Community

To filter routes based on a BGP extended community (route target), you configure a **match** clause in a route map. The command sequence follows below.

```
config
   context context_name
       route-map map_name { deny | permit }
         [no] match extcommunity { named  named_list | standard  identifier }
```

BGP Local Preference

The BGP local preference attribute is sent by a BGP speaker only to IBGP peers. It is set in a route map via the following command sequence:

```
config
  context context_name
      route-map map_name { deny | permit }
         set local-preference pref_number
```

There is no **match** clause corresponding to local preference in the route-map because local-preference is directly used in the route selection algorithm.

ICSR and SRP Groups

BGP is employed with Interchassis Session Recovery (ICSR) configurations linked via Service Redundancy Protocol (SRP). By default an ICSR failover is triggered when all BGP peers within a context are down.

Optionally, you can configure SRP peer groups within a context. ICSR failover would then occur if all peers within a group fail. This option is useful in deployments in which a combination of IPv4 and IPv6 peers are spread across multiple paired VLANs, and IPv4 or IPv6 connectivity is lost by all members of a peer group.

For additional information refer to *Interchassis Session Recovery* in this guide and the description of the **monitor bgp**, **monitor diameter** and **monitor authentication-probe** commands in the *Service Redundancy Protocol Configuration Mode Commands* chapter of the *Command Line Interface Reference*.

Advertising BGP Routes from a Standby ICSR Chassis

An SRP Configuration mode command enables advertising BGP routes from an ICSR chassis in standby state. This command and its keywords allow an operator to take advantage of faster network convergence accrued from deploying BGP Prefix Independent Convergence (PIC) in the Optical Transport Network Generation Next (OTNGN).

BGP PIC is intended to improve network convergence which will safely allow for setting aggressive ICSR failure detection timers.

```
configure
   context context_name
      service-redundancy-protocol
       advertise-routes-in-standby-state [ hold-off-time hold-off-time ] [
reset-bfd-nbrs bfd-down-time ]
       end
```

Notes:

- **hold-off-time** *hold-off-time* delays advertising the BGP routes until the timer expires. Specify *hold-off-time* in seconds as an integer from 1 to 300.
- After resetting BFD, **reset-bfd-nbrs** *bfd-down-time* keeps the BFD sessions down for the configured number of milliseconds to improve network convergence. Specify *bfd-down-time* as an integer from 50 to 120000.

Configurable BGP Route Advertisement Interval for ICSR

By default, the MinRtAdvInterval is set for each peer with a value of 5 seconds for an iBGP peer and 30 seconds for an eBGP peer. An operator can use the **neighbor** *identifier* **advertisement-interval** command to globally change the default interval.

The BGP advertisement-interval can also be separately set for each address family. If configured, this value over-rides the peer's default advertisement-interval for that address-family only. BGP will send route update-message for each AFI/SAFI based on the advertisement-interval configured for that AFI/SAFI. If no AFI/SAFI advertisement-interval is configured, the peer-based default advertisement-interval is used.

In ICSR configurations, this feature can be used to speed route advertisements and improve network convergence times.

The **timers bgp icsr-aggr-advertisement-interval** command is available in both the BGP Address-Family (VPNv4/VPNv6) Configuration and BGP Address-Family (VRF) Configuration modes.

```
configure
   context context_name
      router bgp as_number
         address-family { ipv4 | ipv6 | vpnv4 | vpnv6 }
            timers bgp icsr-aggr-advertisement-interval seconds
```

Notes:

- *seconds* – sets the number of seconds as an integer from 0 to 30. Default: 0.

BGP CLI Configuration Commands

The following table lists the BGP Configuration mode CLI commands that support the configuration of various BGP parameters. For additional information, refer to the *BGP Configuration Mode Commands* chapter of the *Command Line Interface Reference*

.

```
configure
   context context_name
      router bgp as_number
```

Table 23: BGP Configuration Mode CLI Commands

bgp Command	Description
accept-zero-as-rd	Configures to accept VPN prefixes with Route Distinguisher (RD) value having Administrator Subfield, which is an AS number 0.
address-family { ipv4 | ipv6 }	Enters the IPv4 or IPv6 Address Family configuration mode.
address-family { vpnv4 | vpnv6 }	Enters the VPNv4 or VPNv6 Address Family configuration mode.
bgp graceful-restart { restart-time *rest_time* **| stalepath-time** *stale_time* **| update-delay** *delay*	Defines the BGP-specific parameters regarding graceful restarts.

bgp Command	Description
description *text*	Allows you to enter descriptive text for this configuration.
distance { admin *distance* **prefix** *prefix_addr* **[route-access-list** *list_name* **] | bgp external** *ebgp_dist* **internal** *ibgp_dist* local *local_dist* **}**	Defines the administrative distance for routes. The administrative distance is the default priority for a specific route or type route.
enforce-first-as	Enforces the first AS for Exterior Border Gateway Protocol (eBGP) routes.
ip vrf *vrf_name*	Adds a preconfigured IP VRF context instance to the BGP ASN and configures the BGP attributes and related parameters to the VRF.
maximum-paths { ebgp *max_num* **| ibgp** *max_num* **}**	Enables forwarding packets over multiple paths and specifies the maximum number of external BGP (eBGP) or internal BGP (iBGP) paths between neighbors.
neighbor *ip_address* **{ activate | advertisement-interval** *adv_time* **| capability graceful-restart | default-originate [route-map** *map_name* **] | distribute-list** *dist_list***{ in | out } | ebgp-multihop [max-hop number] | encrypted password** *encrypted_password* **| fall-over bfd [multihop] | filter-list filt_list { in | out } | max-prefix** *max_num* **[threshold** *thresh_percent* **] [warning-only] | next-hop-self | password** *password* **| remoteas** *AS_num* **| remove-private-AS | restart-time** *rest_time* **| route-map** *map_name* **{ in | out } | send-community { both | extended | standard } | shutdown | srp-activated-soft-clear | timers { [connect-interval** *conn_time* **] | [keepalive-interval** *keep_time* **holdtimeinterval** *hold_time* **] } | update-source** *ip_address* **| weight** *value* **}**	Configures BGP routers that interconnect to non-broadcast networks. Note that a remote AS number must be specified for a neighbor before other parameters can be configured. **Note:** The **advertisement-interval** must be explicitly configured for an address-family so that it can take effect for that address-family. By default it will be applicable only for the IPv4 address-family. Specify the address family via the **address-family** command. You can then set the neighbor advertisement-interval in the address family configuration mode.
network *ip_address/mask* **[route-map** *map_name* **]**	Specifies a network to announce via BGP.
redistribute { connected | ospf | rip | static } [route-map *map_name* **]**	Redistributes routes via BGP from another protocol to BGP neighbors.
router-id *ip_address*	Overrides the configured router identifier and causes BGP peers to reset.
scan-time *time*	Configures the BGP background scanner interval in seconds. BGP monitors the next hop of the installed routes to verify next-hop reachability and to select, install, and validate the BGP best path. loops.
timers bgp keepalive-interval *interval* **holdtime-interval** *time* **[min-peer-holdtimeinterval** *time* **]**	Configures BGP routing timers.

Confirming BGP Configuration Parameters

To confirm the BGP router configuration, use the following command and look for the section labeled **router bgp** in the screen output:

```
show config context ctxt_name [ verbose ]
```

BGP Peer Limit

Feature Summary and Revision History

Summary Data

Applicable Product(s) or Functional Area	All
Applicable Platform(s)	VPC - DI
Feature Default	Disabled - Configuration Required
Related Changes in This Release	Not Applicable
Related Documentation	• *Command Line Interface Reference* • *Statistics and Counters Reference* • *VPC-DI System Administration Guide*

Revision History

Revision Details	Release
First introduced.	21.8

Feature Description

In the Cisco Virtualized Packet Core Distributed Instance (VPC-DI)with CUPS architecture, the flexibility of BGP peering is provided across packet processing cards namely, Session Function (SF) cards, including the demux SF cards.

In deployment setups based on “contrail” model of the SDN , each packet processing card has a vRouter within the compute node. In this model, with the current flexible BGP peering scheme, the BGP configurations needs to be implemented on each of those vRouters. This poses a challenge to service providers when there are large number of SF cards in their network. The number of lines of configuration required, poses a scaling challenge.

To overcome this challenge, the BGP Peer Limit feature is introduced that restricts BGP peering to only **two** SF cards in the VPC-DI architecture. This feature mandates that the routing table has only two routes corresponding to the two SF cards, with a third route being a “blackhole” or a “null” route. To ensure that the new routes are longest-prefix-match routes, provisioning of only host-addresses only (/32 bitmask) is enforced. This drastically reduces the amount of configuration and the routing table size.

How It Works

This feature is implemented using the **ip route kernel** command. When configured, BGP peering is restricted to only the two SF cards with the special route.

When the **blackhole** keyword is configured, it enables the kernel routing engine to block or drop packets going out of the node. This is not limited to any interface and defaults to a wildcard interface.

For information on configuring the BGP Peer Limit feature, see the "Configuring BGP Peer Limit" section.

Limitations

- This feature support is limited only to the context level.
- There is no support provided at the VRF level.
- This feature is supported only for IPv4.

Configuring BGP Peer Limit

The following section provides the configuration command to enable or disable the functionality.

Configuring Packet Processing Card Routes

Use the following CLI commands to add the special (static) route to any two packet processing interfaces (SF cards) defined in the context configuration.

```
configure
   context context_name
     [ no ] ip route kernel ip_address/ip_address_mask_combo egress_intrfc_name
 cost number
     end
```

NOTES:

- **no**: Deletes the added routes.
- **kernel**: Allows static route in the kernel routing table options.
- **ip_address/ip_address_mask_combo**: Specifies a combined IP address subnet mask bits to indicate what IP addresses the route applies to. *ip_address_mask_combo* must be specified using CIDR notation where the IP address is specified using IPv4 dotted-decimal notation and the mask bits are a numeric value, which is the number of bits in the subnet mask.
- *egress_intrfc_name* : Specifies the name of an existing egress interface as an alphanumeric string of 1 through 79 characters.
- **cost** *number* : Defines the number of hops to the next gateway. The cost must be an integer from 0 through 255 where 255 is the most expensive. Default is 0.
- This functionality is disabled by default.

Configuring Blackhole Route

Use the following CLI commands to block or drop packets going out of the node.

```
configure
   context context_name
```

```
[ no ] ip route kernel ip_address/ip_address_mask_combo egress_intrfc_name
cost number blackhole
end
```

NOTES:

- **no**: Deletes the added routes.
- **kernel**: Allows static route in the kernel routing table options.
- **ip_address/ip_address_mask_combo**: Specifies a combined IP address subnet mask bits to indicate what IP addresses the route applies to. *ip_address_mask_combo* must be specified using CIDR notation where the IP address is specified using IPv4 dotted-decimal notation and the mask bits are a numeric value, which is the number of bits in the subnet mask.
- *egress_intrfc_name* : Specifies the name of an existing egress interface as an alphanumeric string of 1 through 79 characters. The default is "*", that is, a wildcard interface.
- **cost** *number* : Defines the number of hops to the next gateway. The cost must be an integer from 0 through 255 where 255 is the most expensive. The default is 0.
- **blackhole**: Defines the blackhole route to install in the kernel to to block or drop packets.
- This functionality is disabled by default.

Monitoring and Troubleshooting

This section provides information regarding the CLI command available in support of monitoring and troubleshooting the feature.

Show Command(s) and/or Outputs

This section provides information regarding the show command and/or its output in support of this feature.

show ip route

This show command CLI now includes the value for the following new field when a static route is added to any two packet processing interfaces (SF cards).

kernel-only

Bidirectional Forwarding Detection

Bidirectional Forwarding Detection (BFD) is a network protocol used to detect faults between two forwarding engines connected by a link. BFD establishes a session between two endpoints over a particular link. If more than one link exists between two systems, multiple BFD sessions may be established to monitor each one of them. The session is established with a three-way handshake, and is torn down the same way. Authentication may be enabled on the session. A choice of simple password, MD5 or SHA1 authentication is available.

Important On the VPC-DI, BFD routes with IPv6 prefix lengths less than /12 and between the range of /64 and /128 are not supported.

Overview of BFD Support

BFD does not have a discovery mechanism; sessions must be explicitly configured between endpoints. BFD may be used on many different underlying transport mechanisms and layers, and operates independently of all of these. Therefore, it needs to be encapsulated by whatever transport it uses.

Protocols that support some form of adjacency setup, such as OSPF or IS-IS, may also be used to bootstrap a BFD session. These protocols may then use BFD to receive faster notification of failing links than would normally be possible using the protocol's own keepalive mechanism.

In asynchronous mode, both endpoints periodically send Hello packets to each other. If a number of those packets are not received, the session is considered down.

When Echo is active, a stream of Echo packets is sent to the other endpoint which then forwards these back to the sender. Echo can be globally enabled via the **bfd-protocol** command, and/or individually enabled/disabled per interface. This function is used to test the forwarding path on the remote system.

The system supports BFD in asynchronous mode with optional Echo capability via static or BGP routing.

Important On an ASR 5500 one of the packet processing cards must be configured as a demux card in order for BFD to function. See the *Configuring a Demux Card* section in the *System Settings* chapter for additional information.

Configuring BFD

This section describes how to configure and enable basic BFD routing protocol support in the system.

There are several factors affecting the configuration of BFD protocol:

- Configuring a BFD Context, on page 349
- Configuring IPv4 BFD for Static Routes, on page 350
- Configuring IPv6 BFD for Static Routes, on page 350
- Configuring BFD for Single Hop, on page 350
- Configuring Multihop BFD, on page 351
- Scaling of BFD, on page 351
- Associating BGP Neighbors with the Context, on page 351
- Associating OSPF Neighbors with the Context, on page 352
- Associating BFD Neighbor Groups with the BFD Protocol, on page 352
- Enabling BFD on OSPF Interfaces, on page 352
- Monitoring BFD Connection for ICSR, on page 352

Configuring a BFD Context

```
config
  context context_name
    bfd-protocol
    [ bfd echo ]
    exit
```

Notes:

- Echo function can be optionally enabled for all interfaces in this context.

- 16 BFD sessions per context and 64 per chassis.

Configuring IPv4 BFD for Static Routes

Enable BFD on an interface.

```
config
  context bfd_context_name
    interface if_name
      ip address ipv4_address ipv4_mask
      bfd interval interval_value min_rx rx_value multiplier multiplier_value
      [ bfd echo ]
    exit
```

Configure BFD static route.

```
ip route static bfd if_name ipv4_gw_address
```

Add static routes.

```
ip route ipv4_address ipv4_mask
ip route ipv4_address ipv4_mask
```

Configuring IPv6 BFD for Static Routes

Enable BFD on an Interface

```
config
  context bfd_context_name
  interface if_name
    ipv6 address ipv6_address ipv6_mask
      bfd interval interval_value min_rx rx_value multiplier multiplier_value
      [ bfd echo ]
      exit
```

Configure BFD static route.

```
ipv6 route static bfd if_name ipv6_gw_address
```

Add static routes.

```
ipv6 route ipv6_address ipv6_mask
ipv6 route ipv6_address ipv6_mask
```

Important On the ASR 5500, static routes with IPv6 prefix lengths less than /12 and between the range of /64 and /128 are not supported.

Configuring BFD for Single Hop

Enable BFD on an interface.

```
config
  context bfd_context_name
    interface if_name
      ip address ipv4_address ipv4_mask
```

```
ipv6 address ipv6_address ipv6_mask
bfd interval interval_value min_rx rx_value multiplier multiplier_value
[ bfd echo ]
exit
```

Enable BFD on a BGP Neighbor. For additional information, see Associating BGP Neighbors with the Context, on page 351.

Enable BFD on an OSPF Neighbor. For additional information, see Associating OSPF Neighbors with the Context, on page 352.

Important On the ASR 5500, routes with IPv6 prefix lengths less than /12 and between the range of /64 and /128 are not supported.

Configuring Multihop BFD

Enable BFD on an interface.

```
config
  context bfd_context_name
    interface if_name
      ip address ipv4_address ipv4_mask
      ipv6 address ipv6_address ipv6_mask
      bfd interval interval_value min_rx rx_value multiplier multiplier_value
      [ bfd echo ]
      exit
```

Configure a Multihop BFD session.

```
bfd-protocol
  bfd multihop peer destination-address interval interval-value multiplier
multiplier-value
```

Enable BFD on a BGP Neighbor. For additional information, see Associating BGP Neighbors with the Context, on page 351.

Scaling of BFD

Configure an active BFD session using one of the above methods and use same BFD neighbor while configuring the active interface. For additional information, see Associating BFD Neighbor Groups with the BFD Protocol, on page 352.

```
bfd-protocol
  bfd nbr-group-name grp_name active-if-name if_name nexthop_address
```

Apply the same BFD results to one or more passive interfaces.

```
bfd nbr-group-name grp_name passive-if-name if_name nexthop_address
bfd nbr-group-name grp_name passive-if-name if_name nexthop_address
```

Associating BGP Neighbors with the Context

```
config
  context context_name
```

```
router bgp AS_number
  neighbor neighbor_ip-address remote-as rem_AS_number
  neighbor neighbor_ip-address ebgp-multihop max-hop max_hops
  neighbor neighbor_ip-address update-source update-src_ip-address
  neighbor neighbor_ip-address failover bfd [ multihop ]
```

Notes:

- Repeat the sequence to add neighbors.

Associating OSPF Neighbors with the Context

```
config
  context context_name
    router ospf
      neighbor neighbor_ip-address
```

Notes:

- Repeat the sequence to add neighbors.

Associating BFD Neighbor Groups with the BFD Protocol

```
config
  context context_name
    bfd-protocol
      bfd nbr-group-name grp_name active-if-name if_name nexthop_address
      bfd nbr-group-name grp_name passive-if-name if_name nexthop_address
```

Enabling BFD on OSPF Interfaces

All OSPF Interfaces

```
config
  context context_name
    router ospf
      bfd-all-interfaces
```

Specific OSPF Interface

```
config
  context context_name
    interface interface_name
      broadcast
        ip ospf bfd
```

Monitoring BFD Connection for ICSR

For ICSR configurations, the following command sequence initiates monitoring of the connection between the primary chassis and the BFD neighbor in the specified context. If the connection drops, the standby chassis becomes active.

```
config
  context context_name
```

```
    service-redundancy-protocol
      monitor bfd context context_name { ipv4_address | ipv6_address } {
chassis-to-chassis | chassis-to-router }
```

Notes:

- *ipv4 _address* | *ipv6_address* defines the IP address of the BFD neighbor to be monitored, entered using IPv4 dotted-decimal or IPv6 colon-separated-hexadecimal notation
- **chassis-to-chassis** enables BFD to run between primary and backup chassis on non-SRP links.
- **chassis-to-router** enables BFD to run between chassis and router.

Saving the Configuration

Save your configuration as described in the *Verifying and Saving Your Configuration* chapter.

Chassis-to-Chassis BFD Monitoring for ICSR

An operator can configure BFD to more quickly advertise routes during an ICSR switchover. This solution complements the feature that allows the advertising of BGP routes from a Standby ICSR chassis. The overall goal is to support more aggressive failure detection and recovery in an ICSR configuration when implementing of VoLTE.

You must configure the following features for chassis-to-chassis BFD monitoring in ICSR configurations:

- Enable Primary Chassis BFD Monitoring, on page 353.
- Set BFD to Ignore ICSR Dead Interval, on page 353.
- Configure ICSR Switchover Guard Timer, on page 354.
- Enable BFD Multihop Fall-over , on page 354.
- Enable Advertising BGP Routes from Standby ICSR Chassis, on page 356.

Enable Primary Chassis BFD Monitoring

You must enable monitoring of the connection between the primary chassis and specified BFD neighbors. If the connection drops, the standby chassis becomes active. For more information, see Monitoring BFD Connection for ICSR, on page 352.

Set BFD to Ignore ICSR Dead Interval

The SRP Configuration mode **bfd-mon-ignore-dead-interval** command causes the standby ICSR chassis to ignore the dead interval and remain in the standby state until all the BFD chassis-to-chassis monitors fail.

Enable this feature in association with BFD chassis-to-chassis monitoring to support more aggressive ICSR failure detection times.

```
configure
   context context_name
      service-redundancy-protocol variable
         bfd-mon-ignore-dead-interval
         end
```

Configure ICSR Switchover Guard Timer

The SRP Configuration mode **guard timer** command configures the redundancy-guard-period and monitor-damping-period for SRP service monitoring.

Use these guard timers to ensure that local failures, such as card reboots and task restarts, do not result in ICSR events which can be disruptive.

```
configure
   context context_name
      service-redundancy-protocol variable
         guard-timer { aaa-switchover-timers { damping-period seconds |
guard-period seconds } | diameter-switchover-timers { damping-period seconds
 | guard-period seconds } | srp-redundancy-timers { aaa  { damping-period
seconds | guard-period seconds } | bgp { damping-period seconds | guard-period
 seconds } | diam  { damping-period seconds | guard-period seconds } }
         end
```

Notes:

- **aaa-switchover-timers** – sets timers that prevent back-to-back ICSR switchovers due to an AAA failure (post ICSR switchover) while the network is still converging.
  - **damping-period** – configures a delay time to trigger an ICSR switchover due to a monitoring failure within the guard-period.
  - **guard-period** – configures the local-failure-recovery network-convergence timer.
- **diameter-switchover-timers** – sets timers that prevent a back-to-back ICSR switchover due to a Diameter failure (post ICSR switchover) while the network is still converging.
  - **damping-period** – configures a delay time to trigger an ICSR switchover due to a monitoring failure within the guard-period.
  - **guard-period** – configures the local-failure-recovery network-convergence timer.
- **srp-redundancy-timers** – sets timers that prevent an ICSR switchover while the system is recovering from a local card-reboot/critical-task-restart failure.
  - **aaa** – local failure followed by AAA monitoring failure
  - **bgp** – local failure followed by BGP monitoring failure
  - **diam** – local failure followed by Diameter monitoring failure

Enable BFD Multihop Fall-over

A **fall-over bfd multihop** *mhsess_name* keyword in the Context Configuration mode ip route and ipv6 route commands enables fall-over BFD functionality for the specified multihop session. The **fall-over bfd** option uses BFD to monitor neighbor reachability and liveliness. When enabled it will tear down the session if BFD signals a failure.

```
configure
   context context_name
      ip route { ip_address/ip_mask | ip_address ip_mask } { gateway_ip_address |
next-hop next_hop_ip_address | point-to-point | tunnel } egress_intrfc_name [
cost cost ] [ fall-over bfd multihop mhsess_name ] [ precedence precedence ] [
 vrf vrf_name [ cost value ] [ fall-over bfd multihop mhsess_name ] [ precedence
```

```
 precedence ] +
      end
```

The **ip route** command now also allows you to add a static multihop BFD route.

```
      ip route static multihop bfd mhbfd_sess_name local_endpt_ipaddr
remote_endpt_ipaddr
```

Important SNMP traps are generated when BFD sessions go up and down (BFDSessUp and BFDSessDown).

ip route Command

```
configure
   context context_name
      ip route { ip_address/ip_mask | ip_address ip_mask } { gateway_ip_address |
next-hop next_hop_ip_address | point-to-point | tunnel } egress_intrfc_name [
cost cost ] [ fall-over bfd multihop mhsess_name ] [ precedence precedence ] [
 vrf vrf_name [ cost value ] [ fall-over bfd multihop mhsess_name ] [ precedence
 precedence ] +
      end
```

The **ip route** command now also allows you to add a static multihop BFD route.

```
      ip route static multihop bfd mhbfd_sess_name local_endpt_ipaddr
remote_endpt_ipaddr
```

ip routev6 Command

```
configure
   context context_name
     ipv6 route ipv6_address/prefix_length { interface name | next-hop ipv6_address
 interface name } [ cost  cost] [ fall-over bfd multihop mhsess_name ] [
precedence precedence ] [ vrf vrf_name [ cost value ] [ fall-over bfd multihop
 mhsess_name ] [ precedence precedence ]
      end
```

The **ipv6 route** command now also allows you to add a static multihop BFD route.

```
      ipv6 route static multihop bfd mhbfd_sess_name local_endpt_ipv6addr
remote_endpt_ipv6addr
```

Adjust BFD Interval

Set the transmit interval (in milliseconds) between BFD packets to meet the convergence requirements of your network deployment.

```
configure
   context context_name
      interface interface_name broadcast
         bfd interval interval_num  min_rx milliseconds multiplier value
         end
```

Notes:

- *milliseconds* is an integer from 50 through 10000. (Default 50)

Enable Advertising BGP Routes from Standby ICSR Chassis

For information on configuring the feature, see Advertising BGP Routes from a Standby ICSR Chassis , on page 343.

Saving the Configuration

Save your configuration as described in the *Verifying and Saving Your Configuration* chapter.

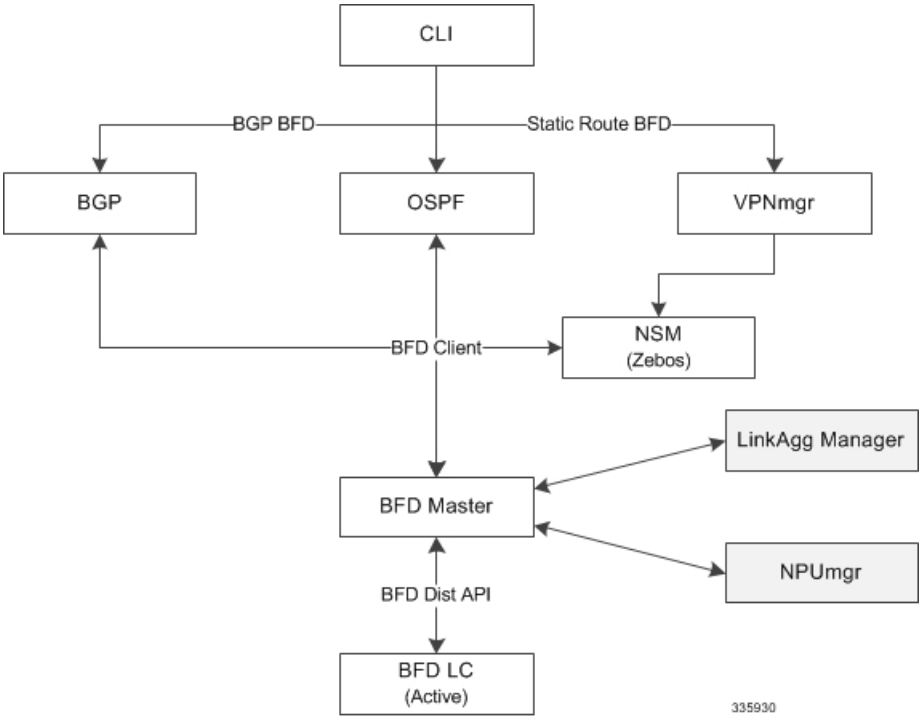
BFD Support for Link Aggregation Member Links

Member-link based BFD detects individual link failures faster than LACP and reduces the overall session/traffic down period as a result of single member link failure.

Overview

A BFD Configuration mode CLI command configures BFD interactions with the linkagg task. Once a session is configured, BFD creates per member link BFD sessions and starts sending packets on each of the linkagg member links. If a member link BFD session fails, StarOS notifies failures to the linkagg task.

Figure 7: BFD Interactions



If you define a linkagg-peer using a slot number, you may configure a linkagg-peer for a redundant LC (Line Card) slot which must also specify a slot in its member-link configuration. Likewise, if you configure a linkagg-peer without a slot, you must delete it before configuring a peer with a slot specified.

Important Only one IPv4 or IPv6 BFD session-based configuration is allowed per linkagg interface for compliance with RFC 7130.

Configuring Support for BFD Linkagg Member-links

The **bfd linkagg-peer** command enables member-link BFD and configures the BFD link aggregation (linkagg) session values [RFC 7130].

```
configure
  context context_name
    bfd-protocol
      bfd linkagg-peer linkagg_group_id local-endpt-addr local-endpt_ipaddress
remote-endpt-addr remote_endpt_ipaddress interval tx_interval min_rx rx_interval
multiplier multiplier_value [ slot slot_number ]
      no bfd linkagg-peer linkagg_group_id [ slot slot_number ]
      end
```

Notes:

- *linkagg_group_id* specifies the LAG number as an integer from 1 through 255.
- **local-endpt-addr** *local-endpt_ipaddress* specifies the source address of the multihop BFD session in IPv4 or IPv6 notation.
- **remote-endpt-addr** *remote-endpt_ipaddress* specifies the remote address of the multihop BFD session in IPv4 or IPv6 notation.
- **interval** *tx_interval* specifies the transmit interval of control packets in milliseconds as an integer from 50 through 10000.
- **min_rx** *rx_interval* specifies the receive interval of control packets in milliseconds as an integer from 50 through 10000.
- **multiplier** *multiplier_value* specifies the value used to compute hold-down time as an integer from 3 through 50.
- **slot** *slot_number* for redundant active-standby link aggregation, this option specifies the card for which this configuration is intended.

Saving the Configuration

Save your configuration as described in the *Verifying and Saving Your Configuration* chapter.

Viewing Routing Information

To view routing information for the current context, run one of the following Exec mode commands;

- **show ip route**: Displays information for IPv4 routes in the current context.
- **show ipv6 route**: Displays information for ipv6 routes in the current context.
- **show ip static-route**: Displays information only for IPv4 static routes in the current contextospf.
- **show ip ospf**: Displays IPv4 OSPF process summary information in the current context.
- **show ipv6 ospf**: Displays IPv6 OSPFv3 process summary information in the current context.

- **show ip bgp**: Displays IPv4 BGP information.

This example shows sample output of the command, **show ip route**.

```
[local]host_name# show ip route
"*" indicates the Best or Used route.

 Destination         Nexthop          Protocol   Prec Cost Interface
*44.44.44.0/24       208.230.231.50   static     1    0    local1
*192.168.82.0/24     0.0.0.0          connected  0    0
*192.168.83.0/24     0.0.0.0          connected  0    0
 208.230.231.0/24    0.0.0.0          ospf       110  10   local1
*208.230.231.0/24    0.0.0.0          connected  0    0    local1
Total route count: 5
```

CHAPTER 24

VLANs

This chapter provides information on configuring virtual local area networks (VLANs) in support of enhanced or extended services. Product-specific and feature-specific *Administration Guides* provide examples and procedures for configuration of services on the system that may utilize VLANs. You should select the configuration example that best meets your service model before using the procedures described below.

VPC-DI uses VLANs to implement signaling and bearer services within the DI network.

Overview

Virtual LANs (VLANs) provide greater flexibility in the configuration and use of contexts and services.

They are configured as "tags" on a per-port basis and allow more complex configurations to be implemented. The VLAN tag allows a single physical port to be bound to multiple logical interfaces that can be configured in different contexts. Therefore, each Ethernet port can be viewed as containing many logical ports when VLAN tags are employed.

VLANs can be created at the hypervisor and StarOS levels. Where you create the VLAN depends on your specific network requirements.

Overlapping IP Address Pool Support – GGSN

Overlapping IP Address pools allow operators to more flexibly support multiple corporate VPN customers with the same private IP address space without expensive investments in physically separate routers or virtual routers.

The system supports two types of overlapping pools:

- *Resource* pools are designed for dynamic assignment only, and use a VPN tunnel (such as a GRE tunnel) to forward and receive the private IP addresses to and from the VPN.
- *Overlap* pools can be used for both dynamic and static addressing, and use VLANs and a next hop forwarding address to connect to the VPN customer.

To forward downstream traffic to the correct PDP context, the GGSN uses either the GRE tunnel ID or the VLAN ID to match the packet. When forwarding traffic upstream, the GGSN uses the tunnel and forwarding information in the IP pool configuration; overlapping pools must be configured in the APN in such instances.

When a PDP context is created, the IP address is assigned from the IP pool. In this case the forwarding rules are also configured into the GGSN. If the address is assigned statically, when the GGSN confirms the IP address from the pool configured in the APN, the forwarding rules are also applied.

The GGSN can scale to as many actual overlapping pools as there are VLAN interfaces per context, and there can be multiple contexts per GGSN. The limit is the number of IP pools. This scalability allows operators who wish to provide VPN services to customers using the customer's private IP address space, not to be concerned about escalating hardware costs or complex configurations.

RADIUS VLAN Support – Enhanced Charging Services

VPN customers often use private address space which can easily overlap with other customers. The subscriber addresses are supported with overlapping pools which can be configured in the same virtual routing context.

RADIUS Server and NAS IP addresses do not need to be in separate contexts, thereby simplifying APN and RADIUS configuration and network design. This feature allows the following scenarios to be defined in the same context:

- Overlapping RADIUS NAS-IP addresses for various RADIUS server groups representing different APNs.
- Overlapping RADIUS server IP addresses for various RADIUS servers groups.

Every overlapping NAS-IP address is given a unique next-hop address which is then bound to an interface that is bound to a unique VLAN, thereby allowing the configuration to exist within the same context.

The system forwards RADIUS access requests and accounting messages to the next hop defined for that NAS-IP; the connected routers forward the messages to the RADIUS server. The next hop address determines the interface and VLAN to use. Traffic from the server is identified as belonging to a certain NAS-IP by the port/VLAN combination.

The number of RADIUS NAS-IP addresses that can be configured is limited by the number of loopback addresses that can be configured.

APN Support – PDN Gateway (P-GW)

P-GW Access Point Name (APN) supports extensive parameter configuration flexibility for the APN. VLAN tagging may be selected by the APN, but are configured in the P-GW independently from the APN.

VLANs and StarOS

StarOS supports VLANs for the DI network and several of its gateway products.

Note VLANs are supported in conjunction with service context SF vNIC traffic ports. StarOS supports the configuration limits for VLANs as described in the *Engineering Rules* appendix.

VLANs and Hypervisors

Depending on the type of packets being processed over the network, the hypervisor performs different VLAN tasks prior to exchanging packets with the VPC-DI virtual machine (VM).

- **Management packets** MGMT packets arrive untagged and the hypervisor exchanges these packets with the VM without additional VLAN processing.
- **Access packets** arrive from the physical network with VLAN tags. The hypervisor removes the VLAN tags before forwarding them to a VM. It retags the received packets prior to sending them out across the physical network.
- **Trunking** packets arrive and depart across the physical network with VLAN tags. The hypervisor filters the tags before sending tagged packets to the VM for additional processing.

Management, access and trunking packets should be defined in separate contexts and bound to unique interfaces. The hypervisor should be configured to provide the appropriate type of VLAN tagging or filtering based on the packet type.

Refer to the following sections for a brief description of VLAN support and sources for additional information.

- VLANs and KVM Hypervisor, on page 361
- VLANs and VMware, on page 362

VLANs and KVM Hypervisor

Network Isolation

The Ubuntu networking stack implementation allows the KVM host to act as a simple layer 2 bridge (that is, an Ethernet switch), a forwarding or NAT router, a stateful firewall, or any combination of those roles.

VLANs versus Bridged Interfaces

In the KVM virtualization scenario, VLAN usage can be seen as an extension to the simple bridge interface sharing. The difference lies in which interface participates in the bridge set. In the standard mode of operation (as seen in the examples in Network port sharing with Ethernet bridges), the physical interfaces (such as eth0, eth1...) are bound to the bridge, which is used by each guest. These interfaces carry unmodified packets coming externally or being generated internally, with or without a VLAN ID tag.

It is possible to filter out every package not carrying a particular VLAN ID by creating subinterfaces. These subinterfaces become part of the VLAN defined by a specific VLAN ID.

Applying this concept to the bridged interface sharing method involves replacing the bound physical interface by a subinterface that is part of a particular VLAN segmentation. This way, every virtual machine guest with interfaces bound to this bridge is part of that particular VLAN. Like in the simple Ethernet bridge environment, the network provided is transparent.

Note Not all vNIC types support VLAN trunking into a bridge, as many filter out VLANs in hardware.

Additional Information

For additional information on configuring VLANs with the KVM hypervisor see the URLs below:

- *Configuring 802.1q VLANs:* https://www.ibm.com/support/knowledgecenter/linuxonibm/liaat/liaatkvmsecconfvlans.htm
- *KVM/Networking:* https://help.ubuntu.com/community/KVM/Networking

VLANs and VMware

VMware supports the configuration of VLANs to meet network deployment requirements.

VLAN Configuration

VLANs enable a single physical LAN segment to be further segmented so that groups of ports are isolated from one another as if they were on physically different segments.

Configuring ESXi with VLANs is recommended for the following reasons:

- It integrates the host into a pre-existing environment.
- It integrates the host into a pre-existing environment.
- It reduces network traffic congestion.
- iSCSI traffic requires an isolated network.

You can configure VLANs in ESXi using three methods: External Switch Tagging (EST), Virtual Switch Tagging (VST), and Virtual Guest Tagging (VGT).

- With EST, all VLAN tagging of packets is performed on the physical switch. Host network adapters are connected to access ports on the physical switch. Port groups that are connected to the virtual switch must have their VLAN ID set to 0.

With VST, all VLAN tagging of packets is performed by the virtual switch before leaving the host. Host network adapters must be connected to trunk ports on the physical switch. Port groups that are connected to the virtual switch must have an appropriate VLAN ID specified.

With VGT, all VLAN tagging is performed by the virtual machine. For VGT the VLAN ID = 4095. VLAN tags are preserved between the virtual machine networking stack and external switch when frames are passed to and from virtual switches. Physical switch ports are set to trunk port.

Additional Information

For additional information on configuring VLANs with the VMware hypervisor see the documents below:

- Configuring VLANs on UCS and VMware
- For information about VLAN Configuration, refer to the VMware documentation.
- For information about assigning a VLAN ID to an ESXi Host, refer to the VMware documentation
- For information about VLAN configuration on virtual switches, physical switches, and virtual machines, refer to the VMware documentation

Creating VLAN Tags

Use the following example to create VLANs on a port and bind them to pre-existing interfaces. For information on creating interfaces, refer to *System Interfaces and Ports*.

```
config
  port ethernet slot/port
    no shutdown
    vlan vlan_tag_ID
    no shutdown
    bind interface interface_name context_name
    end
```

Notes:

- *Optional:* Configure VLAN-subscriber associations. Refer to Configuring Subscriber VLAN Associations, on page 364 for more information.
- Repeat this procedure as needed to configure additional VLANs for the port.
- Refer to VLAN-Related CLI Commands , on page 365 and the *Command Line Interface Reference* for additional information.
- Save your configuration as described in the *Verifying and Saving Your Configuration* chapter.

Verifying the Port Configuration

Run the following command to verify the port configuration:

```
[local]host_name# show port info slot/port
```

An example of this command's output when at least one VLAN has been configured for the port is shown below:

```
Port: 5/11
  Port Type               : 10G Ethernet
  Role                    : Service Port
  Description             : (None Set)
  Redundancy Mode         : Port Mode
  Redundant With          : 6/11
  Preferred Port          : Non-Revertive
  Physical ifIndex        : 85262336
  Administrative State    : Enabled
  Configured Duplex       : Auto
  Configured Speed        : Auto
  Fault Unidirection Mode : 802_3ae clause 46
```

```
  Configured Flow Control : Enabled
  Interface MAC Address   : 64-9E-F3-69-5B-EA
  SRP Virtual MAC Address : None
  Fixed MAC Address       : 64-9E-F3-69-5B-CA
  Link State              : Up
  Link Duplex             : Full
  Link Speed              : 10 Gb
  Flow Control            : Enabled
  Link Aggregation Group  : None
  Untagged:
    Logical ifIndex       : 85262337
    Operational State     : Up, Active
  Tagged VLAN: VID 10
    Logical ifIndex       : 285278210
    VLAN Type             : Standard
    VLAN Priority         : 0
    Administrative State  : Enabled
    Operational State     : Up, Active
  Number of VLANs         : 1
  SFP Module              : Present (10G Base-SR)
```

Notes:

- Repeat this sequence as needed to verify additional ports.
- *Optional:* Configure VLAN-subscriber associations. Refer to Configuring Subscriber VLAN Associations, on page 364 for more information.
- Refer to VLAN-Related CLI Commands , on page 365 for additional information.
- Save your configuration as described in the *Verifying and Saving Your Configuration* chapter.

Configuring Subscriber VLAN Associations

Subscriber traffic can be routed to specific VLANs based on the configuration of their user profile. This functionality provides a mechanism for routing all traffic from a subscriber over the specified VLAN. All packets destined for the subscriber must also be sent using only IP addresses valid on the VLAN or they will be dropped.

RADIUS Attributes Used

The following RADIUS attributes can be configured within subscriber profiles on the RADIUS server to allow the association of a specific VLAN to the subscriber:

- **SN-Assigned-VLAN-ID:** In the Starent VSA dictionary
- **SN1-Assigned-VLAN-ID:** In the Starent VSA1 dictionary

Important Since the instructions for configuring subscriber profiles differ between RADIUS server applications, this section only describes the individual attributes that can be added to the subscriber profile. Please refer to the documentation that shipped with your RADIUS server for instructions on configuring subscribers.

Configuring Local Subscriber Profiles

Use the configuration example below to configure VLAN associations within local subscriber profiles on the system.

Important These instructions assume that you have already configured subscriber-type VLAN tags according to the instructions provided in Creating VLAN Tags, on page 363.

```
config
  context context_name
    subscriber name user_name
      ip vlan vlan_id
      end
```

Verify the Subscriber Profile Configuration

Use the following command to view the configuration for a subscriber profile:

```
[local]host_name# show subscriber configuration username user_name
```

Notes:

- Repeat this command for each subscriber.
- Save your configuration as described in the *Verifying and Saving Your Configuration* chapter.

VLAN-Related CLI Commands

VLAN-related features and functions are supported across several CLI command modes. The following tables identify commands associated with configuration and monitoring of VLAN-related functions.

For detailed information regarding the use of the commands listed below, see the *Command Line Interface Reference*.

Table 24: VLAN-Related Configuration Commands

CLI Mode	Command	Description
AAA Server Group Configuration Mode	**radius attribute nas-ip-address address** *ip_address* **nexthop-forwarding-address** *ip_address* **vlan** *vlan_id*	Sets the RADIUS client to provide the VLAN ID with the nexthop forwarding address to a system when running in single nexthop gateway mode. **Note:** To access the **vlan** keyword, **aaa-large configuration** must be enabled via the Global Configuration mode.
ACS Charging Action Configuration Mode	**ip vlan** *vlan_id*	Configures the VLAN identifier to be associated with the subscriber traffic in the destination context.

CLI Mode	Command	Description
Context Configuration Mode	**ip pool** *pool_name* **nexthop forwarding address** *ip_address* **overlap vlanid** *vlan_id*	When a nexthop forwarding address is configured, the **overlap vlanid** keyword enables support for overlapping IP address pools and associates the pool with the specified VLAN ID.
Context Configuration Mode	**ip routing overlap-pool**	Advertises overlap-pool addresses in dynamic routing protocols when overlap pools are configured using VLAN IDs. When enabled, the overlap addresses are added as interface addresses and advertised.
Context Configuration Mode	**radius attribute nas-ip-address address** *ip_address* **nexthop-forwarding-address** *ip_address* **vlan** *vlan_id*	Specifies the VLAN ID to be associated with the next-hop IP address.
Ethernet Interface Configuration Mode	**[no] logical-port-statistics**	Enables or disables the collection of logical port (VLAN and NPU) bulk statistics for the first 32 configured Ethernet or PVC interface types.
Ethernet Interface Configuration Mode	**vlan-map next-hop** *ipv4_address*	Sets a single next-hop IP address so that multiple VLANs can use a single next-hop gateway. The vlan-map is associated with a specific interface.
Ethernet Port Configuration Mode	**vlan** *vlan_id*	Enters VLAN Configuration mode.
PVC Configuration Mode	**[no] shutdown**	Enables or disables traffic over a specified VLAN. See below.
Subscriber Configuration Mode	**ip vlan** *vlan_id*	Configures the subscriber VLAN ID that is used with the assigned address for the subscriber session to receive packets. If the IP pool from which the address is assigned is configured with a VLAN ID, this subscriber configured VLAN ID overrides it.
VLAN Configuration Mode	**bind interface** *interface_name context_name*	Binds a virtual interface and context to support VLAN service.
VLAN Configuration Mode	**[no] ingress-mode**	Enables or disables port ingress incoming) mode.
VLAN Configuration Mode	**priority** *value*	Configures an 802.1p VLAN priority bit for ASN-GW service only.
VLAN Configuration Mode	**[no] shutdown**	Enables or disables traffic over the current VLAN.

CLI Mode	Command	Description
VLAN Configuration Mode	**vlan-map interface** *if_name context_name*	Associates an IP interface having a VLAN ID with a context.

Table 25: VLAN-Related Monitoring Commands

CLI Mode	Command	Description
Exec Mode show commands	**clear port** *slot/port* **vlan** *vlan_id*	Clears NPU statistics for the port that has a previously configured VLAN ID.
Exec Mode show commands	**show logical-port utilization table vlan { 5-minute | hourly }**	Displays VLAN utilization for a specified collection interval.
Exec Mode show commands	**show port info** *slot/port* **vlan** *vlan_id*	Displays NPU counters for a previously configured VLAN ID.

CHAPTER 25

BGP MPLS VPNs

This chapter describes services that are supported for Border Gateway Protocol (BGP) Multi-Protocol Label Switching (MPLS) Virtual Private Networks (VPNs).

Important MPLS is a licensed Cisco feature that requires a separate license. Contact your Cisco account representative for detailed information on specific licensing requirements. For information on installing and verifying licenses, refer to the *Managing License Keys* section of *Software Management Operations*.

It includes the following topics:

Introduction

Service providers require the ability to support a large number of corporate Access Point Names (APNs) which have a number of different addressing models and requirements. VPC-DI uses BGP MPLS Layer 3 VPNs to segregate corporate customer APNs in a highly scalable manner. This solution conforms to RFC 4364 – *BGP/MPLS IP Virtual Private Networks (VPNs)*.

Note Option b is supported for connectivity between multi-AS backbones.

The BGP/MPLS solution supports the following scenarios:

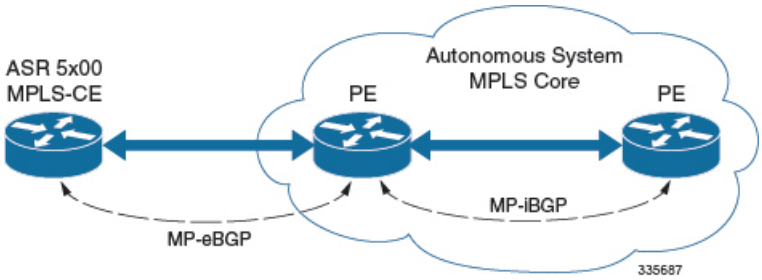
- MPLS-CE Connected to PE, on page 370
- VPC-DI as a PE, on page 370

VPC-DI also supports VPNv6 as described in RFC 4659 – *BGP-MPLS IP Virtual Private Network (VPN) Extension for IPv6 VPN*. See IPv6 Support for BGP MPLS VPNs, on page 372 for details.

MPLS-CE Connected to PE

In this scenario the VPC-DI functions as an MPLS-CE (Customer Edge) network element connected to a Provider Edge (PE) Label Edge Router (LER), which in turn connects to the MPLS core (RFC 4364). See the figure below.

Figure 8: VPC-DI MPLS-CE to PE



The MPLS-CE functions like a PE router within its own Autonomous System (AS). It maintains Virtual Routing and Forwarding (VRF) routes and exchanges VPN route information with the PE via an MP-eBGP (Multi-Protocol-external BGP) session.

The PE is also configured with VRFs and exchanges VPN routes with other PEs in its AS via MP-iBGP (Multi-Protocol-internal BGP) connections and the MPLS-CE via an MP-eBGP connection.

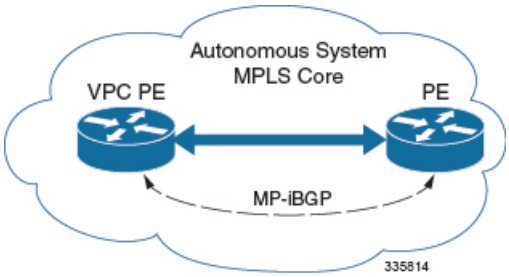
The EBGP connection allows the PE to change next-hop IP addresses and labels in the routes learned from IBGP peers before advertising them to the MPLS-CE. The MPLS-CE in this case uses only MP-eBGP to advertise and learn routes. Label Distribution Protocol (LDP) and Resource Reservation Protocol (RSVP) are not required because of direct-connect EBGP peering. The MPLS-CE in this scenario pushes/pops a single label (learned over the MP-eBGP connection) to/from the PE.

VPC-DI as a PE

Overview

In this scenario, the VPC-DI functions as a PE router sitting at the edge of the MPLS core. See the figure below.

Figure 9: VPC-DI as a PE



The VPC-DI eliminates the need for an ASBR or PE as shown in the first two scenarios. In this scenario, two main requirements are introduced: IBGP functionality and MPLS label distribution protocols.

The VPC-DI can be configured to add two labels:

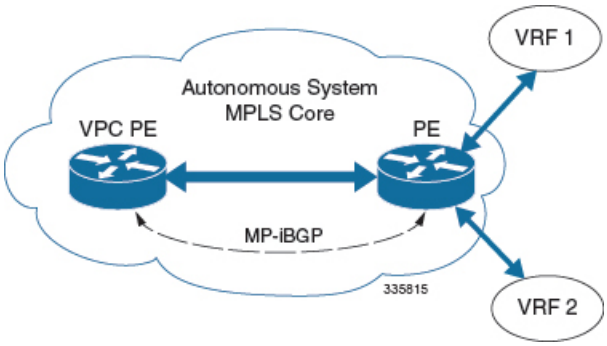
- an outer label learned from LDP or RSVP-TE (RSVP-Traffic Engineering)
- an inner label learned from MP-iBGP

This solution supports traffic engineering and QoS initiated via the VPC-DI.

Sample Configuration

In this example, VRFs are configured on the ASR 5500 PE and pools are associated with VRFs. The VPC-DI exchanges VPN routes with its IBGP peers (PE routers) and learns the MPLS paths to reach PEs via LDP. The VPC-DI forwards the packets to the next-hop with two labels – an inner label learned from PE and an outer label learned from the next hop IBGP neighbor.

Figure 10: Sample Configuration




```
mpls ip
  protocol ldp
    enable
  exit
exit

ip vrf vrf1
  mpls traffic-class copy
exit
ip vrf vrf2
  mpls traffic-class value 5
exit

router bgp 300
  ip vrf vrf1
    route-target export 300 1
      route-target import 300 1
      route-distinguisher 300 1
  exit
    ip vrf vrf2
      route-target export 300 2
        route-target import 300 2
        route-distinguisher 300 2
  exit

  router-id 2.2.2.2
  neighbor 192.168.107.20 remote-as 300
```

```
  neighbor 192.168.107.20 update-source node1_loopback

  address-family vpnv4
    neighbor 192.168.107.20 activate
      neighbor 192.168.107.20 send-community both
      neighbor 192.168.107.20 next-hop-self
  exit

  address-family ipv4 vrf vrf1
    redistribute connected
  exit

  address-family ipv4 vrf vrf2
    redistribute connected
exit

interface interface_to_internet
  ip address 192.168.109.65/24
  mpls ip
exit
router ospf
  network 192.168.109.0/24 area 0.0.0.0
exit
```

IPv6 Support for BGP MPLS VPNs

Overview

The VPC-DI supports VPNv6 as described in RFC 4659 – *BGP-MPLS IP Virtual Private Network (VPN) Extension for IPv6 VPN*.

An IPv6 VPN is connected over an IPv6 interface or sub-interface to the Service Provider (SP) backbone via a PE router. The site can be both IPv4 and IPv6 capable. Each VPNv6 has its own address space which means a given address denotes different systems in different VPNs. This is achieved via a VPNv6 address-family which prepends a Route Distinguisher (RD) to the IP address.

A VPNv6 address is a 24-byte quantity beginning with an 8-byte RD and ending with a 16-byte IPv6 address. When a site is IPv4 and IPv6 capable, the same RD can be used for the advertisement of both IPv4 and IPv6 addresses.

The system appends RD to IPv6 routes and exchanges the labeled IPv6-RD using the VPNv6 address-family. The Address Family Identifier (AFI) and Subsequent Address Family Identifier (SAFI) fields for VPNv6 routes will be set to 2 and 128 respectively.

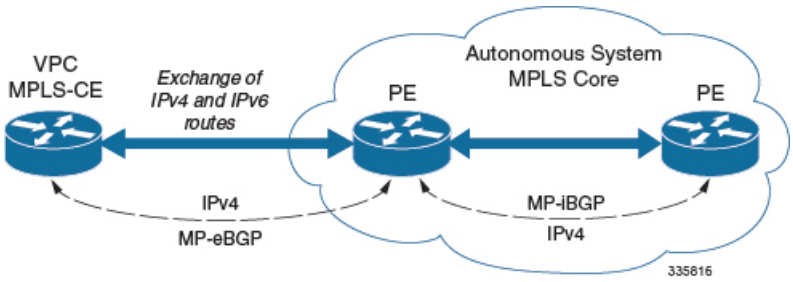
The IPv6 VPN traffic will be transported to the BGP speaker via IPv4 tunneling. The BGP speaker advertises to its peer a Next Hop Network Address field containing a VPN-IPv6 address whose 8-octet RD is set to zero and whose 16-octet IPv6 address is encoded as an IPv4-mapped IPv6 address (RFC 4291) containing the IPv4 address of the advertising router. It is assumed that only EBGP peering will be used to exchange VPNv6 routes.

Support for VPN-IPv6 assumes the following:

- Dual Stack (IPv4/IPv6) routing
- IPv6 pools in VRFs
- BGP peering over a directly connected IPv4 interface

See the figure below.

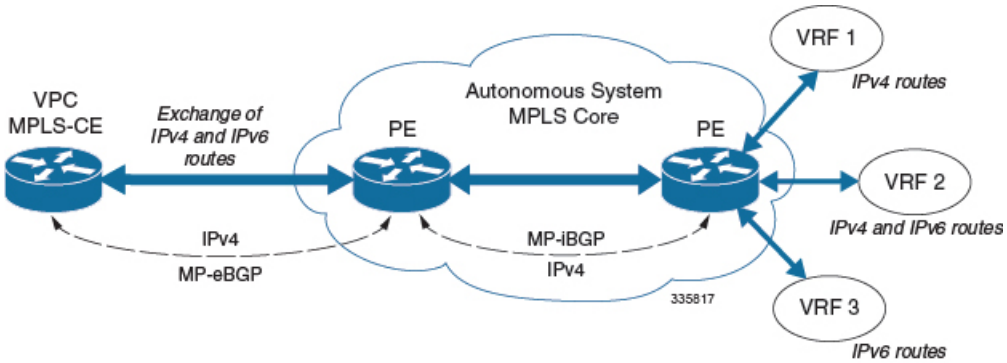
Figure 11: IPv6-RD Support for VPNv6



Sample Configuration

This example assumes three VRFs. VRF 1 has only IPv4 routes, VRF f2 has both IPv4 and IPv6 routes, and VRF 3 has only IPv6 routes.

Figure 12: VPNv6 Sample Configuration



Configure VRFs.

```
ip vrf vrf1
exit
ip vrf vrf2
exit
ip vrf vrf3
exit
```

Enable MPLS BGP forwarding.

```
mpls bgp forwarding
```

Configure pools.

```
ip pool vrf1-pool 51.52.53.0 255.255.255.0 private 0 vrf vrf1
exit
ip pool vrf2-pool 51.52.53.0 255.255.255.0 private 0 vrf vrf2

exit
ipv6 pool vrf2-v6pool prefix 2005:0101::/32 private 0 vrf vrf2
exit
ipv6 pool vrf3-v6pool prefix 2005:0101::/32 private 0 vrf vrf3
exit
```

Configure interfaces.

```
interface ce_interface_to_rtr
  ip address 192.168.110.90 255.255.255.0
exit
interface ce_v6_interface
  ip address 2009:0101:0101:0101::1/96
exit
interface ce_loopback loopback
  ip address 52.1.2.3 255.255.255.255
exit
interface vrf1-loop loopback
  ip vrf forwarding vrf1
  ip address 1.52.53.54 255.255.255.255
exit
interface vrf2-loop loopback
  ip vrf forwarding vrf2
  ip address 2.52.53.54 255.255.255.255
exit
interface vrf2-v6loop loopback
  ip vrf forwarding vrf2
  ip address 2005:0202:0101::1/128
exit
interface vrf3-v6loop loopback
  ip vrf forwarding vrf3
  ip address 2005:0303:0101::1/128
exit
```

Configure BGP along with address families and redistribution rules.

```
router bgp 800
  router-id 1.1.1.1
neighbor 192.168.110.20 remote-as 1003
  neighbor 192.168.110.20 activate
address-family vpnv4
  neighbor 192.168.110.20 activate
  neighbor 192.168.110.20 send-community both
exit
address-family vpnv6
  neighbor 192.168.110.20 activate
  neighbor 192.168.110.20 send-community both
exit
ip vrf vrf1
  route-distinguisher 800 1
  route-target export 800 1
  route-target import 800 1
exit
address-family ipv4 vrf vrf1
  redistribute connected
  redistribute static
exit
ip vrf vrf2
  route-distinguisher 800 2
  route-target export 800 2
  route-target import 800 2
exit
address-family ipv4 vrf vrf2
  redistribute connected
  redistribute static
exit
address-family ipv6 vrf vrf2
  redistribute connected
  redistribute static
exit
ip vrf vrf3
  route-distinguisher 800 3
  route-target export 800 3
```

```
  route-target import 800 3
exit
address-family ipv6 vrf   vrf3
  redistribute connected
  redistribute static
exit
```

Configure APNs.

```
apn walmart51.com
  selection-mode sent-by-ms
  accounting-mode none
  aaa group walmart-group
  authentication pap 1 chap 2 allow-noauth
  ip context-name Gi_ce
  ip address pool name vrf1-pool
exit
apn amazon51.com
  selection-mode sent-by-ms
  accounting-mode none
  aaa group amazon-group
  authentication pap 1 chap 2 allow-noauth
  ip context-name Gi_ce
  ip address pool name vrf2-pool
  ipv6 address prefix-pool vrf2-v6pool
exit
apn apple51.com
  selection-mode sent-by-ms
  accounting-mode none
  aaa group apple-group
  authentication pap 1 chap 2 allow-noauthip context-name Gi_ce
  ipv6 address prefix-pool vrf3-v6pool
exit
aaa-group amazon-group
  radius ip vrf vrf2
aaa group default
exit
gtpp group default
exit
ip igmp profile default
exit
```

Bind physical interfaces with the port.

VPN-Related CLI Commands

VPN-related features and functions are supported across several CLI command modes. The following tables identify commands associated with configuration and monitoring of VPN-related functions.

For detailed information regarding the use of the commands listed below, see the *Command Line Interface Reference*.

Table 26: VPN-Related Configuration Commands

CLI Mode	Command	Description
BGP Address-Family (IPv4/IPv6) Configuration Mode	**neighbor** *ip_address* **activate**	Enables the exchange of routing information with a peer router.

CLI Mode	Command	Description
BGP Address-Family (IPv4/IPv6) Configuration Mode	**neighbor** *ip_address* **send community { both | extended | standard }**	Sends the community attributes to a peer router (neighbor).
BGP Address-Family (IPv4/IPv6) Configuration Mode	**redistribute connected**	Redistributes routes into BGP from another protocol as BGP neighbors.
BGP Address-Family (VPNv4) Configuration Mode	**neighbor** *ip_address* **activate**	Enables the exchange of routing information with a peer router.
BGP Address-Family (VPNv4) Configuration Mode	**neighbor** *ip_address* **send community { both | extended | standard }**	Sends the extended-community attribute to a peer router. In VPN, route-distinguisher and route-target are encoded in the BGP extended-community. This command enables sending of BGP routes with extended community to a neighbor.
BGP Address-Family (VRF) Configuration Mode	**neighbor** *ip_address* **activate**	Enables the exchange of routing information with a peer router.
BGP Address-Family (VRF) Configuration Mode	**neighbor** *ip_address* **send community { both | extended | standard }**	Sends the extended-community attribute to a peer router. In VPN, route-distinguisher and route-target are encoded in the BGP extended-community. This command enables sending of BGP routes with extended community to a neighbor.
BGP Address-Family (VRF) Configuration Mode	**redistribute connected**	Redistributes routes into BGP from another protocol as BGP neighbors.
BGP Configuration Mode	**address-family { ipv4 vrf** *vrf_name* **| vpnv4 }**	Enables the exchange of IPv4 VRF routing information. There is a different mode for each address-family.
BGP Configuration Mode	**address-family { ipv6 vrf** *vrf_name* **| vpnv6 }**	Configures a VPNv6 address family and IPv6 VRF routing in BGP.
BGP Configuration Mode	**ip vrf** *vrf_name*	Adds a VRF to BGP and switches to the VRF Configuration mode to allow configuration of BGP attributes for the VRF.
BGP IP VRF Configuration Mode	**route-distinguisher {** *as_value* **|** *ip_address* **}** *rd_value*	Assigns a Route Distinguisher (RD) for the VRF. The RD value must be a unique value on the router for each VRF.

CLI Mode	Command	Description
BGP IP VRF Configuration Mode	**route-target { both | import | export } {** *as_value* | *ip_address* **}** *rt_value*	Adds a list of import and export route-target extended communities to the VRF.
Context Configuration Mode	**ip pool** *pool_name addr_range* **vrf** *vrf_name* **[mpls-label input** *inlabel1* **output** *outlabel1 outlabel2* **]**	Configures a pool into the specified VRF. This parameter must be specified with the Next-Hop parameter. *inlabel1* is the MPLS label that identifies inbound traffic destined for this pool. *outlabel1* and *outlabel2* specify the MPLS labels to be added to packets sent for subscribers from this pool.
Context Configuration Mode	**ip vrf** *vrf_name*	Creates a VRF and assigns a VRF-ID. A VRF is created in the router.
Context Configuration Mode	**ipv6 pool** *pool_name* **vrf** *vrf_name*	Associates the pool with that VRF. **Note:** By default the configured ipv6 pool will be associated with the global routing domain.
Context Configuration Mode	**mpls bgp forwarding**	Globally enables MPLS Border Gateway Protocol (BGP) forwarding.
Context Configuration Mode	**mpls exp** *value*	Sets the default behavior as Best Effort using a zero value in the 3-bit MPLS EXP header. This value applies to all the VRFs in the context. The default behavior is to copy the DSCP value of mobile subscriber traffic to the EXP header, if there is no explicit configuration for DSCP to EXP (via the **mpls map-dscp-to-exp dscp** *n* **exp** *m* command). **mpls exp** disables the default behavior and sets the EXP value to the configured *value*.
Context Configuration Mode	**mpls ip**	Globally enables the MPLS forwarding of IPv4 packets along normally routed paths.

CLI Mode	Command	Description
Context Configuration Mode	**radius change-authorize-nas-ip ip_address** *ip_address* **{ encrypted | key }** *value* **port** *port_num* **mpls input** *inlabel* **output** *outlabel1 outlabel2*	Configures COA traffic to use the specified MPLS labels. *inlabel* identifies inbound COA traffic. *outlabel1* and *outlabel2* specify the MPLS labels to be added to the COA response. *outlabel1* is the inner output label; *outlabel2* is the outer output label.
Ethernet Interface Configuration Mode	**mpls ip**	Enables dynamic MPLS forwarding of IP packets on this interface.
Exec Mode	**clear ip bgp peer**	Clears BGP sessions.
Exec Mode	**lsp-ping** *ip_prefix_FEC*	Checks MPLS Label-Switched Path (LSP) connectivity for the specified forwarding equivalence class (FEC). It must be followed by an IPv4 or IPv6 FEC prefix.
Exec Mode	**lsp-traceroute** *ip_prefix_FEC*	Discovers MPLS LSP routes that packets actually take when traveling to their destinations. It must be followed by an IPv4 or IPv6 FEC prefix.
IP VRF Context Configuration Mode	**mpls map-dscp-to-exp dscp** *dscp_bit_value* **exp** *exp_bit_value*	Maps the final differentiated services code point (DSCP) bit value in the IP packet header to the final Experimental (EXP) bit value in the MPLS header for incoming traffic.
IP VRF Context Configuration Mode	**mpls map-exp-to-dscp exp** *exp_bit_value* **dscp** *dscp_bit_value*	Maps the incoming EXP bit value in the MPLS header to the internal DSCP bit value in IP packet headers for outgoing traffic.
MPLS-IP Configuration Mode	**protocol ldp**	Creates the MPLS protocol family configuration modes, or configures an existing protocol and enters the MPLS-LDP Configuration Mode in the current context. This command configures the protocol parameters for the MPLS protocol family.
MPLS-LDP Configuration Mode	**advertise-labels { explicit-null | implicit-null }**	Configure advertisement of Implicit NULL or Explicit NULL label for all the prefixes advertised by the system in this context.

CLI Mode	Command	Description
MPLS-LDP Configuration Mode	**discovery { hello { hello-interval** *seconds* **| hold-interval** *seconds* **} | transport-address** *ip_address* **}**	Configures the Label Distribution Protocol (LDP) neighbor discovery parameters.
MPLS-LDP Configuration Mode	**enable**	Enables Label Distribution Protocol (LDP).
MPLS-LDP Configuration Mode	**router-id** *ip_address*	Configures the LDP Router ID.
MPLS-LDP Configuration Mode	**session timers { hold-interval** *seconds* **| keepalive-interval** *seconds* **}**	Configures the LDP session parameters.

Table 27: VPN-Related Monitoring Commands

CLI Mode	Command	Description
Exec Mode show Commands	**show ip bgp neighbors**	Displays information regarding BGP neighbors.
Exec Mode show Commands	**show ip bgp vpnv4 { all | route-distinguisher | vrf }**	Displays all VPNv4 routing data, routing data for a VRF or a route-distinguisher.
Exec Mode show Commands	**show ip bgp vpnv6**	Displays contents of VPNv6 routing table.
Exec Mode show Commands	**show ip bgp vpnv6 { all | route-distinguisher | vrf }**	Displays all VPNv6 routing data, routing data for a VRF or a route-distinguisher.
Exec Mode show Commands	**show ip pool**	Displays pool details including the configured VRF.
Exec Mode show Commands	**show mpls cross-connect**	Displays MPLS cross-connect information. MPLS tunnel cross-connects between interfaces and Label-Switched Paths (LSPs) connect two distant interface circuits of the same type via MPLS tunnels that use LSPs as the conduit.
Exec Mode show Commands	**show mpls ftn [vrf** *vrf_name*	Displays MPLS FEC-to-NHLFE (FTN) table information.
Exec Mode show Commands	**show mpls ftn [vrf** *vrf_name* **]**	Displays contents of the MPLS FTN table for a specified VRF.
Exec Mode show Commands	**show mpls ilm**	Displays MPLS Incoming Label Map (ILM) table information.

CLI Mode	Command	Description
Exec Mode show Commands	**show mpls ldp**	Displays the MPLS LDP information.
Exec Mode show Commands	**show mpls nexthop-label-forwarding-entry**	Displays MPLS Next-Hop Label Forwarding Entry (NHLFE) table information.

CHAPTER 26

Content Service Steering

This chapter provides information on configuring Content Service Steering (CSS). The product administration guides provide examples and procedures for configuration of basic services on the system. You should select the configuration example that best meets your service model, and configure the required elements for that model as described in the respective product administration guide, before using the procedures described below.

Important Internal CSS is a generic feature, if an ECSv2 license is installed on your system, internal CSS can be enabled. A separate license is not required to enable internal CSS. Contact your local Cisco account representative for information on how to obtain a license.

This chapter contains the following topics:

Overview

Content Server Selection (CSS) is a StarOS function that defines how traffic will be handled based on the "content" of the data presented by a mobile subscriber (or to a mobile subscriber). CSS is a broad term that includes features such as load balancing, NAT, HTTP redirection, and DNS redirection.

The content server (services) can be either external to the platform or integrated inside the platform.

CSS uses Access Control Lists (ACLs) to redirect subscriber traffic flows. ACLs control the flow of packets into and out of the system. ACLs consist of "rules" (ACL rules) or filters that control the action taken on packets matching the filter criteria.

ACLs are configurable on a per-context basis and applies to a subscriber through either a subscriber profile (or an APN profile in the destination context. For additional information, refer to the *Access Control Lists* chapter.

Configuring Internal Content Service Steering

To configure and activate a single CSS service for redirecting all of a subscriber's IP traffic to an internal in-line service:

Step 1 Define an IP ACL as described in Defining IP Access Lists for Internal CSS, on page 382 .

Step 2 *Optional:* Apply an ACL to an individual subscriber as described in Applying an ACL to an Individual Subscriber (Optional), on page 383.

Step 3 *Optional:* Apply a single ACL to multiple subscribers as described inApplying an ACL to Multiple Subscribers (Optional), on page 383 .

Step 4 *Optional:* Apply an ACL to multiple subscribers via APNs as described in Applying an ACL to Multiple Subscriber via APNs, on page 322.

Step 5 Save your configuration to flash memory, an external memory device, and/or a network location using the Exec mode command save configuration. For additional information on how to verify and save configuration files, refer to the *System Administration Guide* and the *Command Line Interface Reference*.

Commands used in the configuration examples in this section provide base functionality to the extent that the most common or likely commands and/or keyword options are presented. In many cases, other optional commands and/or keyword options are available. Refer to the *Command Line Interface Reference* for complete information regarding all commands. Not all commands or keywords/variables may be supported or available. Availability varies on the platform type and installed license(s).

Defining IP Access Lists for Internal CSS

IP ACLs specify what type of subscriber traffic and which direction (uplink, downlink, or both) traffic is redirected. The IP ACL must be specified in the context in which subscriber authentication is performed.

Caution To minimize the risk of data loss, do not make configuration changes to ACLs while the system is facilitating subscriber sessions.

Use the following configuration example to define an IP ACL for internal CSS; start in the Exec mode of the CLI:

```
configure
  context context_name
    ip access-list acl_name
      redirect css service service_name keywords options
      end
```

Notes:

- *service_name* must be an ACL service name.
- For information on the keywords and options available with the **redirect css service** command, see the *ACL Configuration Mode Commands* chapter in the *Command Line Interface Reference*.
- For IPv6 ACLs, the same configurations must be done in the IPv6 ACL Configuration Mode. See the *IPv6 ACL Configuration Mode Commands* chapter in the *Command Line Interface Reference*.

Applying an ACL to an Individual Subscriber (Optional)

For information on how to apply an ACL to an individual subscriber, refer to the *Applying an ACL to an Individual Subscriber* section of the *Access Control Lists* chapter.

Applying an ACL to Multiple Subscribers (Optional)

IP ACLs are applied to subscribers via attributes in their profiles. The subscriber profile can be configured locally on the system or remotely on a RADIUS server.

The system provides for the configuration of subscriber functions that serve as default values when specific attributes are not contained in the individual subscriber's profile. When configured properly, the functions can be used to apply an ACL to:

- All subscribers facilitated within a specific context by applying the ACL to the profile of the subscriber named *default*.
- All subscribers facilitated by specific services by applying the ACL to a subscriber profile and then using the **default subscriber** command to configure the service to use that subscriber as the "default" profile.

Applying an ACL to the Subscriber Named default (Optional)

For information on how to apply an ACL to the default subscriber, refer to the *Applying an ACL to the Subscriber Named default* section in the *Access Control Lists* chapter.

Applying an ACL to Service-specified Default Subscribers (Optional)

For information on how to apply an ACL to the subscriber to be used as the "default" profile by various system services, refer to the *Applying an ACL to Service-specified Default Subscribers* section in the *Access Control Lists* chapter.

Applying an ACL to Multiple Subscribers via APNs (Optional)

IP ACLs are applied to subscribers via attributes in their profiles. The subscriber profile can be configured locally on the system or remotely on a RADIUS server.

To reduce configuration time, ACLs can alternatively be applied to APN templates. When configured, any subscriber packets facilitated by the APN template would then have the associated ACL applied.

For information on how to apply an ACL to multiple subscribers via APNs, refer to the *Applying a Single ACL to Multiple Subscribers via APNs* section in the *Access Control Lists* chapter.

CHAPTER 27

Session Recovery

With robust hardware failover and redundancy protection, any hardware or software failures on the system can quickly be corrected. However, software failures can occur for numerous reasons, often without prior indication.

This chapter describes the Session Recovery feature that provides seamless failover and reconstruction of subscriber session information in the event of a hardware or software fault.

Important Session Recovery is a licensed Cisco feature. A separate feature license may be required. Contact your Cisco account representative for detailed information on specific licensing requirements. For information on installing and verifying licenses, refer to the *Managing License Keys* section of *Software Management Operations*.

This chapter includes the following sections:

How Session Recovery Works

This section provides an overview of how this feature is implemented and the recovery process.

The Session Recovery feature provides seamless failover and reconstruction of subscriber session information in the event of a hardware or software fault within the system preventing a fully connected user session from being disconnected.

Session recovery is performed by mirroring key software processes (for example, session manager and AAA manager) within the system. These mirrored processes remain in an idle state (standby-mode) wherein they perform no processing, until they may be needed in the event of a software failure (for example, a session manager task aborts).

There are some situations wherein session recovery may not operate properly. These include:

- Additional software or hardware failures occur during the session recovery operation. For example, an AAA manager fails while the state information it contained was being used to populate the newly activated session manager task.
- A lack of hardware resources (packet processing card memory and control processors) to support session recovery.

Important After a session recovery operation, some statistics, such as those collected and maintained on a per manager basis (AAA Manager, Session Manager, etc.) are in general not recovered, only accounting and billing related information is checkpointed and recovered.

Session Recovery is available for the following functions:

- Any session needing L2TP LAC support (excluding regenerated PPP on top of an HA or GGSN session)
- ASR 5500 only – Closed RP PDSN services supporting simple IP, Mobile IP, and Proxy Mobile IP
- ASR 5500 only – eHRPD service (evolved High Rate Packet Data)
- ASR 5500 only – ePDG service (evolved Packet Data Gateway)
- GGSN services for IPv4 and PPP PDP contexts
- HA services supporting Mobile IP and/or Proxy Mobile IP session types with or without per-user Layer 3 tunnels
- ASR 5500 only – HNB-GW: HNB Session over IuH
- ASR 5500 only – HNB-GW: HNB-CN Session over IuPS and IuCS
- ASR 5500 only – HNB-GW: SeGW Session IPSec Tunnel
- ASR 5500 only – HSGW services for IPv4
- IPCF (Intelligent Policy Control Function)
- ASR 5500 only – IPSG-only systems (IP Services Gateway)
- LNS session types (L2TP Network Server)
- MME (Mobility Management Entity)
- ASR 5500 only – NEMO (Network Mobility)
- P-GW services for IPv4
- ASR 5500 only – PDIF (Packet Data Interworking Function)
- PDSN services supporting simple IP, Mobile IP, and Proxy Mobile IP
- S-GW (Serving Gateway)
- SGSN (Serving GPRS Support Node) services
- ASR 5000 and VPC-DI – IPv6 and IPv4IPv6 (dual) PDP session recovery is supported for 3G and 2G services
- SaMOG (S2a Mobility over GTP) Gateway (CGW and MRME)
- ASR 5500 only – SAE-GW (System Architecture Evolution Gateway)
- ASR 5500 only – SGSN services (3G and 2.5G services) for IPv4 and PPP PDP contexts

Session recovery is **not supported** for the following functions:

- Destination-based accounting recovery

- GGSN network initiated connections
- GGSN session using more than 1 service instance
- MIP/L2TP with IPSec integration
- MIP session with multiple concurrent bindings
- Mobile IP sessions with L2TP
- Multiple MIP sessions
- :RAB recovery

Important Always refer to the Administration Guides for individual products for other possible session recovery and Interchassis Session Recovery (ICSR) support limitations.

When session recovery occurs, the system reconstructs the following subscriber information:

- Data and control state information required to maintain correct call behavior.
- A minimal set of subscriber data statistics; required to ensure that accounting information is maintained.
- A best-effort attempt to recover various timer values such as call duration, absolute time, and others.
- The idle time timer is reset to zero and the re-registration timer is reset to its maximum value for HA sessions to provide a more conservative approach to session recovery.

Important Any partially connected calls (for example, a session where HA authentication was pending but has not yet been acknowledged by the AAA server) are not recovered when a failure occurs.

Note Failure of critical tasks will result in restarting StarOS. Kernel failures, hypervisor failures or hardware failures will result in the VM restarting or going offline. The use of ICSR between two VPC-DIs or two VPC-SIs is the recommended solution for these types of failure.

Configuring the System to Support Session Recovery

The following procedures allow you to configure the session recovery feature for either an operational system that is currently in-service (able to accept incoming calls) or a system that is out-of-service (not part of your production network and, therefore, not processing any live subscriber/customer data).

Important The session recovery feature, even when the feature use key is present, is disabled by default on the system.

Enabling Session Recovery

As noted earlier, session recovery can be enabled on a system that is out-of-service (OOS) and does not yet have any contexts configured, or on an in-service system that is currently capable of processing calls. However, if the system is in-service, it must be restarted before the session recovery feature takes effect.

Enabling Session Recovery on an Out-of-Service System

The following procedure is for a system that does not have any contexts configured.

To enable the session recovery feature on an out-of-service system, follow the procedure below. This procedure assumes that you begin at the Exec mode prompt.

Step 1 At the Exec mode prompt, verify that the session recovery feature is enabled via the session and feature use licenses on the system by running the **show license info** command.

If the current status of the Session Recovery feature is Disabled, you cannot enable this feature until a license key is installed in the system.

Step 2 Use the following configuration example to enable session recovery.

```
configure
 require session recovery
     end
```

Note After you configure this command, you must save the configuration and then reload the chassis for the command to take effect. For information on saving the configuration file and reloading the chassis, refer to the System Administration Guide for your deployment.

Step 3 Save your configuration as described in *Verifying and Saving Your Configuration*.

The system, when started, enables session recovery, creates all mirrored "standby-mode" tasks, and performs packet processing card reservations and other operations automatically.

Step 4 After the system has been configured and placed in-service, you should verify the preparedness of the system to support this feature as described in Viewing Session Recovery Status, on page 389

Enabling Session Recovery on an In-Service System

When enabling session recovery on a system that already has a saved configuration, the session recovery commands are automatically placed before any service configuration commands in the configuration file.

To enable the session recovery feature on an in-service system, follow the procedure below. This procedure assumes that you begin at the Exec mode prompt.

Step 1 At the Exec mode prompt, verify that the session recovery feature is enabled via the session and feature use licenses on the system by running the **show license info** command:

If the current status of the Session Recovery feature is Disabled, You cannot enable this feature until a license key is installed in the system.

Step 2 Use the following configuration example to enable session recovery.

```
configure
  require session recovery
  end
```

This feature does not take effect until after the system has been restarted.

Step 3 Save your configuration as described in *Verifying and Saving Your Configuration*.

Step 4 Perform a system restart by entering the **reload** command:

The following prompt appears:

```
Are you sure&quest; [Yes|No]:
```

Confirm your desire to perform a system restart by entering **yes**.

The system, when restarted, enables session recovery and creates all mirrored "standby-mode" tasks, performs packet processing card reservations, and other operations automatically.

Step 5 After the system has been restarted, you should verify the preparedness of the system to support this feature as described in Viewing Session Recovery Status, on page 389

More advanced users may opt to simply insert the **require session recovery** command syntax into an existing configuration file using a text editor or other means, and then applying the configuration file manually. Exercise caution when doing this to ensure that this command is placed among the first few lines of any existing configuration file; it must appear before the creation of any non-local context.

Disabling the Session Recovery Feature

To disable the session recovery feature on a system, enter the **no require session recovery** command from the Global Configuration mode prompt.

Important If this command is issued on an in-service system, then the system must be restarted by issuing the **reload** command.

Viewing Session Recovery Status

To determine if the system is capable of performing session recovery, when enabled, enter the **show session recovery status verbose** command from the Exec mode prompt.

The output of this command should be similar to the examples shown below.

```
[local]host_name# show session recovery status
Session Recovery Status:
     Overall Status             :  SESSMGR Not Ready For Recovery
     Last Status Update         :  1 second ago

[local]host_name# show session recovery status
Session Recovery Status:
    Overall Status              :  Ready For Recovery
    Last Status Update          :  8 seconds ago

[local]host_name# show session recovery status verbose
```

```
Session Recovery Status:
    Overall Status              :  Ready For Recovery
    Last Status Update          :  2 seconds ago

               ----sessmgr----       ----aaamgr----         demux
 cpu state     active    standby     active    standby      active    status
---- -------   ------    -------     ------    -------      ------    ------------
 1/1 Active    2         1           1         1            0         Good
 1/2 Active    1         1           0         0            0         Good
 1/3 Active    1         1           3         1            0         Good
 2/1 Active    1         1           1         1            0         Good
 2/2 Active    1         1           0         0            0         Good
 2/3 Active    2         1           3         1            0         Good
 3/0 Active    0         0           0         0            1         Good (Demux)
 3/2 Active    0         0           0         0            1         Good (Demux)
 4/1 Standby   0         2           0         1            0         Good
 4/2 Standby   0         1           0         0            0         Good
 4/3 Standby   0         2           0         3            0         Good
[local]host_name#
```

Viewing Recovered Session Information

To view session state information and any session recovery status, enter the following command:

```
[local]host_name# show subscriber debug-info { callid id | msid id | username
 name }
```

The following example shows the output of this command both before and after a session recovery operation has been performed. The "Redundancy Status" fields in this example have been bold-faced for clarity.

```
username: user1             callid: 01ca11b1                 msid: 0000100003
    Card/Cpu: 4/2
    Sessmgr Instance: 7
    Primary callline:
   Redundancy Status: Original Session
      Checkpoints        Attempts        Success        Last-Attempt       Last-Success
          Full:                69             68             29800ms            29800ms
          Micro:              206            206             20100ms            20100ms
      Current state: SMGR_STATE_CONNECTED
      FSM Event trace:
            State                              Event
            SMGR_STATE_OPEN                    SMGR_EVT_NEWCALL
            SMGR_STATE_NEWCALL_ARRIVED         SMGR_EVT_ANSWER_CALL
            SMGR_STATE_NEWCALL_ANSWERED        SMGR_EVT_LINE_CONNECTED
            SMGR_STATE_LINE_CONNECTED          SMGR_EVT_LINK_CONTROL_UP
            SMGR_STATE_LINE_CONNECTED          SMGR_EVT_AUTH_REQ
            SMGR_STATE_LINE_CONNECTED          SMGR_EVT_IPADDR_ALLOC_SUCCESS
            SMGR_STATE_LINE_CONNECTED          SMGR_EVT_AUTH_SUCCESS
            SMGR_STATE_LINE_CONNECTED          SMGR_EVT_UPDATE_SESS_CONFIG
            SMGR_STATE_LINE_CONNECTED          SMGR_EVT_LOWER_LAYER_UP
    Data Reorder statistics
    Total timer expiry:                0        Total flush (tmr expiry):    0
            Total no buffers:          0        Total flush (no buffers):    0
            Total flush (queue full):  0        Total flush (out of range):  0
            Total flush (svc change):  0        Total out-of-seq pkt drop:   0
            Total out-of-seq arrived:  0
    IPv4 Reassembly Statistics:
              Success:                 0        In Progress: 0
              Failure (timeout):       0        Failure (no buffers): 0
              Failure (other reasons): 0
    Redirected Session Entries:              Allowed:
2000            Current:                                   0
```

```
            Added:                                        0                Deleted:
                             0
            Revoked for use by different subscriber: 0
Peer callline:
 Redundancy Status: Recovered Session
Checkpoints        Attempts        Success        Last-Attempt        Last-Success
      Full:               0              0                 0ms                 0ms
      Micro:              0              0                 0ms                 0ms
  Current state: SMGR_STATE_CONNECTED
  FSM Event trace:
        State                                                        Event
        SMGR_STATE_LINE_CONNECTED            SMGR_EVT_LOWER_LAYER_UP
        SMGR_STATE_CONNECTED                 SMGR_EVT_AUTH_REQ
        SMGR_STATE_CONNECTED                 SMGR_EVT_AUTH_SUCCESS
        SMGR_STATE_CONNECTED                 SMGR_EVT_REQ_SUB_SESSION
        SMGR_STATE_CONNECTED                 SMGR_EVT_RSP_SUB_SESSION
        SMGR_STATE_CONNECTED                 SMGR_EVT_ADD_SUB_SESSION
        SMGR_STATE_CONNECTED                 SMGR_EVT_AUTH_REQ
        SMGR_STATE_CONNECTED                 SMGR_EVT_AUTH_SUCCESS
        SMGR_STATE_CONNECTED                 SMGR_EVT_AUTH_REQ
        SMGR_STATE_CONNECTED                 SMGR_EVT_AUTH_SUCCESS
        SMGR_STATE_CONNECTED                 SMGR_EVT_AUTH_REQ
        SMGR_STATE_CONNECTED                 SMGR_EVT_AUTH_SUCCESS
        SMGR_STATE_CONNECTED                 SMGR_EVT_AUTH_REQ
        SMGR_STATE_CONNECTED                 SMGR_EVT_AUTH_SUCCESS
        SMGR_STATE_CONNECTED                 SMGR_EVT_AUTH_REQ
        SMGR_STATE_CONNECTED                 SMGR_EVT_AUTH_SUCCESS
Data Reorder statistics
        Total timer expiry:        0          Total flush (tmr expiry):  0
        Total no buffers:          0          Total flush (no buffers):  0
        Total flush (queue full):  0          Total flush (out of range):0
        Total flush (svc change):  0          Total out-of-seq pkt drop: 0
        Total out-of-seq arrived:  0
IPv4 Reassembly Statistics:
          Success:                   0        In Progress:               0
          Failure (timeout):         0        Failure (no buffers):      0
          Failure (other reasons):   0
Redirected Session Entries:
          Allowed:                2000         Current:             0
          Added:                               Deleted:             0
          Revoked for use by different subscriber:  0
```

Recovery Control Task Statistics

Recovery Control Task (RCT) statistics show the following:

- Recovery action taken – Migration, Shutdown, Switchover
- Type of event – Planned or Unplanned
- From card to card – slot numbers
- Start time – YYYY-MMM-DD+hh:mm:sss.sss
- Duration – seconds
- Card failure device (such as CPU*n*)
- Card failure reason
- Card is in usable state or not failed
- Recovery action status – Success or failure reason
- If recovery action failed, failure time stamp
- If recovery action failed, failure task facility name

- If recovery action failed, failure instance number

show rct stats Command

The Exec mode **show rct stats** command employs the following syntax:

```
[local]host_name# show rct stats [verbose]
```

Without the **verbose** keyword, a summary output is displayed as show in the example below:

```
RCT stats details (Last 1 Actions)

 #  Action        Type      From  To  Start Time                Duration    Status
--- ------------- --------- ---- ---- ------------------------  ----------  -------
 1  Migration(st) Planned     2    1  2016-Jul-12+13:12:21.865   0.003 sec  Success


RCT stats summary
-----------------
Migrations  =              0
  Management Card:     0             Average time: 0.000 sec
  Packet Card    :     1             Average time: 0.006 sec
Switchovers =              1, Average time - 25.855 sec
```

With the verbose keyword the detailed statistics show in Sample Output for show rct stats verbose, on page 392 are provided.

Sample Output for show rct stats verbose

```
[local]host_name# show rct stats verbose

RCT stats Details (Last 5 Actions)

Stats 1:
    Action          : Migration
    Type            : Planned
    From            : 5
    To              : 6
    Start Time      : 2017-Apr-04+03:02:00.132
    Failure Reason  : CPU_CRITICAL_TASK_FAILURE
    Failure Device  : CPU_0
    Is Card Usable  : Yes
    Recovery Status : Success
    Facility        : N.A
    Instance        : N.A
    Duration        : 066.050 sec
    Graceful        : Enabled
      Recovered [1] :[f:sessmgr, i:6, cpu:50, pid:13170
      Recovered [2] :[f:sessmgr, i:3, cpu:50, pid:13167]

RCT stats Details (Last 5 Actions)

Stats 2:
    Action          : Shutdown
    From            : 12
    To              : 13
    Start Time      : 2017-Apr-04+03:02:10.100
    Is Card Usable  : Yes
    Failure Reason  : NPU_LC_CONNECT_TOP_FAIL
    Failure Device  : PAC_LC_CONNECT_HARDWARE
    Recovery Status : Success
```

```
    Facility        : N.A
    Instance        : N.A
    Duration        : 002.901 sec
    Graceful        : Enabled
      Recovered [1] :[f:sessmgr, i:6, cpu:50, pid:13170
      Recovered [2] :[f:sessmgr, i:3, cpu:50, pid:13167]

Stats 3:
    Action          : Migration
    From            : 7
    To              : 11
    Start Time      : 2017-Apr-04+03:03:40.120
    Is Card Usable  : Yes
    Failure Reason  : N.A.
    Failure Device  : N.A
    Recovery Status : Success
    Facility        : N.A
    Instance        : N.A
    Duration        : 003.423 sec
    Graceful        : Enabled
      Recovered [1] :[f:sessmgr, i:6, cpu:50, pid:13170
      Recovered [2] :[f:sessmgr, i:3, cpu:50, pid:13167]

Stats 4:
    Action          : Migration
    From            : 7
    To              : 11
    Start Time      : 2017-Apr-04+03:03:41.256
    Is Card Usable  : Yes
    Failure Reason  : N.A.
    Failure Device  : N.A
    Recovery Status : TASK_MIGRATION_FAIL_PREMIGRATE
    Facility        : vpnmgr
    Instance        : 13
    Duration        : 005.222 sec
    Graceful        : Enabled
      Recovered [1] :[f:sessmgr, i:6, cpu:50, pid:13170
      Recovered [2] :[f:sessmgr, i:3, cpu:50, pid:13167]

Stats 5:
    Action          : Migration
    From            : 6
    To              : 7
    Start Time      : 2017-Apr-04+04:18:30.106
    Is Card Usable  : Yes
    Failure Reason  : N.A.
    Failure Device  : N.A
    Recovery Status : TASK_MIGRATION_FAIL_RENAME
    Facility        : sessmgr
    Instance        : 63
    Duration        : 004.134 sec
    Graceful        : Enabled
      Recovered [1] :[f:sessmgr, i:6, cpu:50, pid:13170
      Recovered [2] :[f:sessmgr, i:3, cpu:50, pid:13167]

RCT stats Summary
-----------------
Migrations  =       3, Average time = 4.260 sec
Switchovers =       0
```

CHAPTER 28

Interchassis Session Recovery

This chapter describes how to configure Interchassis Session Recovery (ICSR). The product Administration Guides provide examples and procedures for configuration of basic services on the system. You should select the configuration example that best meets your service model, and configure the required elements for that model as described in the respective product Administration Guide, before using the procedures described below.

In the context of VPC-DI, a chassis is a VPC-DI instance. ICSR provides failover protection for identically configured VPC-DI instances.

Important ICSR is a licensed Cisco feature that requires a separate license. Contact your Cisco account representative for detailed information on specific licensing requirements. For information on installing and verifying licenses, refer to the *Managing License Keys* section of *Software Management Operations*.

This chapter discusses the following:

Overview

The ICSR feature provides the highest possible availability for continuous call processing without interrupting subscriber services. ICSR allows the operator to configure gateways for redundancy purposes. In the event of a gateway failure, ICSR allows sessions to be transparently routed around the failure, thus maintaining the user experience. ICSR also preserves session information and state.

The system supports ICSR between two instances that support ICSR in the same StarOS release. For combination VMs where more than one service type is in use, only those services that support ICSR can make use of ICSR.

ICSR can provide redundancy for site/row/rack/host outages and major software faults. The two instances must be run on non-overlapping hosts and network interconnects. ICSR is only supported between identically configured VPC-DI or VPC-SI instances.

VPC-DI supports both L2 and L3 ICSR.

ICSR is implemented through the use of redundant virtual chassis. The virtual chassis for each VPC-DI instance are configured as primary and backup, with one being active and one standby. Both virtual chassis are connected to the same AAA server. A checkpoint duration timer controls when subscriber data is sent from the active chassis to the standby chassis. If the active chassis handling the call traffic goes out of service, the standby chassis transitions to the active state and continues processing the call traffic without interrupting the subscriber session.

The virtual chassis determine which is active through a proprietary TCP-based connection known as the Service Redundancy Protocol (SRP) link. The SRP link is used to exchange Hello messages between the active CFs in the primary and backup chassis and must be maintained for proper system operation. For additional information, refer to the *Session Recovery* chapter.

ICSR licenses are currently supported for the following services:

- GGSN Gateway GPRS Support Node
- P-GW Packet Data Network Gateway
- S-GW Serving Gateway
- SAE-GW System Architecture Evolution Gateway

L2TP Access Concentrator (LAC) functionality for ICSR is supported by the following protocol and services:

- eGTP enhanced GPRS Tunneling Protocol
- GGSN Gateway GPRS Support Node
- P-GW Packet Data Network Gateway
- SAEGW System Architecture Evolution Gateway

L2TP Access Concentrator (LAC) functionality for ICSR is not supported by the following service:

- PMIP Proxy Mobile IP

L2TP Network Server (LNS) functionality for ICSR is not supported by any services.

Note ICSR support for LAC requires a separate LAC license, as well as an Inter-Chassis Session Recovery license.

Note Contact your Cisco account representative to verify whether a specific service supports ICSR as an option.

Interchassis Communication

VPC-DI virtual chassis are configured to support ICSR communication using periodic Hello messages. These messages are sent by the active CF in each virtual chassis to notify the peer of its current state. The Hello message contains information about the virtual chassis such as its configuration and priority. A dead interval is used to set a time limit for a Hello message to be received from the chassis' CF peer. If the CF in the standby chassis does not receive a Hello message from the active CF in the active chassis within the dead interval, the standby virtual chassis transitions to the active state.

In situations where the SRP link goes out of service, a priority scheme is used to determine which chassis processes the session. The following priority scheme is used:

- route modifier
- chassis priority
- MIO/UMIO/MIO2 MAC address

Checkpoint Messages

Checkpoint messages are sent from the active CF in the active virtual chassis to the active CF in the standby virtual chassis. These messages are sent at specific intervals and contain all the information needed to recreate the sessions on the standby chassis, if that chassis were to become active. Once a session exceeds the checkpoint duration, checkpoint data is collected on the session.

For additional information, refer to the *ICSR Checkpointing* appendix.

SRP CLI Commands

Exec Mode CLI Commands

Exec mode **srp** CLI configuration commands can be used to enable, disable and initiate SRP functions. The table below lists and briefly describes these commands. For complete information see the *Exec Mode Commands (D-S)* chapter of the *Command Line Interface Reference*.

Table 28: srp CLI Commands

Command	Description
srp disable nack micro-chkpt-cmd	Disables the sending of NACK messages from the standby chassis that may trigger a full checkpoint from the active chassis. Sending full checkpoints increases SRP bandwidth. This command disables the NACK feature for a specific micro-checkpoint which is failing continuously.
srp initiate-audit manual-with-sync	Initiates a forced audit between ICSR chassis. This audit ensures that two ICSR peers are synchronized and identifies any discrepancies prior to scheduled or unscheduled switchover events.
srp initiate-switchover	Executes a forced switchover from active to inactive. When executed on the active chassis, this command switches the active chassis to the inactive state and the inactive chassis to an active state. See Note below.
srp reset-auth-probe-fail	Resets the auth probe monitor failure information to 0.
srp reset-diameter-fail	Resets the Diameter monitor failure information to 0.
srp terminate-post-process	Forcibly terminates post-switchover processing.

Command	Description
srp validate-configuration	Validates the configuration for an active chassis.
srp validate-switchover	Validates that both active and standby chassis are ready for a planned SRP switchover.

Important For release 20.0 and higher, ICSR will verify session manager connectivity on both chassis prior to allowing a manual switchover. If one or more of the session managers in the active chassis is not connected on the standby chassis, the switchover will not be initiated. An error message will appear on the screen noting the number of session managers that are mismatched. The **force** keyword can be used to initiate the switchover despite the mismatch(es). The output of the **show checkpoint statistics verbose** command will not indicate "Ready" for a session manager instance ("smgr inst") in the "peer conn" column for any instance that is not connected to the peer chassis.

show Commands

Exec mode **show srp** commands display a variety of information related to SRP functions. The table below lists and briefly describes these commands. For complete information on these commands, see the *Exec Mode show Commands (Q-S)* chapter of the *Command Line Interface Reference*.

Table 29: show srp Commands

Command	Description
show srp audit-statistics	Displays statistics of an external audit.
show srp call-loss statistics	Displays the history of calls lost during switchover.
show srp checkpoint statistics	Displays check pointing statistics on session redundancy data (session managers, current call recovery records, etc.).
show srp info	Displays Service Redundancy Protocol information (context, chassis state, peer, connection state, etc.).
show srp monitor	Displays SRP monitor information.
show srp statistics	Displays SRP statistics (hello messages sent, configuration validation, resource messages, switchovers, etc.).

For additional information about the output of **show srp** commands, see the *Statistics and Counters Reference*.

AAA Monitor

AAA servers are monitored using the authentication probe mechanism. AAA servers are considered Up if the authentication-probe receives a valid response. AAA servers are considered Down when the **max-retries count** specified in the configuration of the AAA server has been reached. SRP initiates a switchover when none of the configured AAA servers responds to an authentication probe. AAA probing is only performed on the active chassis.

Important A switchover event caused by an AAA monitoring failure is non-revertible.

If the newly active chassis fails to monitor the configured AAA servers, it remains as the active chassis until one of the following occurs:

- a manual switchover
- another non-AAA failure event causes the system to switchover
- a CLI command is used to clear the AAA failure flag and allow the chassis to switch to standby

BGP Interaction

The Service Redundancy Protocol implements revertible switchover behavior via a mechanism that adjusts the route modifier value for the advertised loopback/IP Pool routes. The initial value of the route modifier value is determined by the chassis' configured role and is initialized to a value that is higher than a normal operational value. This ensures that in the event of an SRP link failure and an SRP task failure, the correct chassis is still preferred in the routing domain.

Important For ICSR you must configure **busyout ip pool** commands in the same order on Active and Standby chassis to avoid SRP validation failures.

The Active and Standby chassis share current route modifier values. When BGP advertises the loopback and IP pool routes, it converts the route modifier into an autonomous systems (AS) path prepend count. The Active chassis always has a lower route modifier, and thus prepends less to the AS-path attribute. This causes the route to be preferred in the routing domain.

If communication on the SRP link is lost, and both chassis in the redundant pair are claiming to be Active, the previously Active chassis is still preferred since it is advertising a smaller AS-path into the BGP routing domain. The route modifier is incremented as switchover events occur. A threshold determines when the route modifier should be reset to its initial value to avoid rollover.

Requirements

ICSR configurations require the following:

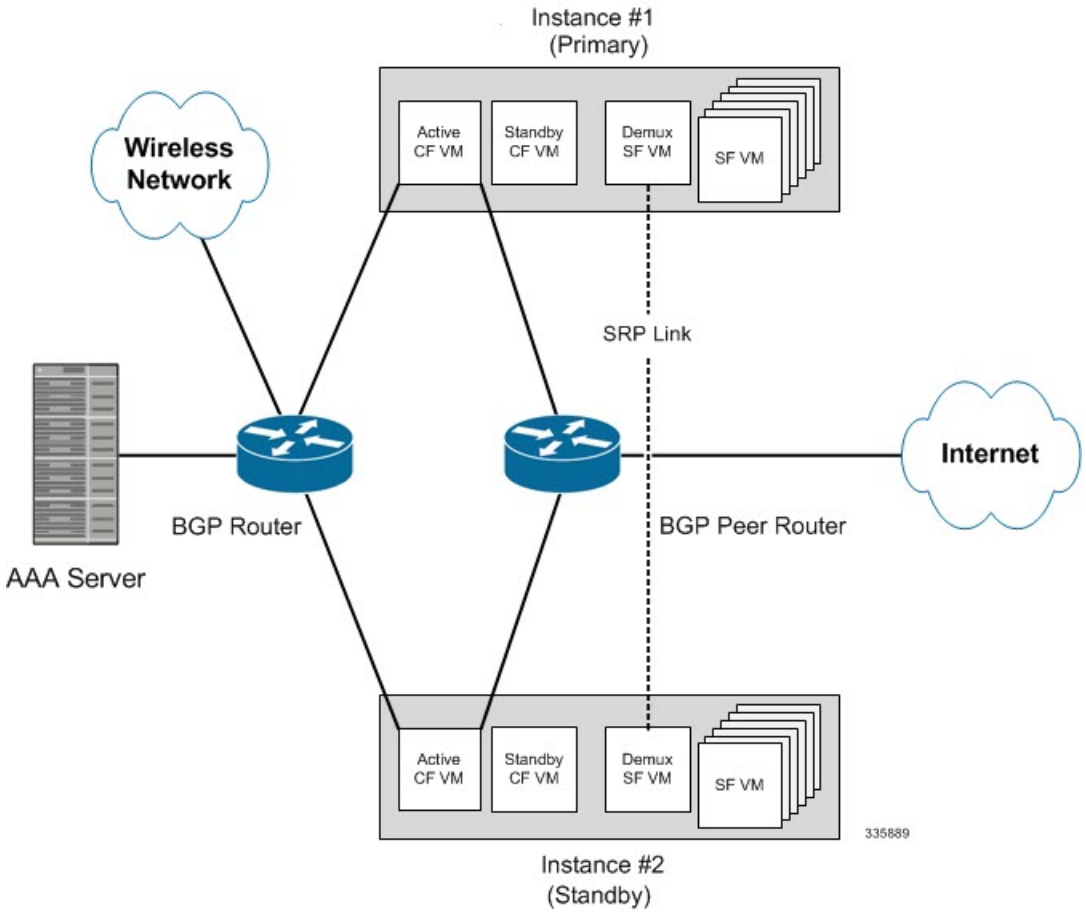
- Two VPC-DI instances or UGP instances identically configured for the same service types. The services must be bound on an SRP-activated loopback interface. Both instances must have identical hardware.
- Three contexts:
  - **Redundancy** – to configure the primary and backup chassis redundancy.
  - **Source** – AAA configuration of the specified nas-ip-address must be the IP address of an interface bound to an HA, or any core network service configured within the same context.
  - **Destination** – to configure monitoring and routing to the PDN.
- Border Gateway Protocol (BGP) – ICSR uses the route modifier to determine the chassis priority.

Important ICSR is a licensed Cisco feature. Verify that each chassis has the appropriate license before using these procedures. To do this, log in to both chassis and execute a **show license information** command. Look for "Inter-Chassis Session Recovery". If the chassis is not licensed, please contact your Cisco account representative.

RADIUS and Diameter protocols can be monitored to trigger a switchover.

The following figure shows an ICSR network.

Figure 13: Virtualized StarOS ICSR Network

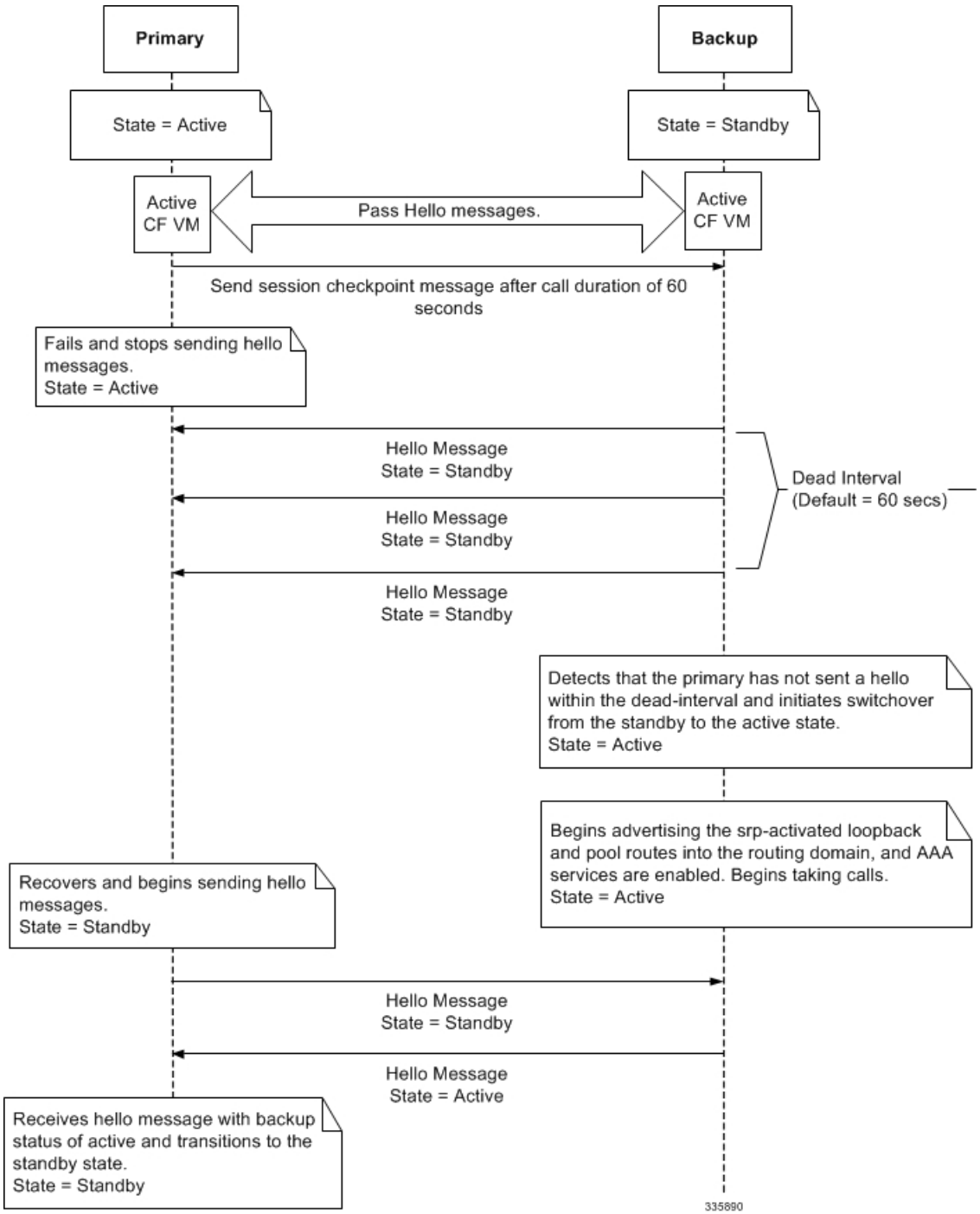


ICSR Operation

This section shows operational flows for ICSR.

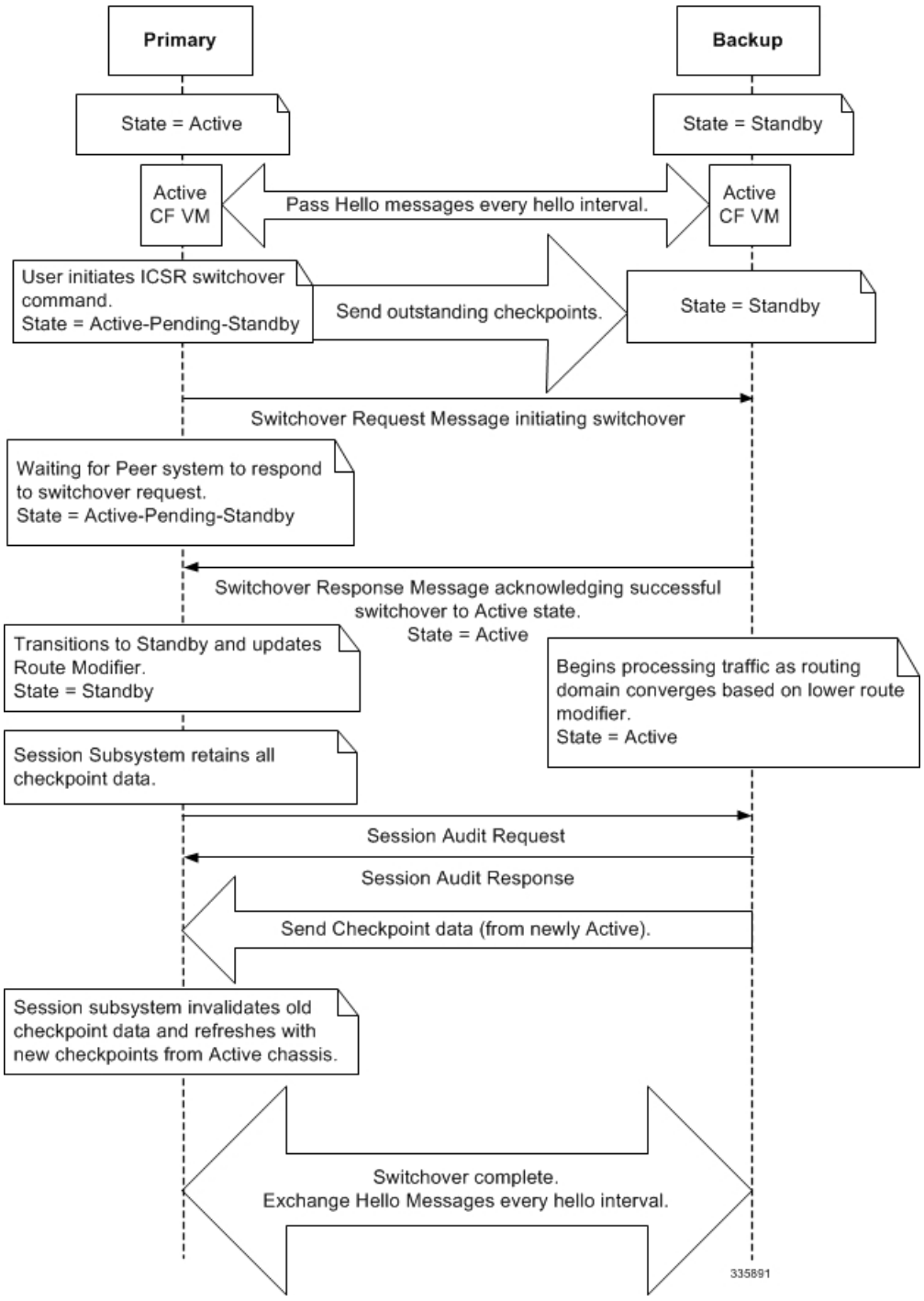
The following figure shows an ICSR process flow due to a primary failure.

Figure 14: ICSR Process Flow (Primary Failure)



The following figure shows an ICSR process flow due to a manual switchover.

Figure 15: ICSR Process Flow (Manual Switchover)



Chassis Initialization

When StarOS is simultaneously initialized on each VPC-DI virtual chassis, the active CFs send Hello messages to each other. The peer sends a response, establishes communication between the chassis, and messages are sent that contain configuration information.

During initialization, if both virtual chassis are misconfigured in the same mode - both active (primary) or both standby (backup), the chassis with the highest priority (lowest number set with the ICSR **priority** command) becomes active and the other chassis becomes the standby.

If the chassis priorities are the same, StarOS compares the two MAC addresses of the active CFs and the chassis with the higher CF MAC address becomes active. For example, if the CFs have MAC addresses of *00-02-43-03-1C-2B* and *00-02-43-03-01-3B*, the last 3 sets of octets (the first 3 sets are the vendor code) are compared. In this example, the *03-1C-2B* and *03-01-3B* are compared from left to right. The first pair of octets in both MAC addresses are the same, so the next pairs are compared. Since the *01* is lower than the *1C*, the VPC-DI virtual chassis with the CF MAC address of *00-02-43-03-1C-2B* becomes active and the other chassis the standby.

Chassis Operation

This section describes how the chassis communicate, maintain subscriber sessions, and perform chassis switchover.

Chassis Communication

If one virtual chassis is in the active state and one in the standby state, they both send Hello messages at each hello interval via their active CFs. Subscriber sessions that exceed the checkpoint session duration are included in checkpoint messages that are sent to the standby chassis. The checkpoint message contains subscriber session information so if the active chassis goes out of service, the backup chassis becomes active and is able to continue processing the subscriber sessions. Additional checkpoint messages occur at various intervals whenever subscriber session information is updated on the standby chassis.

The SRP Configuration mode **checkpoint session** command includes a number of keywords that enable you to:

- Set the type of compression algorithm to be used for SRP payload messages.
- Set the amount of time the chassis waits before check pointing an existing call session. Checkpoints can be separately set for IMS and/or non-IMS sessions.
- Configure the interval between the sending of macro-checkpoints (full checkpoints) between the active and standby chassis.

For additional information see the *Service Redundancy Protocol Configuration Mode Commands* chapter in the *Command Line Interface Reference*.

Chassis Switchover

If the active virtual chassis goes out of service, the standby chassis continues to send Hello messages. If the standby chassis does not receive a response to the Hello messages from the active CF within the dead interval, the standby chassis initiates a switchover. During the switchover, the active CF in the standby chassis begins advertising its srp-activated loopback and pool routes into the routing domain. Once the chassis becomes active, it continues to process existing AAA services and subscriber sessions that had checkpoint information, and is also able to establish new subscriber sessions.

When the primary virtual chassis is back in service, it sends Hello messages to the active CF in the configured peer. The peer sends a response, establishes communication between the active CFs in the chassis, and sends Hello messages that contain configuration information. The primary chassis receives an Hello message that shows the backup chassis state as active and then transitions to standby. The Hello messages continue to be sent to each peer, and checkpoint information is now sent from the active chassis to the standby chassis at regular intervals.

When chassis switchover occurs, the session timers are recovered. The access gateway session recovery is recreated with the full lifetime to avoid potential loss of the session and the possibility that a renewal update was lost in the transitional checkpoint update process.

Configuring ICSR

Important The ICSR configuration must be the same on the primary and backup chassis. If each chassis has a different Service Redundancy Protocol (SRP) configuration, the session recovery feature does not function and sessions cannot be recovered when the active chassis goes out of service.

This section describes how to configure basic ICSR on each chassis. For information on commands that configure additional parameters and options, refer to the *Command Line Interface Reference*.

Important *For releases prior to StarOS 17.0,* ICSR should not be configured for chassis supporting L2TP calls.

The procedures described below assume the following:

- The chassis have been installed and configured with core network services.

  For more configuration information and instructions on configuring services, refer to the respective product Administration Guide.

- In addition, the IP address pools must be **srp activated**.
- AAA server is installed, configured and accessible by both chassis.

  For more information on configuring the AAA server, refer to the *AAA Interface Administration and Reference*.

- BGP router installed and configured. See *Routing* for more information on configuring BGP services.

To configure ICSR on a primary and/or backup chassis:

Step 1 Configure the SRP context by applying the example configuration in Configuring the Service Redundancy Protocol (SRP) Context, on page 405.

Step 2 Modify the source context of the core network service by applying the example configuration in Modifying the Source Context for ICSR, on page 414.

Step 3 Modify the destination context of core network service by applying the example configuration in Modifying the Destination Context for ICSR, on page 415.

Step 4 *Optional:* Disable bulk statistics collection on the standby system by applying the example configuration in Disabling Bulk Statistics Collection on a Standby System, on page 417.

Step 5 Verify your primary and backup chassis configuration as described in Verifying the Primary and Backup Configuration, on page 417.

Step 6 Save your configuration as described in *Verifying and Saving Your Configuration*.

Configuring the Service Redundancy Protocol (SRP) Context

To configure the system to work with ICSR:

Step 1 Create the chassis redundancy context and bind it to the IP address of the primary chassis by applying the example configuration in Creating and Binding the SRP Context, on page 405. For VPC-DI instances, this should be the IP address of the active CF in the primary VPC-DI intance.

Step 2 Configure the chassis redundancy context with priority, chassis mode, hello interval, dead-interval and peer IP address by applying the example configuration in Configuring SRP Context Parameters, on page 406.

Step 3 Configure the SRP context with interface parameters (including interface name, IP address and port number) for interchassis communication by applying the example configuration in Configuring the SRP Context Interface Parameters, on page 411.

Step 4 Verify your SRP context configuration as described in Verifying SRP Configuration, on page 414.

Step 5 Save your configuration as described in *Verifying and Saving Your Configuration*.

Creating and Binding the SRP Context

Use the example below to create the SRP context and bind it to the IP address of the active CF in the primary VPC-DI instance:

Important ICSR is configured on two VPC-DI instances. Be sure to create the redundancy context on both systems. CLI commands must be executed on both systems. Log onto both active CFs before continuing. Always make configuration changes on the active CF in the primary VPC-DI instance first. Before starting this configuration, identify which VPC-DI to configure as the primary and use that login session.

```
configure
  context srp_ctxt_name [-noconfirm]
    service-redundancy-protocol
      bind address ip_address
      end
```

Notes:

- ICSR should be configured and maintained in a separate context.
- Be sure to bind the local IP address to the active CF in the primary VPC-DI instance. When configuring the backup VPC-DI instance, be sure to bind the local IP address to the backup chassis.

Configuring SRP Context Parameters

Important CLI commands must be executed on both VPC instances. Log onto both active CFs before continuing. Always make configuration changes on the primary VPC instance first.

Basic Parameters

This configuration assigns a chassis mode and priority, and also configures the redundancy link between the primary and backup chassis:

```
configure
  context srp_ctxt_name
    service-redundancy-protocol
      chassis-mode { primary | backup }
      priority priority
      peer-ip-address ip_address
      hello-interval dur_sec
      dead-interval dead_dur_sec
      end
```

Notes:

- ICSR should be configured and maintained in a separate context.
- When assigning the chassis mode on the backup chassis be sure to enter the **backup** keyword.
- The **checkpoint** command sets the amount of time the chassis waits before check pointing an existing call session. Checkpoints can be set for IMS (VoLTE) and/or non-IMS sessions. The checkpoint is a snapshot of the current application state that can be used to restart its execution in case of failure. The default setting is 60 seconds.
- The **priority** determines which chassis becomes active in the event that both chassis are misconfigured with the same chassis mode; see Chassis Initialization, on page 403. The higher priority chassis has the lower number. Be sure to assign different priorities to each chassis.
- Enter the IP chassis of the backup chassis as the **peer-ip-address** to the primary chassis. Assign the IP address of the primary chassis as the **peer-ip-address** to the backup chassis.
- The **dead-interval** must be at least three times greater than the **hello-interval**. For example, if the hello interval is 10, the dead interval should be at least 30. System performance is severely impacted if the hello interval and dead interval are not set properly. An optional **delay-interval** command allows you to delay the start dead-interval for an interval following the loading of configuration files.

SRP Redundancy, AAA and Diameter Guard Timers

Guard timers ensure that local failures, such as reboots and task restarts, do not result in ICSR events which can be disruptive.

The **guard timer** command configures the redundancy-guard-period and monitor-damping-period for SRP service monitoring.

```
configure
  context context_name
    service-redundancy-protocol variable
```

```
      guard-timer { aaa-switchover-timers { damping-period seconds |
guard-period seconds } | diameter-switchover-timers { damping-period seconds
 | guard-period seconds } | srp-redundancy-timers { aaa  { damping-period
seconds | guard-period seconds } | bgp { damping-period seconds | guard-period
 seconds } | diam  { damping-period seconds | guard-period seconds } }
      end
```

Notes:

- **aaa-switchover-timers** – sets timers that prevent back-to-back ICSR switchovers due to an AAA failure (post ICSR switchover) while the network is still converging.
  - **damping-period** – configures a delay time to trigger an ICSR switchover due to a monitoring failure within the guard-period.
  - **guard-period** – configures the local-failure-recovery network-convergence timer.
- **diameter-switchover-timers** – sets timers that prevent a back-to-back ICSR switchover due to a Diameter failure (post ICSR switchover) while the network is still converging.
  - **damping-period** – configures a delay time to trigger an ICSR switchover due to a monitoring failure within the guard-period.
  - **guard-period** – configures the local-failure-recovery network-convergence timer.
- **srp-redundancy-timers** – sets timers that prevent an ICSR switchover while the system is recovering from a local card-reboot/critical-task-restart failure.
  - **aaa** – local failure followed by AAA monitoring failure
  - **bgp** – local failure followed by BGP monitoring failure
  - **diam** – local failure followed by Diameter monitoring failure

DSCP Marking of SRP Messages

You can enable separate DSCP marking of SRP control and checkpoint messages. The **dscp-marking** command sets DSCP marking values for SRP control and checkpoint (session maintenance) messages.

```
configure
  context context_name
    service-redundancy-protocol
      dscp-marking { control | session } dscp_value
```

Notes:

- *dscp_value* can be:
  - **af11** – Assured Forwarding Class 1 low drop PHB (Per Hop Behavior)
  - **af12** – Assured Forwarding Class 1 medium drop PHB
  - **af13** – Assured Forwarding Class 1 high drop PHB
  - **af21** – Assured Forwarding Class 2 low drop PHB
  - **af22** – Assured Forwarding Class 2 medium drop PHB
  - **af23** – Assured Forwarding Class 2 high drop PHB
  - **af31** – Assured Forwarding Class 3 low drop PHB
  - **af32** – Assured Forwarding Class 3 medium drop PHB

- **af33** – Assured Forwarding Class 3 high drop PHB
- **af41** – Assured Forwarding Class 4 low drop PHB
- **af42** – Assured Forwarding Class 4 medium drop PHB
- **af43** – Assured Forwarding Class 4 high drop PHB
- **be** – Best effort Per-Hop-Behaviour (default)
- **cs1** – Class selector 1 PHB
- **cs2** – Class selector 2 PHB
- **cs3** – Class selector 3 PHB
- **cs4** – Class selector 4 PHB
- **cs5** – Class selector 5 PHB
- **cs6** – Class selector 6 PHB
- **cs7** – Class selector 7 PHB
- **ef** – Expedited Forwarding PHB, for low latency traffic

Optimizing Switchover Transitions

There are several SRP configuration options that reduce the transition time from the active to standby gateways (primarily P-GW) in support of VoLTE traffic.

Important These features require an updated ICSR license to support the enhancements. Contact your Cisco account representative for additional information.

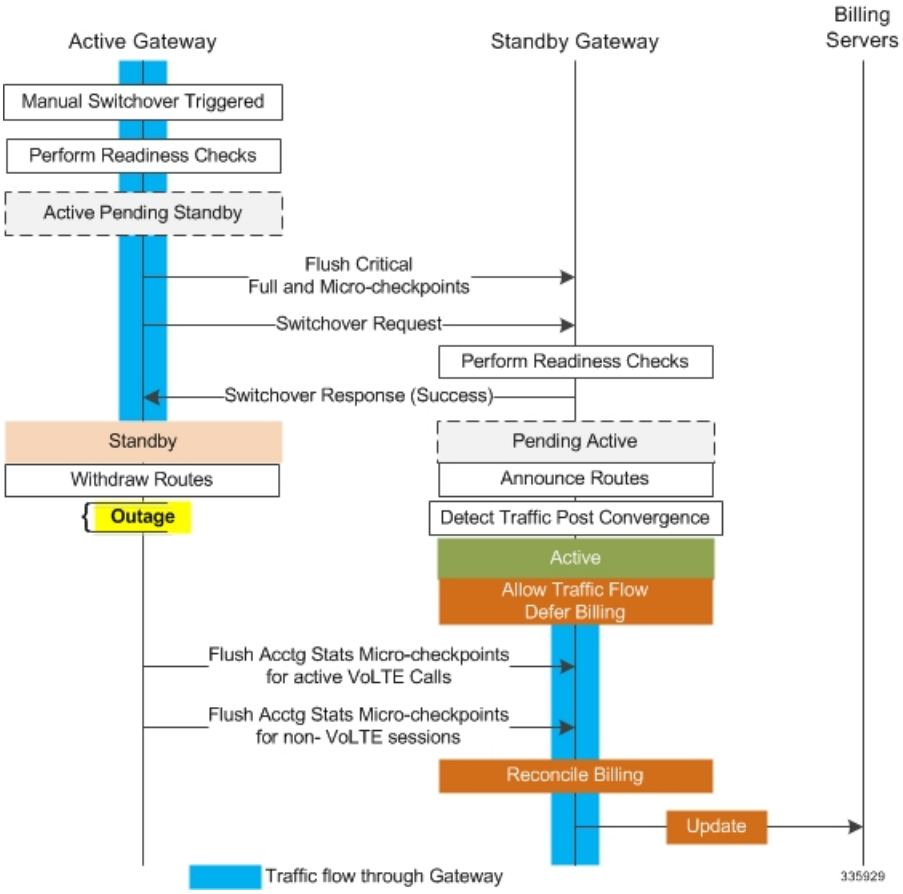
Allow Non-VoLTE Traffic During ICSR Switchover

The ICSR framework reduces switchover disruption for VoLTE traffic by enabling VoLTE traffic on the newly active gateway prior to reconciling the billing information and enabling communication with the newly active gateway when accounting is not deemed critical.

This functionality extends to all other traffic, including data sessions and default bearer traffic for IMS/e911, The following ICSR functionality is provided for all non-VoLTE data traffic:

- When a switchover occurs, the newly active gateway forwards all traffic the moment the gateway becomes active.
- External communication with billing servers is deferred. See the traffic flow diagram below.
- When the newly active gateway receives all billing-related checkpointing information from the previously active gateway, it reconciles the billing data before communicating with external billing servers OCS (Online Charging System) or OFCS (Offline Charging System).

Figure 16: Call Flow: Reduce Non-VoLTE Data Outage



The **switchover allow-all-data-traffic** SRP Configuration mode CLI command allows all data traffic (VoLTE and non-VoLTE) during switchover transition. This command overwrites the **switchover allow-volte-data-traffic** command if enabled on a P-GW.

```
configure
  context context_name
    service-redundancy-protocol
      switchover allow-all-data-traffic
```

Important The **switchover allow-all-data-traffic** command must be run on both chassis to enable this feature.

The **switchover allow-volte-data-traffic** SRP Configuration mode CLI command allows VoLTE data traffic during ICSR switchover transition.

```
configure
  context context_name
    service-redundancy-protocol
      switchover allow-volte-data-traffic [ maintain-accounting ]
```

Notes:

- When **maintain-accounting** is enabled, accounting accuracy is maintained for VoLTE calls.VoLTE data is allowed on the active gateway after VoLTE accounting statistics are flushed.

Allow All Data Traffic

The SRP Configuration mode **switchover allow-all-data-traffic** command allows all data traffic (VoLTE and non-VoLTE) during switchover transition. This command overwrites the **switchover allow-volte-data-traffic** command if enabled on a P-GW. This feature reduces data traffic outage during the switchover.

Important This CLI command must be run on both the active and standby chassis to enable this feature.

All data traffic is allowed on the active chassis during flushing and internal auditing. The billing information is reconciled in the background once the flush is complete.

Allow Early Active Transition

The SRP Configuration mode **switchover allow-early-active-transition** command enables early transition to active state during an ICSR switchover. By default this feature is disabled.

Use this command in concert with the **switchover allow-all-data-traffic** or **allow-volte-data-traffic** (without **maintain accounting** option) command to further reduce data outage during a planned switchover. The outage window is the amount time between initiating an ICSR switchover and when the newly active chassis starts processing data.

Important You must enable one of the commands identified above on both ICSR chassis prior to enabling this command.

Graceful Cleanup of ICSR After Audit of Failed Calls

During an Audit on the gateways (P-GW/S-GW/GGSN/SAE-GW) after Session Recovery or an ICSR event, if any critical information, internally or externally related to a subscriber session seems inconsistent, ICSR will locally purge the associated session information.

Since external gateways (peer nodes) are unaware of the purging of this session, the UE session may be maintained at other nodes. This leads to hogging of resources external to the gateway and an unreachable UE for VoLTE calls.

When this feature is enabled, graceful cleanup for an ICSR audit of failed calls occurs. External signaling notifies peers of session termination before purging the session. The gateway will attempt to notify external peers of the removal of the session. External nodes to the local gateway include S-GW, P-GW, SGSN, MME, AAA, PCRF and IMSA.

Audit failure can occur because of missing or incomplete session information. Therefore, only the peers for which the information is available will be notified.

The **require graceful-cleanup-during-audit-failure** Global Configuration mode CLI command enables or disables the graceful cleanup feature.

```
configure
  require graceful-cleanup-during-audit-failure [ del-cause non-ims-apn
  { system-failure | none } ]
```

Optimization of Switchover Control Outage Time

The ICSR framework minimizes control outage time associated with the flushing of critical full checkpoint statistics, network convergence and internal auditing.

The amount of time consumed by the following activities affects control outage time during switchover:

- **Critical Flush** – During the Active to Pending-Standby transition, all sessmgrs flush any pending critical FCs (Full Checkpoints). During this time, the active chassis drops all control packets. If control signaling is allowed during this stage, a call may get disconnected based on the control message type and accounting information will be lost.
- **Network Convergence** – This encompasses the amount of time taken to update routes and send control/data to the newly active chassis. Control messages are dropped during the transition.
- **Accounting Flush** – During this flush stage data counts are synchronized between chassis. If control signaling is allowed during this flush, the call may get disconnected based on the control message type, and accounting information will be lost for calls that existed before switchover.
- **Audit** – During audit new calls are not allowed because synchronization of call resources may result in clearing of the calls.

The **switchover control-outage-optimization** CLI command allows new calls during the Accounting Flush, as soon as the Audit is completed. This SRP Configuration mode command enables the quicker restoration of control traffic (call-setup, modification, deletion) following an ICSR switchover.

```
configure
  context context_name
    service-redundancy-protocol
      switchover control-outage-optimization
      end
```

Configuring the SRP Context Interface Parameters

This procedure configures the communication interface with the IP address and port number within the SRP context. This interface supports interchassis communication.

Important CLI commands must be executed on both chassis. Log onto both chassis before continuing. Always make configuration changes on the primary chassis first.

```
configure
  context vpn_ctxt_name [-noconfirm]
      interface srp_if_name
      ip-address { ip_address | ip_address/mask }
      exit
    exit
  port ethernet slot_num/port_num
    description des_string
    medium { auto | speed { 10 | 100 | 1000 } duplex { full | half } }
    no shutdown
    bind interface srp_if_name srp_ctxt_name
    end
```

Configuring NACK Generation for SRP Checkpoint Messaging Failures

Enabling NACK Messaging from the Standby Chassis

Transport (TCP) level re-transmission is supported on the SRP link between ICSR chassis. SRP configuration also supports optional application level checks to ensure checkpoints are received at the Standby chassis. Failed attempts to receive and apply checkpoints send NACK messages to the Active chassis.

When this feature is enabled and the standby chassis sends a NACK in response to the receipt of a micro-checkpoint (MC) that fails to be successfully applied, the standby chassis sends another NACK. The standby chassis will send more NACKs (configurable, default = 3) within a 10-minute window if a macro-checkpoint (FC) is not received. NACKs will continue to be sent and the 10-minute reset until an FC is received and applied, or the configured number of max-responses is reached.

You can also specify the number of times a NACK is sent within the 10-minute window in response to a failed MC or FC (Default = 3).

A **nack** keyword in the SRP Configuration mode **checkpoint session** command allows you to enable generation of NACK messages in response to checkpoint message failures on a Standby ICSR chassis.

Important The **nack** keyword will only appear if a special ICSR optimization feature license has been purchased and installed. Contact your Cisco account representative for assistance.

```
configure
   context context_name
      service-redundancy-protocol variable
         checkpoint session nack { macro | micro } [ max-response number ]

         no checkpoint session nack { macro | micro }
         end
```

Notes:

- **max-response** is the number of times a NACK is sent within the 10-minute window in response to a failed MC or FC expressed as an integer from 0 through 65535 (Default = 3).

A **periodic-interval** keyword in the SRP Configuration mode **checkpoint session** command allows you to configure the interval between the sending of macro-checkpoints (FCs) between the active and standby chassis.

Important The **periodic-interval** keyword will only appear if a special ICSR optimization feature license has been purchased and installed. Contact your Cisco account representative for assistance.

```
configure
   context context_name
      service-redundancy-protocol variable
         checkpoint session periodic-interval minutes
         default checkpoint session periodic-interval
         no checkpoint session periodic-interval
         end
```

Selective Disabling of NACK Messaging

The NACK mechanism sends a NACK message for any ICSR checkpoint failure on the standby chassis. Every NACK sent from the standby chassis triggers a full checkpoint from the active chassis.

If the micro-checkpoint is failing continuously and sending NACKs, the active chassis keeps sending full-checkpoints. This increases SRP bandwidth.

CLI commands allow an operator to selectively disable and re-enable NACK messages for specific micro-checkpoints.

The Exec mode **srp disable nack micro-chkpt-cmd** disables the sending of a NACK from the standby chassis.

```
srp disable nack micro-chkpt-cmd chkpt_number
```

chkpt_number specifies the checkpoint number to be disabled as an integer from 1 through 255. You can obtain checkpoint numbers (CMD IDs) from the output of the **show srp checkpoint info** command.

You can re-enable the micro-checkpoint using the **srp enable nack micro-chkpt-cmd** command.

```
srp enable nack micro-chkpt-cmd chkpt_number
```

Configuring LZ4 Compression Algorithm

You can optionally enable LZ4 compression algorithm for SRP messaging payload. The zlib algorithm remains as the default.

LZ4 is a very fast lossless compression algorithm with near-linear scalability for multi-threaded applications.

The **compression** keyword in the SRP Configuration mode **checkpoint session** command allows you to enable the use of the LZ4 compression algorithm.

Important The **compression** keyword will only appear if a special ICSR optimization feature license has been purchased and installed. Contact your Cisco account representative for assistance.

The following command sequence enables the use of LZ4 compression:

```
configure
   context context_name
      service-redundancy-protocol
         checkpoint session compression lz4
         end
```

LZ4 compression is effective only if both chassis are configured with LZ4. If any one chassis has zlib (default) configured, the compression algorithm reverts to zlib. The algorithm is negotiated only during initial socket establishment. Once agreed no more negotiation takes place until the TCP socket connection is reset.

Reducing Sync-Up Time with Standby ICSR Chassis

The default method for synchronizing the SRP database requires tens of seconds of delay whenever the TCP connection between the Active and Standby session managers is established. Once the TCP connection is established, heart beat messages are exchanged between both ICSR chassis every 3 seconds. The standby chassis waits for seven heart beat messages from the active chassis before it is ready to accept data. This may cause significant delay in session manager database synchronization on the standby chassis.

You can enable an aggressive method for synchronizing the session manager database reduces recovery time in the following scenarios:

- Standby Session Manager crash
- Packet processing card failure on Standby chassis
- Standby chassis reboot
- Temporary loss and recovery of SRP connection

The aggressive method reduces the number of heartbeat messages and amount of housekeeping information exchanged between ICSR chassis.

The SRP Configuration mode **standby database-recovery aggressive** command allows you to select normal or aggressive restoration of the SRP database.

The following command sequence enables the aggressive recovery mode:

```
configure
   context context_name
      service-redundancy-protocol
         standby database-recovery aggressive
         end
```

The default form of this command restores the normal mode of SRP database recovery.

Verifying SRP Configuration

Verify that your SRP contexts were created and configured properly by running the **show srp info** command (Exec Mode) on each chassis.

Notes:

- The interval is specified as an integer divisible by 15 in the range from 30 through 1440 (Default = 45 minutes). The interval range for sending full checkpoints is 30 minutes to 24 hours (1140 minutes).

Modifying the Source Context for ICSR

To modify the source context of core service:

Step 1 Add the Border Gateway Protocol (BGP) router AS-path and configure the gateway IP address, neighbor IP address, remote IP address in the source context where the core network service is configured, by applying the example configuration in Configuring BGP Router and Gateway Address, on page 414.

Step 2 Configure the service redundancy context with the BGP neighbor context and IP address to monitor the BGP link activity by applying the example configuration in Configuring the SRP Context for BGP, on page 415.

Step 3 Verify your BGP context configuration by following the steps in Verifying BGP Configuration, on page 415.

Step 4 Save your configuration as described in *Verifying and Saving Your Configuration*.

Configuring BGP Router and Gateway Address

Use the following example to create the BGP context and network addresses.

```
configure
  context source_ctxt_name
    router bgp AS_num
      network gw_ip_address
      neighbor neighbor_ip_address remote-as AS_num
      end
```

Notes:

- *source_ctxt_name* is the context where the core network service is configured.

Configuring the SRP Context for BGP

Use the following example to configure the BGP context and IP addresses in the SRP context.

```
configure
  context srp_ctxt_name
    service-redundancy-protocol
      monitor bgp context source_ctxt_name neighbor_ip_address
      end
```

neighbor_ip_address can be entered in IPv4 dotted-decimal or IPv6 colon-separated-hexadecimal notation. Multiple IP addresses can be added per context as IPv4 and/or IPv6 IP addresses.

An ICSR failover is triggered when all BGP peers within a context are down.

Optionally, you can configure SRP peer groups within a context. ICSR failover would then occur if all peers within a group fail. This option is useful in deployments in which a combination of IPv4 and IPv6 peers are spread across multiple paired VLANs, and IPv4 or IPv6 connectivity is lost by all members of a peer group.

A sample configuration for SRP peer groups within a context ("PGWin") appears below.

```
monitor bgp context PGWin 10.1.1.16 group 1
monitor bgp context PGWin 10.1.1.17 group 1
monitor bgp context PGWin 69.2.215.0 group 2
monitor bgp context PGWin 69.2.215.1 group 2
monitor bgp context PGWin 2001:4333:201:1102:103:2a1:: group 3
monitor bgp context PGWin 2001:4333:201:1102:103:2a1:0:1 group 3
```

In the above sample configuration, ICSR failover would occur if both addresses in group 1, 2 or 3 lost connectivity.

For additional information refer to the description of the **monitor bgp**, **monitor diameter** and **monitor authentication-probe** commands in the *Service Redundancy Protocol Configuration Mode Commands* chapter of the *Command Line Interface Reference*.

Verifying BGP Configuration

Verify your BGP configuration by entering the **show srp monitor bgp** command (Exec Mode).

Modifying the Destination Context for ICSR

To modify the destination context of core service:

Step 1 Add the BGP router and configure the gateway IP address, neighbor IP address, remote IP address in the destination context where the core network service is configured, by applying the example configuration in Configuring BGP Router and Gateway Address in Destination Context, on page 416.

Step 2 Configure the service redundancy context with BGP neighbor context and IP address to monitor the BGP link activity by applying the example configuration in Configuring SRP Context for BGP for Destination Context, on page 416.

Step 3 Set the subscriber mode to **default** by following the steps in Setting Subscriber to Default Mode, on page 416.

Step 4 Verify your BGP context configuration by following the steps in Verifying BGP Configuration in Destination Context, on page 416.

Step 5 Save your configuration as described in *Verifying and Saving Your Configuration*.

Configuring BGP Router and Gateway Address in Destination Context

Use the following example to create the BGP context and network addresses.

```
configure
  context dest_ctxt_name
    router bgp AS_num
      network gw_ip_address
      neighbor neighbor_ip_address remote-as AS_num
      end
```

Notes:

- *AS_num* is the autonomous systems path number for this BGP router.

Configuring SRP Context for BGP for Destination Context

Use the following example to configure the BGP context and IP addresses in the SRP context.

```
configure
  context srp_ctxt_name
    service-redundancy-protocol
      monitor bgp context dest_ctxt_name neighbor_ip_address
      end
```

Setting Subscriber to Default Mode

Use the following example to set the subscriber mode to **default**.

```
configure
  context dest_ctxt_name
    subscriber default
      end
```

Verifying BGP Configuration in Destination Context

Verify your BGP configuration by entering the **show srp monitor bgp** command (Exec Mode).

Disabling Bulk Statistics Collection on a Standby System

You can disable the collection of bulk statistics from a system when it is in the standby mode of operation.

Important When this feature is enabled and a system transitions to standby state, any pending accumulated statistical data is transferred at the first opportunity. After that no additional statistics gathering takes place until the system comes out of standby state.

Use the following example to disable the bulk statistics collection on a standby system.

```
configure
  bulkstat mode
    no gather-on-standby
    end
```

Repeat this procedure for both systems.

Verifying the Primary and Backup Configuration

This section describes how to compare the ICSR configuration on the primary and backup systems.

Step 1 Enter the **show configuration srp** command on each system (Exec mode).

Step 2 Verify that both chassis have the same SRP configuration information.

The output looks similar to following:

```
config
  context source
    interface haservice loopback
      ip address 172.17.1.1 255.255.255.255 srp-activate
    #exit
    radius attribute nas-ip-address address 172.17.1.1
    radius server 192.168.83.2 encrypted key 01abd002c82b4a2c port 1812
    radius accounting server 192.168.83.2 encrypted key 01abd002c82b4a2c port 1813
    ha-service ha-pdsn
      mn-ha-spi spi-number 256  encrypted secret 6c93f7960b726b6f6c93f7960b726b6f hash-algorithm md5
      fa-ha-spi remote-address 192.168.82.0/24 spi-number 256 encrypted secret 1088bdd6817f64df
      bind address 172.17.1.1
    #exit
  #exit
  context destination
    ip pool dynamic 172.18.0.0 255.255.0.0 public 0 srp-activate
    ip pool static 172.19.0.0 255.255.240.0 static srp-activate
  #exit
  context srp
    service-redundancy-protocol
    #exit
  #exit
```

Configuring Subscriber State Management Audit Process

This audit is to ensures that two ICSR peers are in synch and identifies any discrepancies prior to any scheduled or unscheduled switchover events.

Step 1 Enter the SRP Context mode and enter the **service-redundancy-protocol** command.

Step 2 Enter the **audit daily-start-time** command. Specify the daily start time as an hour and minute. For example, a start time of 06 00 indicates that the audit will begin at 6:00 AM.

Step 3 Enter the **audit periodicity** command. Specify the interval in minutes for generating SRP audit statistics as an integer from 60 through 1440. For example, a periodicity of 90 indicates that SRP audit statistics will be generated every 90 minutes beginning at the specified start time. Default = 60.

A sample configuration sequence appears below.

```
config
  context srp
    service-redundancy-protocol
      audit daily-start-time 06 00
      audit periodicity 90
      end
```

Troubleshooting ICSR Operation

SSD

StarOS supports an ICSR-specific **show support details** (SSD) command that outputs the results from a series of Exec mode **show** commands.This mini SSD reduces capture time when debugging ICSR timing issues between the Active and Standby chassis, facilitating quicker resolution of the problem.

The **show support details icsr** command produces a mini SSD that contains the output of the following **show** commands:

- show srp info
- show srp checkpoint statistics
- show srp checkpoint statistics verbose
- show srp checkpoint statistics debug-info
- show srp checkpoint statistics sessmgr all
- show srp checkpoint statistics sessmgr all debug-info
- show srp checkpoint statistics ipsecmgr all
- show srp checkpoint statistics sessmgr all write-list-stats
- show srp checkpoint info
- show srp monitor
- show srp monitor all
- show srp monitor diameter debug
- show srp statistics
- show srp call-loss statistics

- show srp audit-statistics
- show session subsystem facility sessmgr all debug-info

The SSD output can be directed to a file that can be stored to **/flash** or off the chassis. For additional information, see the *Command Line Interface Reference*.

show srp details

The Exec mode **show srp details** command displays comprehensive information used by TAC personnel to troubleshoot ICSR/SRP issues.

Updating the Operating System

Updating the operating system (StarOS™) on an ICSR system is performed separately on each system while it is in standby mode. Traffic disruption is minimal since an active system will be handling call sessions while the standby system is being updated.

The general upgrade sequence is as follows:

1. Download the StarOS software image and copy/transfer it to both the active and standby system.
2. Save the currently running configurations on both systems.
3. Update the standby backup system first.
4. Initiate an SRP switchover from the active primary system to make the standby backup system active.
5. Update the standby primary system.
6. Initiate an SRP switchover from the active backup system to make the standby primary system active.

The four-part flowchart below shows a more complete view of all the procedures required to complete the StarOS upgrade process.

Caution Enabling the Demux on MIO/UMIO/MIO2 feature changes resource allocations within the system. This directly impacts an upgrade or downgrade between StarOS versions in ICSR configurations. In Star OS 21.24 release, the ICSR upgrade above two releases (N-2) is not fully qualified. For more information, contact Cisco Account Representative for procedural assistance prior to upgrading or downgrading your ICSR deployment.

Figure 17: ICSR Software Upgrade – Part 1

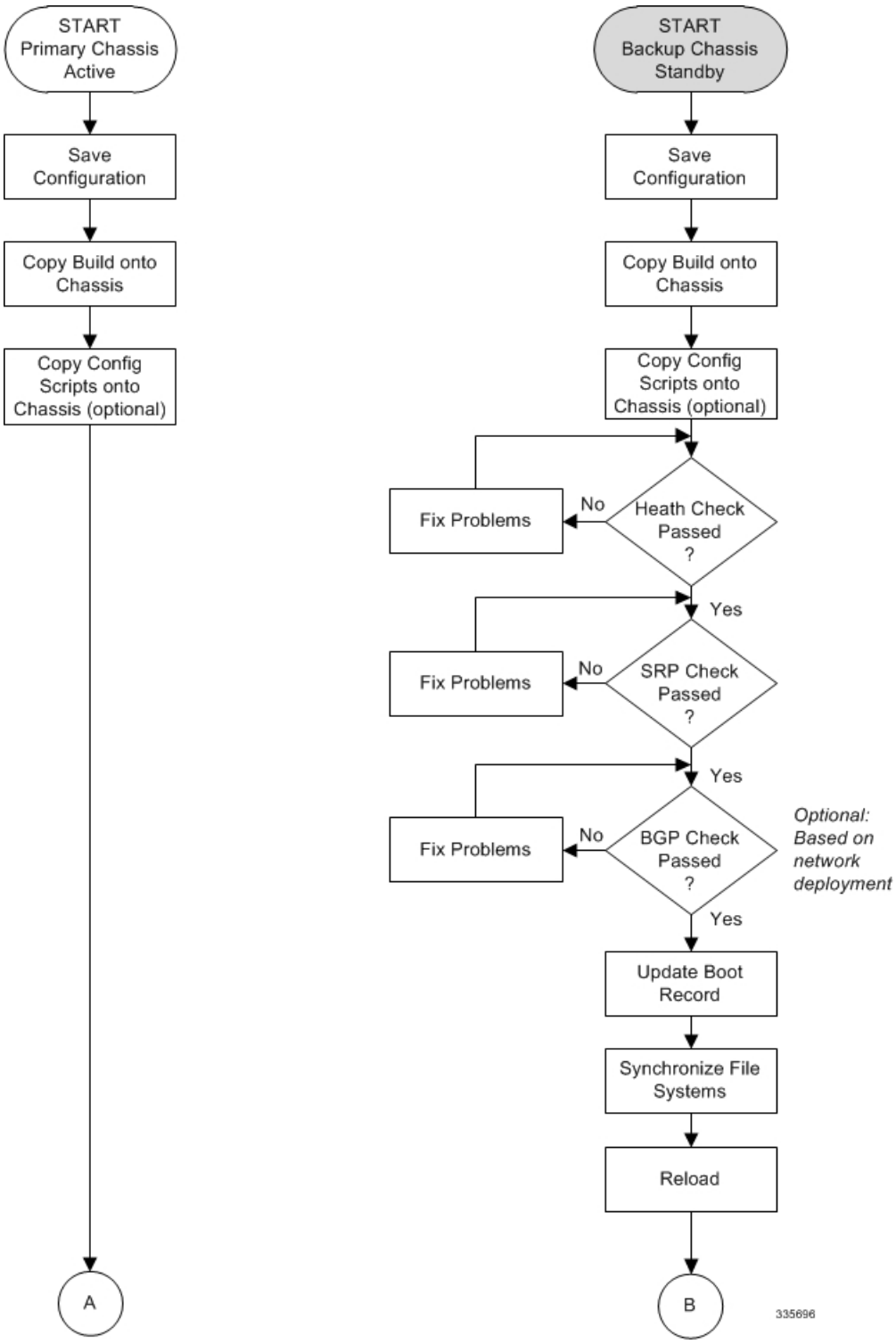


Figure 18: ICSR Software Upgrade – Part 2

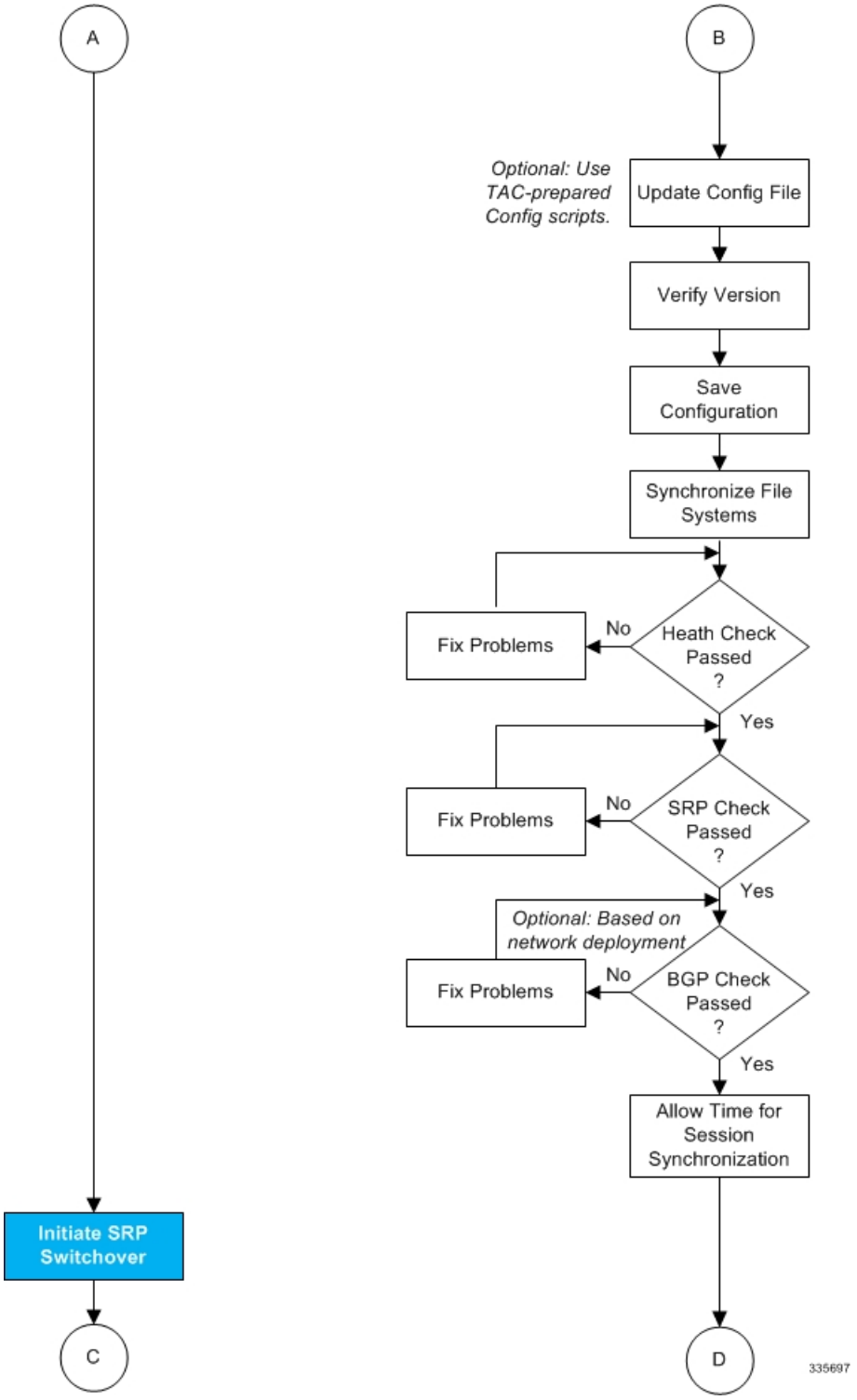


Figure 19: ICSR Software Upgrade – Part 3

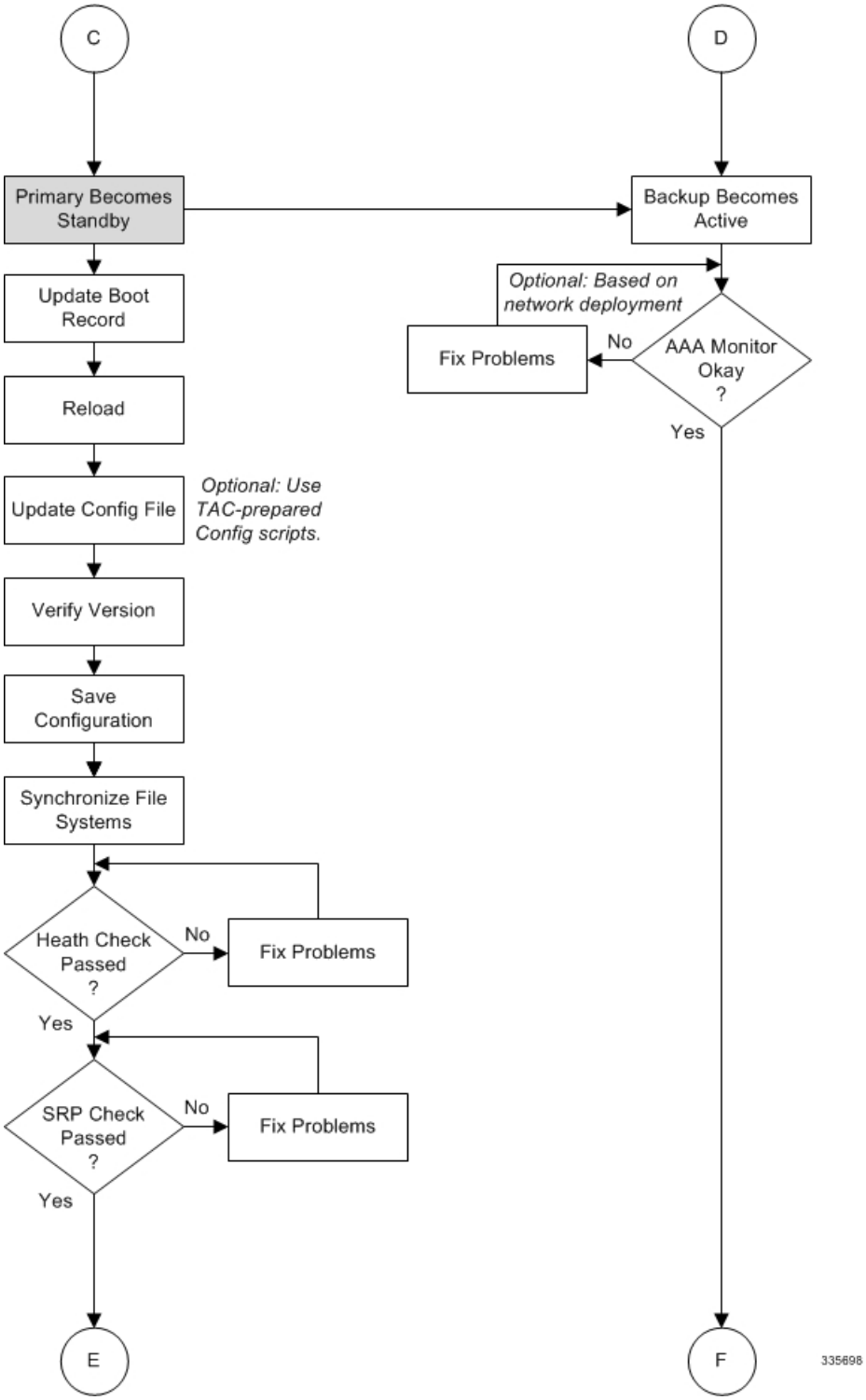
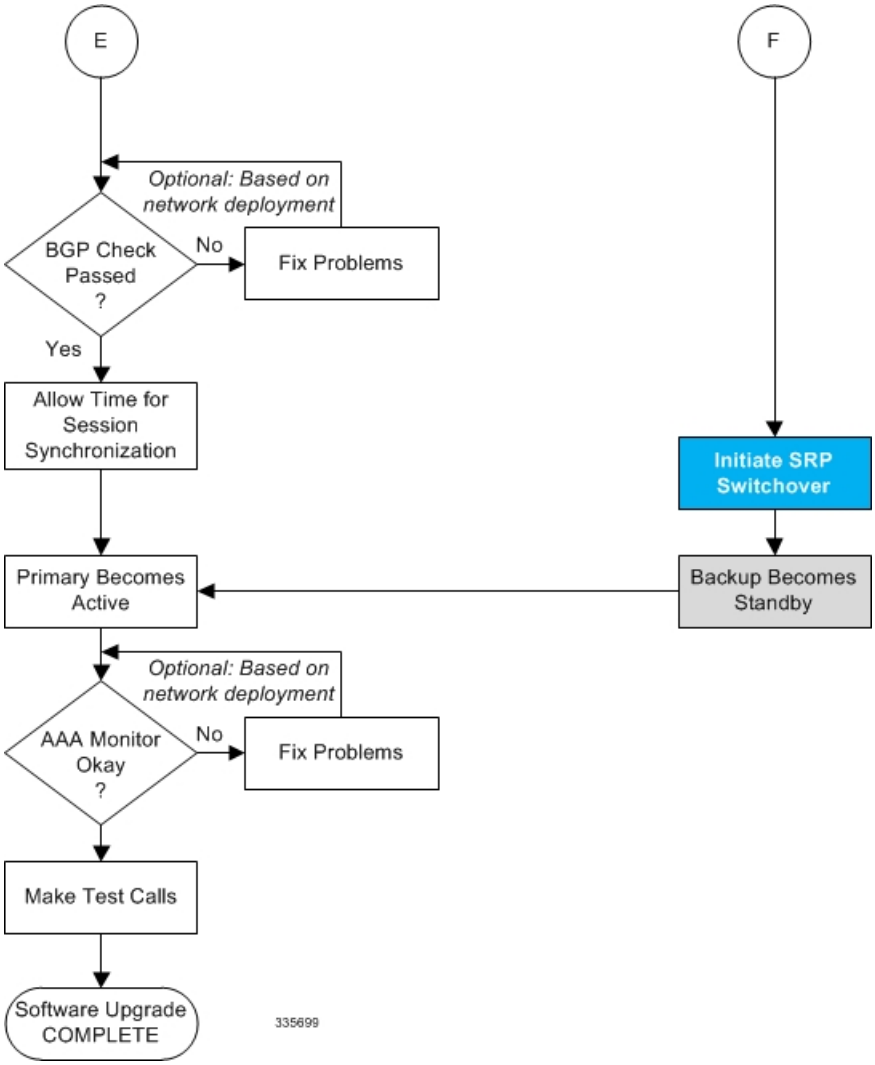


Figure 20: ICSR Software Upgrade – Part 4



Both ICSR Systems

Perform the tasks described below on both the primary (active) and backup (standby) ICSR systems.

Downloading and Transferring the StarOS Image

Step 1 Verify that there is enough free space on the **/flash** device to accommodate the new operating system image file by entering the following Exec mode command:

```
[local]host_name directory /flash
```

Step 2 Access to the Cisco support site and download facility is username and password controlled. Download the software image to a network location or local drive from which it can be uploaded to the **/flash** device.

Step 3 Transfer the new operating system image file to the **/flash** device using one of the following methods:

a) Copy the file from a network location or local drive using the copy command

```
[local]host_name copy from_url to_url [-noconfirm]
```

b) Transfer the file to the **/flash** device using an FTP client with access to the system. The FTP client must be configured to transfer the file using binary mode.

c) Transfer the file to the **/flash** device using an SFTP client with access to the system.

Step 4 Verify that the image file was successfully transferred to the **/flash** device by running the following Exec mode command:

```
[local]host_name directory /flash
```

Step 5 Run the **show version /flash/***image_filename* command to verify the build information. For example:

```
local]host_name show version /flash/image_filename.bin
```

Note Any CRC errors will be displayed in the output of the above command. If any errors appear, check the build and re-transfer it onto the chassis. Confirm that the correct image version and build description is displayed

Standby ICSR System

Perform the tasks described below on the backup or standby ICSR system.

Performing Health Checks

Health checks are a series of Exec mode **show** commands to determine the readiness of the system to handle a software update.

Step 1 Run **show card table all |grep unknown**. No output should be displayed.

Step 2 Run **show card table |grep offline**. No output should be displayed.

Step 3 Run **show resources |grep Status**. The output should display "Within acceptable limits".

Step 4 Run **show alarm outstanding**. Review the output for any issues that may preclude performing the software update.

Performing SRP Checks

Service Redundancy Protocol (SRP) checks verify that the mechanism for monitoring ICSR system status is operational.

Step 1 Run **show srp monitor all**.

Step 2 Review the output for any issues that may preclude performing the software update.

Performing BGP Checks

Border Gateway Protocol (BGP) checks are only required when BGP is used to support redundant interchassis communication. These checks are run per context and per service type.

Step 1 For each BGP-enabled context, run **show ip bgp summary**. Verify that the BGP peers are connected and that IPv4 and IPv6 peers are up. Repeat for all BGP-enable contexts.

Step 2 Run **show** *service_name* **all |grep "Service Status:"**. The service should be "Started". Repeat for all services running on the chassis.

Updating the Boot Record

You must add a new boot stack entry for the recently downloaded software image (.bin) file.

Step 1 Run the Exec mode **show boot** command to verify that there are less than 10 entries in the boot.sys file and that a higher priority entry is available (minimally there is no priority 1 entry in the boot stack).

Step 2 Create a new boot stack entry for the new file group, consisting of the new operating system image file and the currently used CLI configuration file by entering the following Global Configuration command:

```
[local]host_name(config)# boot system priority number image image_url /flash/filename config
 cfg_url /flash/filename
```

Step 3 Assign the next highest priority to this entry, by using the <N-1> method, wherein you assign a priority number that is one number less than your current highest priority.

If priority 1 is in use, you must renumber the existing entries to ensure that at least that priority is available.

The maximum number of boot stack entries that can be contained in the boot.sys file is 10. If there are already 10 entries in the boot stack, you must delete at least one of these entries (typically, the lowest priority) and, if necessary, renumber some or all of the other entries before proceeding. Use the **no boot system priority** command to delete a book stack entry.

For information on using the **boot system priority** command, refer to the *Adding a New Boot Stack Entry* section in this guide

Synchronizing File Systems

Synchronize the local file systems by entering the following Exec mode command:

```
[local]host_name# filesystem synchronize all
```

Reboot StarOS

Reboot the StarOS by entering the following command:

```
[local]host_name# reload [-noconfirm]
```

As the system reboots, it loads the new operating system software image and its corresponding CLI configuration file using the new boot stack entry configured earlier.

After the system reboots, establish a CLI session and enter the **show version** command to verify that the active software version is correct.

Optional for PDSN: If you are using the IP Pool Sharing Protocol during your upgrade, refer to *Configuring IPSP Before the Software Upgrade* in the *PDSN Administration Guide*.

Updating the Configuration File

Features in the new operating system may require changes to the configuration file. These changes can be done manually or facilitated by custom scripts prepared by Cisco TAC. Make whatever changes are necessary prior to saving the updated configuration file.

Verifying the Software Version

After the system has successfully booted, verify that the new StarOS version is running by executing the Exec mode **show version** command.

You can run the Exec mode **show build** command to display additional information about the StarOS build release.

Saving the Configuration File

Use the Exec mode save configuration command to save the currently running configuration to the **/flash** device and to an off-chassis location (external memory device or network URL). The off-chassis copy assures that you will have a fallback, loadable configuration file should a problem be encountered.

Completing the Update Process

Repeat the following tasks to complete the upgrade process on the standby secondary chassis:

- Synchronizing File Systems, on page 425
- Performing Health Checks, on page 424
- Performing SRP Checks, on page 424
- Performing BGP Checks, on page 424

Waiting for Session Synchronization

Allow time for session synchronization to occur between the ICSR chassis before preceding to the next steps.

Step 1 Run the **show session recovery status verbose** command on both chassis. Proceed to the next steps only when no errors are seen in the output of this command.

Step 2 On the standby chassis, run **show srp checkpoint statistics |more**.

Step 3 On active chassis, run **show subs summary |grep Total**.

Step 4 Compare the number of subscribers on the active chassis and the number of Current pre-allocated calls: on the standby chassis. They should be similar (within 5%). Allow a few minutes for systems to complete synchronization.

Primary System

Perform the tasks described below on the primary (active) ICSR system.

Initiating an SRP Switchover

An SRP switchover places the primary chassis in standby mode and makes the backup chassis active. The secondary chassis is now processing sessions with the upgraded software.

Step 1 On the primary chassis run the **srp initiate-switchover** command. All existing sessions will be migrated to the backup chassis and it begins servicing new session requests. Allow the switchover process to complete.

Step 2 On the primary chassis, run the **show srp info** command. Chassis State should indicate Standby when switchover is complete.

Step 3 On the backup chassis, confirm the switchover is complete by running the **show srp info** command. Chassis State should indicate Active when switchover is complete.

Checking AAA Monitor Status on the Newly Active System

If your network deployment requires communication with AAA servers, log into the newly active system and perform an AAA monitor check. You will be checking for the existence of any SNMP traps that indicate the system cannot communicate with AAA servers (**starSRPAAAUnreachable**).

Step 1 Run the Exec mode command **show snmp trap history |grep starSRPAAAUnreachable**.

Step 2 There should be no output for this command, or no very recent SNMP trap notifications (based on the event timestamp).

Step 3 If the active system cannot communicate with one or more AAA servers, refer to AAA Monitor for additional information.

Completing the Software Update

Log into the backup (standby) system and repeat the following tasks to complete the upgrade process on the backup (standby) system:

- Updating the Boot Record, on page 425
- Reboot StarOS, on page 425
- Updating the Configuration File, on page 426
- Verifying the Software Version, on page 426
- Saving the Configuration File, on page 426
- Synchronizing File Systems, on page 425
- Performing Health Checks, on page 424
- Performing SRP Checks, on page 424
- Performing BGP Checks, on page 424
- Waiting for Session Synchronization, on page 426

Initiating an SRP Switchover

This SRP switchover places the primary system in active mode and returns the backup system to the standby. The primary chassis is now processing sessions with the upgraded software.

Step 1 On the backup chassis run the **srp initiate-switchover** command. All existing sessions will be migrated to the primary chassis which begins servicing new session requests. Allow the switchover process to complete.

Step 2 On the backup system, run the **show srp info** command. Chassis State should indicate Standby when switchover is complete.

Step 3 On the primary system, confirm the switchover is complete by running the **show srp info** command. Chassis State should indicate Active when switchover is complete.

Making Test Calls

Once the chassis state is verified and subscribers are migrated, perform new call testing to make sure calls are successful.

Fallback Procedure

To revert to the previous configuration and software build, perform the following steps as a user with administrative privileges.

Step 1 Run the Exec mode **show boot** command. The topmost lowest numbered entry of the displayed output should be the new configuration with the new software build. The second topmost entry should be the backup configuration.

Step 2 Remove the topmost boot entry n, and synchronize the configuration across the management cards.

```
[local]host_name# config
[local]host_name(config)# no boot system priority n
[local]host_name(config)# end
[local]host_name# filesystem synchronize all
```

Step 3 Reboot the system to load its previous configuration.

```
[local]host_name# reload
```

Step 4 Perform health checks as described in Performing Health Checks, on page 424

CHAPTER 29

Support Data Collector

The Support Data Collector (SDC) is a system feature that allows scheduled collection of process state, counter, event and attribute data that may be useful when troubleshooting problems at an installation site.

This chapter includes the following sections:

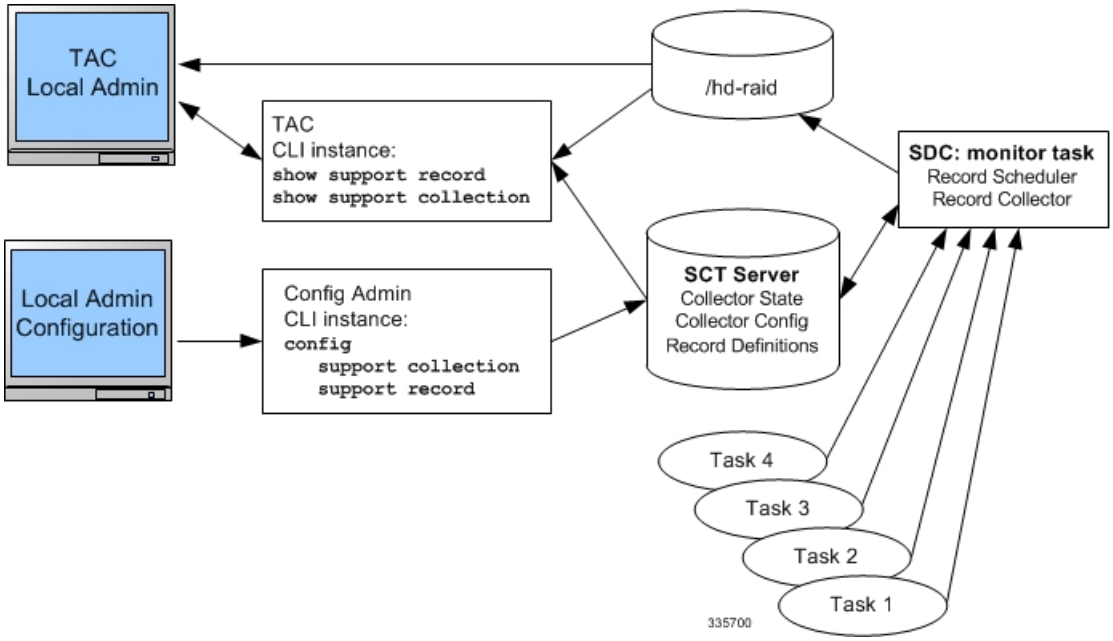
Overview

The task of collecting the support data is performed by a background CLI task called the record collector. The administrator configures the SDC via the CLI with the commands to be executed on a periodic basis. The record collector always runs in the background and checks if there are records to be collected.

When it is time to collect support data, the scheduler executes the configured sequence of CLI commands and stores the results in a gunzipped (.gz) file on the hard-disk. This file is called an SDR (Support Data Record), and represents a snapshot of the overall state of the system at that time.

Technical Assistance Center (TAC) personnel and local administrators can review the SDRs on-line or by transferring them off the system. They may also wish to investigate the collector state information. The figure below shows system tasks that contain state and counter information. Arrows between tasks and processes represent messenger requests and indicate the predominant flow of data.

Figure 21: SDC Tasks and Processes<



Configuring SDR Collection

The Support Data Record (SDR) is an ordered set of the CLI support commands' display output that is stored in a stand-alone compressed file. Each CLI support command output is stored in its own record section. The record section is identified by a record section name and its ASCII command syntax. For example, the record section *show_version* would have a CLI command string of "show version".

The order in which the record section commands appear in the configuration is significant. All of the support record section commands must be configured together as an ordered set. In other words, just specifying one command by itself will result in just that one command output constituting the contents of the entire SDR.

The user may configure a specific set of record sections for the SDR which may or may not include some or all of the default SDR record sections. This configuration is stored in the Global Configuration section of the configuration file. Refer to Configuration Commands (Global Configuration Mode), on page 434 for more detail on the **support record section** command.

Displaying the SDR Collection Configuration

The **show configuration verbose** command displays the default support record sections, if the user has not specified any support record sections. If the user has configured support record sections, then the **show configuration** command displays user-configured support record sections. The support collection schedule configuration also appears in the **show configuration** output under the Global Configuration section.

Collecting and Storing the SDR Information

At the scheduled time, the Support Data Collector (SDC), if active, runs in the background to collect all the record section commands that have been specified. This information is concatenated as one contiguous output. The output is compressed and stored as a file on disk in the **/hd-raid/support/record/** directory.

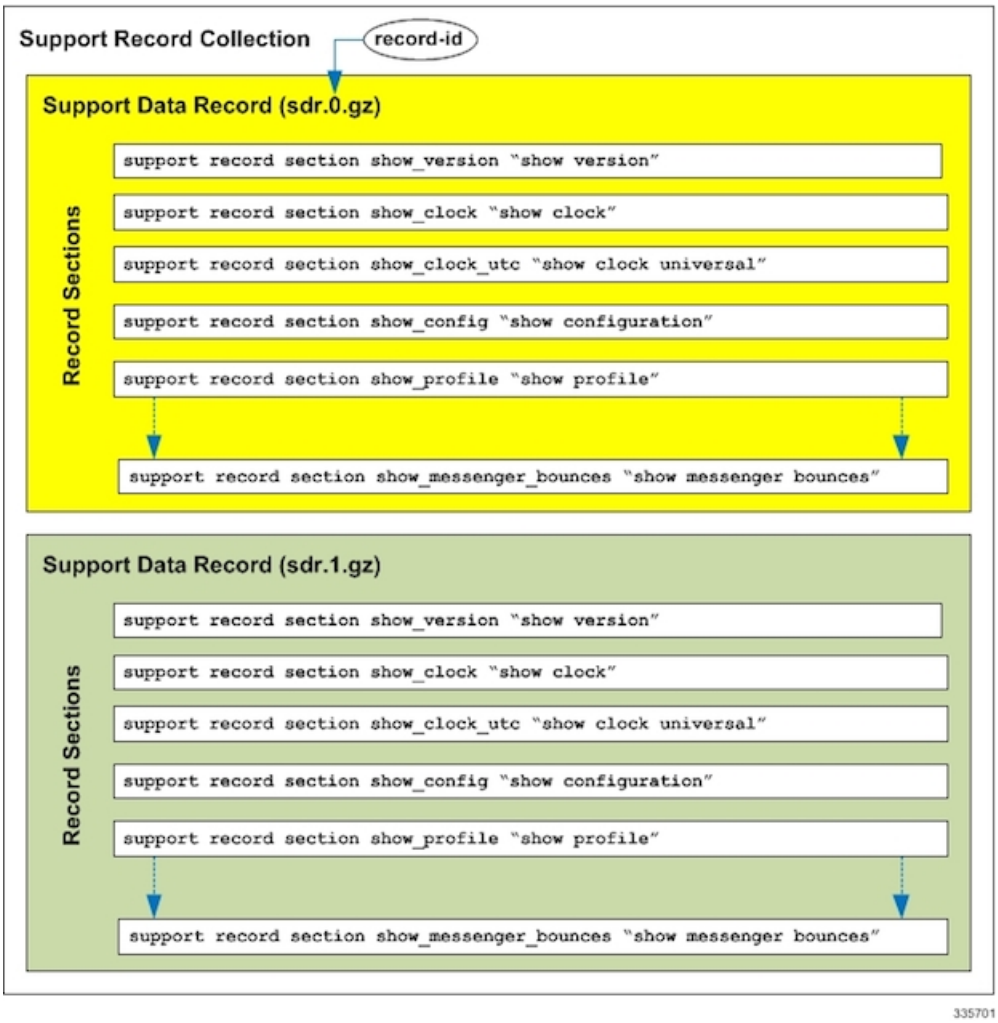
The periodicity of the SDC is configured by the **support collection schedule** command under Global Configuration Mode. Once the SDR is stored, the SDC waits the sleep-duration interval specified via the **support collection** command before collecting another SDR.

Important The period between SDRs is equal to the configured sleep-duration interval + the time taken to collect the previous record.

Managing Record Collection

The SDRs are stored together in a self-relative set. This self-relative set is called a Support Record Collection. Each individual SDR is identified with a record-id. The record-id of the most recent SDR is always 0 (zero). The next older SDR is record-id 1, and so on, for the number of records in the stored collection. For example, if there are five SDRs, they are identified as SDR-0 through SDR-4.

Figure 22: Support Data Collection Hierarchy



When a new SDR is created, the numbers all increment by one and the newest SDR is given the value of 0. If the total number of records exceeds a configured maximum, then the oldest SDR is deleted.

Using the example above, when the maximum SDR count of 5 is reached, the SDRs continue to be SDR-0 through SDR-4, with the file timestamps indicating that the files are changing over time.

The time interval between collections may vary by several minutes in relation to the specified sleep-duration. This is because the interval specifies the idle time between scheduled collection runs. Since the actual overhead of the collecting process is not included in the scheduled intervals, the time differences between collections includes this non-deterministic amount of time.

Important Using a shorter interval to compensate for this behavior is not recommended, since it will only add to the overhead incurred by the collection process and will ultimately impact the overall system performance. The sleep-duration (idle-time) between scheduled collections is an important component of the "self-throttling" mechanism that should not be circumvented by the user.

The Exec Mode **show support collection** command displays useful information about the Support Data Collector. The output includes information about when the collector last ran, how long it took to run, when it is scheduled to run again, as well as the number of SDRs currently stored, where they are stored, and how much storage space is being used. Refer to Exec Mode Commands, on page 435 for more detail about this command.

Using SDRs to Diagnose Problems

The user can compare the SDRs by examining two or more in sequence. These SDRs are dumped out in their CLI-formatted output display. Comparing the display outputs reveals trends and performance or configuration differences that indicate problem areas.

Once specific record sections have been identified as having problematic characteristics, only the CLI **show** commands associated with those sections need be monitored and compared to further isolate the problem areas. In addition, individual SDRs may be transferred via system-supported protocols to remote system, or the current collection may be transferred as a set for later analysis.

SDR CLI Commands

You may use the collected support data records to view support data chronologically. If the default list and sequence of sections is inadequate for system monitoring, you can configure your own set of record section commands that make up a particular support record.

Important Refer to the *SDR CLI Command Strings* appendix for a listing of supported CLI strings (**show** commands) for record sections. The listing also identifies the CLI strings supported as default record sections. You can obtain the same listing by running the **show support collection definitions** command.

Important You may enter up to 200 SDR CLI strings in a single record section command. If you attempt to add more than 200 CLI strings, an error message appears. You may also receive an error message if the system is unable to parse all of the requested CLI strings because they are too complicated to parse.

After configuring the SDR you then configure the sleep-duration interval between record collections and the number of historical records to be retained before being overwritten. By default, configuring this collection information makes the collector mechanism active (if not already active).

After one or more collection intervals have passed, the SDR data becomes available for analysis. The administrator can then use CLI commands to examine the SDR information to perform root cause analysis and trend analysis based on how the data has changed over time. The administrator may decide to transfer the SDRs off the system to be analyzed remotely, for example, by Cisco TAC.

For complete descriptions of the CLI commands discussed below, refer to the *Command Line Interface Reference*.

Configuration Commands (Global Configuration Mode)

support record

support record section *section-name* **command** "*command-string*" **[section** *section-name* **command** "*command-string*" **] ...**

no support record [all | section *section_name* **]**

default support record [all | section *section_name* **]**

The **support record section** command configures a specific record section or set of record sections for a support information output command. The order in which record sections are saved is fixed, regardless of the sequence in which the CLI commands were entered.

For example:

[local]*host_name*(config)# **support record section show_context command "show context"**

If the **support record section** command is not explicitly configured by the user, a default set of record section commands are used. These default record section commands are displayed when you run the **show configuration verbose** command. If support record section commands are explicitly configured, they replace the default commands.

Important Refer to the *SDR CLI Command Strings* appendix for a listing of supported CLI strings (**show** commands) for record sections. The listing also identifies the CLI strings included in default record sections.

The **no support record** command removes either a specific section of the record definition or all of the sections. If you specify the **default support record** command, the default record section definition of that specified record section is used. If neither the keyword **all** or **section** is specified, all the record section definitions are removed.

support collection

support collection [sleep-duration [hours *h* **| minutes** *m* **]] [max-records** *n* **]**

no support collection

default support collection

The **support collection** command modifies and/or enables the support collection process. If support collection has been previously disabled, this command enables the collection activity. If the support collection is currently enabled, this command may be used to modify the sleep-duration interval and/or the maximum number of SDRs that can be collected and stored.

The **sleep duration** keyword specifies the time interval between the collection of support data. It can be specified in hours or minutes with a default of one hour (60 minutes).

The **max-records** keyword specifies the number of SDRs to store as an integer from 1 to 65535. When this value is exceeded, the new SDR overwrites the oldest SDR. The default value is 168.

Important SDR files will be stored in the **/hd-raid/support/records/** directory.

For example:

```
[local]host_name(config)# support collection sleep-duration minute 30
max-records 50
```

Use the **no support collection** command to explicitly disable the collection of the SDRs. If no record section commands are defined, the support data collector mechanism is also effectively disabled.

Use the **default support collection** command to enable the support data collector using the default record sections.

Exec Mode Commands

show support record

show support record *record-id* **[to** *record-id* **] [section** *section_name* **]**

The **show support record** command displays a collection of SDRs. The SDRs are displayed in order from lowest record-id to highest record-id.

Each SDR is identified by a time index called the record-id. For example, the most recent record is always record-id 0 (filename = sdr.0.gz). The next older record is record-id 1 (filename = sdr.1.gz), and so on.

When a new record is collected it is given a record-id of 0. The previously most recent record is renamed to record-id 1, and so on. The display includes the record-id along with the collection time-stamp.

The *record-id* variable identifies a single SDR. The **to** keyword specifies the endpoint record-id when displaying a range of SDRs.

The **section** keyword displays a particular section of the record.

delete support record

delete support record *record-id* **[to** *record-id* **]**

The **delete support records** command removes an SDR with a specified record-id or all SDRs in the specified range of record-ids.

show support collection

show support collection [definitions]

The **show support collection** command displays information on SDC activity. It display informations such as the start time of the last scheduled collection, the duration of the last scheduled collection, whether the collection is still in progress, etc. In addition this command lists the currently stored set of SDR record-ids, their respective timestamps, and size of each SDR.

```
[local]host_name# show support collection
  Record Collection Enabled  : yes
  Last Collection Start Time : Monday October 21 06:29:05 PDT 2013
  Last Collection End Time   : Monday October 21 06:29:09 PDT 2013
  Est. Collection Next Start : Monday October 21 07:29:13 PDT 2013 (40 minutes)
```

```
Support Data Records at /var/tmp/support-records/
  ID        Name      Size       Date/Time
 167  sdr.167.gz     42863       Monday October 21 04:40:00 PDT 2013
 166  sdr.166.gz    170425       Monday October 21 05:40:08 PDT 2013
total SDRs 2, total bytes 2132880, time span is last 1 day(s) 1 hour(s)
```

The optional **definitions** keyword displays the list of default support record section definitions. This is the list of all valid record section definitions. The display also indicates whether the record section is enabled or disabled by default.

```
[local]host_name# show support collection definitions
```

The output of this command reflects the sequence in which record sections will be output, regardless of the sequence in which they may have been entered by the user. Refer to the *SDR CLI Command Strings* appendix for additional information.

APPENDIX A

Engineering Rules

This appendix provides engineering guidelines for configuring the system to meet network deployment requirements.

Note The Engineering Rules listed in this appendix reflect the maximum capacity of StarOS. The actual limits of VPC running in a VM are governed by the amount of vCPU and vMemory capacity allocated to the instance.

CLI Session Rules

Multiple CLI session support is based on the amount of available memory. The internal Resource Manager reserves enough resources to support a minimum of six CLI sessions at all times. One of the six sessions is further reserved for use exclusively by a CLI session on the serial interface.

Additional CLI sessions beyond the pre-reserved limit are permitted if sufficient resources are available. If the Resource Manager is unable to reserve resources for a CLI session beyond those that are pre-reserved, users with administrator-privileges are prompted to create the new CLI session, even without reserved resources.

VPC-DI Interface and Port Rules

The rules discussed in this section pertain to the vNIC Ethernet ports used for context traffic on Service Function (SF) VMs.

vNIC Ethernet Ports

- Give all logical interfaces a unique name to identify the interface from others in the same context. Logical interfaces in different contexts may have the same name.
- A single virtual port can support multiple logical interfaces when you configure VLAN tags for that port. You can use VLAN tagging to bind a single port to multiple logical interfaces that reside in different contexts.
- Assign all logical interfaces a valid IP address and subnet.
  - Give each logical interface within a context a unique IP address. Logical interfaces in different contexts can have the same IP address.
  - If multi-homing is supported on the network, you can assign all logical interfaces a single primary IP address and up to 16 secondary IP addresses.
  - Shared subnets are configurable for both primary and secondary IP addresses. For example, this configuration is valid:

```
interface intf1
   ip address 190.20.20.2 255.255.255.0
   ipv6 address 2001::34/64 secondary
   exit
interface intf2
   ip address 190.20.20.3 255.255.255.0
   ipv6 address 2001::35/64 secondary
   exit
interface intf3
   ipv6 address 2001::36/64
   ip address 190.20.20.4 255.255.255.0 secondary
   exit
```

- You can configure a logical interface in only one context, but you can configure multiple interfaces (up to 512) in a single context.
- You can apply a maximum of 256 access control list (ACL) rules to a single logical interface.
- All ports are identified by their *slot*/*port*. For VPC-DI, the *slot* corresponds to a CF or SF VM within the virtual chassis (VPC-DI instance). The hypervisor assigns a unique slot number to each VM during initial configuration of the VPC-DI instance. Slots 1 and 2 are assigned to the CF VMs slot numbers 3 to 32 are assigned to SF VMs. The CF only supports port 1.

  Each SF supports 12 vNICs numbered 1 to 12 with corresponding virtual Ethernet ports numbered 10 to 21. SF port number 10 must be configured.
- Each vNIC port for subscriber traffic may contain up to a maximum of 1,024 VLAN tags.
- A logical interface is limited to using a single VLAN on a single physical port, identified by its <*card*/*slot*/*port*>.

Packet Data Network (PDN) Interface Rules

The following engineering rules apply to the interface to the packet data network (PDN):

- Configure the logical interfaces used to facilitate the PDN interface within the egress context.

- The default is to use a single interface within the egress context to facilitate the PDN interface.
- You can configure multiple interfaces in the egress context by using static routes or dynamic routing protocols.
- You may also configure next-hop default gateways.

Context Rules

- A maximum of 63 contexts may be configured per chassis.Enabling demux functions on an MIO card reduces the maximum number of contexts to 10.
- Interfaces per Context
  - *Prior to Release 15.0:* Up to 16 interfaces can be configured within a single context.
  - *For Release 15.0 and higher:* With the Demux MIO/UMIO/MIO2 feature enabled, up to 64 interfaces can be configured within a single context.
  - 512 Ethernet+PPP+tunnel interfaces
  - 32 ipv6ip tunnel interfaces
  - 511 GRE tunnels (2,048 GRE tunnels per chassis)
  - 256 loopback interfaces
- IP Addresses and IP Address Pools
  - Up to 2,000 IPv4 address pools can be configured within a single context.
  - *Prior to Release 15.0:* Up to 32 IPv6 pools can be configured within a single context.
  - *For Release 15.0 and higher:* Up to 256 IPv6 pools can be configured within a single context.
  - Up to a combined total of 5,000 IPv4 and IPv6 addresses can be configured per chassis.
  - Each context supports up to 32,000,000 static IP pool addresses. You can configure a maximum total of 96,000,000 static IP pool addresses per chassis. Each static IP pool can contain up to 500,000 addresses.
  - Each context supports up to 16,000,000 dynamic IP pool addresses. You can configure a maximum total of 32,000,000 dynamic IP pool addresses per chassis. Each dynamic IP pool can contain up to 500,000 addresses.

Important The actual number of IP Pools supported per context and chassis depends on how many addresses are being used and how they are subnetted.

Important Each address in the pool requires approximately 60 bytes of memory. The amount of memory required, however, depends on a number of factors such as the pool type, and hold-timer usage. Therefore, in order to conserve available memory, you may need to limit the number of pools depending on the number of addresses to be configured and the number of installed application cards.

- The maximum number of simultaneous subscriber sessions is controlled by the installed capacity license for the service(s) supported.
- The maximum number of static address resolution protocol (ARP) entries per context is 128.
- The maximum number of domains per context is 2,048.
- ASN-GW services configured within the same context cannot communicate with each other.
- Routes
  - Up to 1,200 static routes per context (48,000 per chassis).
  - 6,000 pool routes per context (6,000 per chassis)
  - *Releases prior to 18.5:* 5,000 pool explicit host routes per context (6,000 per chassis)
  - *Release 18.5 and higher:* 24,000 pool explicit host routes per context (24,000 per chassis)
  - 64 route maps per context
- BGP
  - *Releases 12 and 14:* 16,000 BGP prefixes can be learned/advertised per context (64,000 per chassis)
  - *Releases 15 and 16:* 32,000 BGP prefixes can be learned/advertised per context (64,000 per chassis)
  - *Releases 17, 18 and higher:* 64,000 BGP prefixes can be learned/advertised per context (64,000 per chassis)
  - 64 EBGP peers can be configured per context (512 per chassis)
  - 16 IBGP peers per context
  - 512 BGP/AAA monitors per context in support of Interchassis Session Recovery (ICSR)
- OSPF
  - 200 OSPF neighbors per chassis
  - 10,000 OSPF routes per context (64,000 per chassis)
- MPLS

  From Release 19.x to Release 21.6
  - 16 label distribution protocol (LDP) sessions per context
  - Up to 8,000 incoming label map (ILM) entries per context (48, 000 per chassis)

- Combine Table size of 128,000 next hop label forwarding entries (NHLFE) and 64,000 prefixes that could potentially yield:
  - 1,000 forwarding equivalence class (FEC) entries per context (4,000 per chassis) - with 32 paths
  - 2,000 forwarding equivalence class (FEC) entries per context (8,000 per chassis) - with 16 paths
  - 16,000 forwarding equivalence class (FEC) entries per context (64,000 per chassis) - with 2 paths
  - 64,000 forwarding equivalence class (FEC) entries per context (64k per chassis) - with 1 path

- *Release 21.7 and higher*
  - 16 label distribution protocol (LDP) sessions per context
  - Up to 8,000 incoming label map (ILM) entries per context (48,000 per chassis)
  - Combine Table size of 256,000 next hop label forwarding entries (NHLFE) and 64,000 prefixes that could potentially yield:
    - 1,000 forwarding equivalence class (FEC) entries per context (4,000 per chassis) - with 64 paths
    - 2,000 forwarding equivalence class (FEC) entries per context (8,000 per chassis) - with 32 paths
    - 32,000 forwarding equivalence class (FEC) entries per context (64,000 per chassis) - with 2 paths
    - 64,000 forwarding equivalence class (FEC) entries per context (64,000 per chassis) - with 1 path

- VRF
  - *Prior to Release 15.0:* 250 virtual routing and forwarding (VRF) tables per context (1,024 or 2,048 [release 14.0+] VRFs per chassis)
  - *Release 15.0 and higher:* 300 virtual routing and forwarding (VRF) tables per context (2,048 VRFs per chassis) [256 VRFs per context with demux functions enabled on the MIO card]
  - APN limit is 2,048 per chassis; VRF limits and APN limits should be identical.
  - 64,000 IP routes

- NEMO (Network Mobility)
  - *Prior to Release 15.0:* 256K prefixes/framed routes per chassis and up to 8 dynamically learned prefixes per MR (Mobile Router)
  - *Release 15.0 and higher:* 512K prefixes/framed routes per chassis and up to 16 dynamically learned prefixes per MR (Mobile Router)

- 128 AAA servers per context for a default AAA server group. The servers can be configured as accounting, authentication, charging servers, or any combination thereof.

- You can configure up to 800 AAA server groups per context with following limitations:
  - 128 servers per AAA server group (accounting, authentication, charging server, or any combination thereof)
  - 1,600 servers per context in AAA Server group mode (accounting, authentication, charging server, or any combination thereof)
  - 800 NAS-IP address/NAS identifier (one primary and one secondary per server group) per context
- Up to 12 charging gateway functions (CGFs) for GTPP accounting can be configured per context.
- Up to 16 bidirectional forwarding detection (BFD) sessions per context (64 per chassis)

Important Refer to *Engineering Rules* in your product administration guide for additional information on product-specific operating limits.

Subscriber Rules

The following engineering rules apply to subscribers configured within the system:

- Configure a maximum of 2,048 local subscribers per context.
- You may configure attributes for each local subscriber.
- The system creates a default subscriber for each context when the context is made. Configure attributes for each default subscriber. If a AAA-based subscriber is missing attributes in the authentication reply message, the default subscriber attributes in the context where the subscriber was authenticated are used.

  Important Default is not used when local authentication (for local subscribers) is performed.

- Configure default subscriber templates on a per AAA realm (domain aliases configured within a context) basis.
- Configure default subscriber templates on a per PDSN, FA, ASN-GW, or HA service.
- For AAA authenticated subscribers, the selection of local subscriber template to use for setting attributes is in the following order:
  - If the username (NAI) matches any local domain name and the domain name has a local subscriber name configured, that local subscriber template is used.
  - If the first case fails, and if the serving service has a default username configured, that subscriber template is used.
  - If the first two cases fail, the default subscriber template in the AAA context is used.

Service Rules

The following engineering rules apply to services configured within the system:

- Configure a maximum of 256 services (regardless of type) per system.

  Caution Large numbers of services greatly increase the complexity of management and may affect overall system performance. Therefore, you should not configure a large number of services unless your application absolutely requires it. Please contact your Cisco service representative for more information.

- The total number of entries per table and per chassis is limited to 256.
- Although you can use service names that are identical to those configured in different contexts on the same system, this is not a good practice. Services with the same name can lead to confusion and difficulty in troubleshooting problems, and make it difficult to understand the output of **show** commands.

Access Control List (ACL) Engineering Rules

The following rules apply to Access Control Lists:

- The maximum number of rules per ACL is 128.
- The maximum number of ACL rules applied per port is 128.
- The maximum number of ACL rules applied per context is 1,024.
- The maximum number of ACL rules per IPSec policy is 1.
- The maximum number of IPSec ACL rules per context is 1,024.
- The maximum number of IPSec ACL rules per crypto map is 8.
- The maximum number of ACLs you can configure per context is limited by the number of rules allowed within each ACL. If each ACL contained the maximum number of rules (128), the maximum number of ACLs per context is 8 (128 X 8 ACLs = 1,024 ACL rules per context).
- The maximum number of ACLs applied to an IP access group is 1, whether it is configured for a port or context. Since the maximum number of IP access groups you can apply to an interface or context is 16, the following calculations apply:
  - For each interface/port: 8 rules per ACL multiplied by 16 IP access groups = 128 (the ACL rules limit per port)
  - For each context: 64 rules per ACL multiplied by 16 IP access groups = 1,024 (the ACL rules limit per context)

ECMP Groups

The maximum number of Equal Cost Multiple Path (ECMP) groups are as follows:

- *For releases prior to 17.0*, StarOS supports a maximum of 512 groups.
- *For release 17.0 and higher*, StarOS supports a maximum of 2048 groups.

Note

- *max_num* is an integer from 1 through 10 (*releases prior to 18.2*)

  Release 18.2x

  - QVPC-DI: 32
  - QVPC-SI: 32

  Release 21.4x

  - QVPC-DI: 64
  - QVPC-SI: 64
  - ASR 5500: 24

- Save your configuration as described in the Verifying and Saving Your Configuration, on page 137 chapter.

- *For release 20.0 and higher*, StarOS supports a maximum of 64000 groups.

VPC-DI Network Dependencies

This section outlines the services and network devices required to use functionality of StarOS gateways in a VPC-DI instance.

Routers

Routers supporting the VPC-DI network must support:

- IPv4 and IPv6 interfaces (NOTE: Within each routing context, StarOS requires each IP interface to be in a unique subnet.)
- Static routes
- *Optional:* OSPFv2 and/or OSPFv3 (Requires multicast support for IPv4/IPv6)
- eBGP (External Border Gateway Protocol)
  - Multihop
  - Multipath

  - IPv4 address Family
  - IPv6 address Family
  - *Optional:* VPNv4 and VPNv6 for Enterprise style configurations
  - Via eGBP these routers would learn IP pools, routes, service interface addresses, etc.
- Routing Policies to filter the routes.
- MPLS encapsulation and forwarding (NOTE: required only for enterprise style configurations.)
- ICMP and ICMPv6 support
- BFD (Bidirectional Forwarding Detection) [for quicker detection of failures]
- ECMP (Equal Cost Multiple Path)

External Network Dependencies

- RADIUS or Diameter Services
- DNS Services
- NTPv4 Services
- SNMP services
- syslog services
- SSH/SFTP services (for bulkstats and administration)
- *Optional:* TACACS+ services
- Billing Services (CDR/EDR)
- VLAN encapsulation
- ICSR requirements:
  - Dedicated connectivity between primary and backup virtual chassis
  - eBGP
  - Layer 2 L2 Switch functionality between two virtual chassis (for processing GARP/Mac Address Changes, when chassis state is changed)

VPN Scaling Requirements

The following VPN scaling numbers are supported for specific releases.

Parameter	Scaling Number (Release 12.x, 14.x)	Scaling Number (Release 15.x, 16.x)	Scaling Numbrer (Release 17.x, 18.x, 19.x, 20.x +)
BFD Sessions	16 per context 64 per chassis	16 per context 64 per chassis	16 per context 64 per chassis
Context level ACLs	256 per context	256 per context	256 per context
Dynamic pool addresses	16 million per context 32 million per chassis	16 million per context 32 million per chassis	16 million per context 32 million per chassis

Parameter	Scaling Number (Release 12.x, 14.x)	Scaling Number (Release 15.x, 16.x)	Scaling Numbrer (Release 17.x, 18.x, 19.x, 20.x +)
IPv4 pools per context	2000 per context 5000 per chassis (combined IPv4 and IPv6)	2000 per context 5000 per chassis (combined IPv4 and IPv6)	2000 per context 5000 per chassis (combined IPv4 and IPv6)
IPv6 pools per context	32 per context 5000 per chassis (combined IPv4 and IPv6)	256 per context 5000 per chassis (combined IPv4 and IPv6)	256 per context 5000 per chassis (combined IPv4 and IPv6)
Number of BGP prefixes	16,000 per context 64,000 per chassis.	32,000 per context 64,000 per chassis.	64,000 per context 64,000 per chassis.
Number of contexts	63 (though PSC migrations do not work well beyond 32 contexts)	63 (though PSC migrations do not work well beyond 32 contexts) Note the information about "Demux on MIO Cards" at the end of this section.	63 (though PSC migrations do not work well beyond 32 contexts) Note the information about "Demux on MIO Cards" at the end of this section.
Number of dynamically learned prefixes per MR	8	16	16
Number of EBGP peers	64 per context 512 per chassis	64 per context 512 per chassis	64 per context 512 per chassis
Number of FEC entries	8000 labels per context 48,000 per chassis	8000 labels per context 48,000 per chassis	8000 labels per context 48,000 per chassis
Number of IBGP peer	16 per context	16 per context	16 per context
Number of ILM entries	8000 labels per context 48,000 per chassis	8000 labels per context 48,000 per chassis	8000 labels per context 48,000 per chassis

Parameter	Scaling Number (Release 12.x, 14.x)	Scaling Number (Release 15.x, 16.x)	Scaling Numbrer (Release 17.x, 18.x, 19.x, 20.x +)
Number of interfaces	512 ethernet+ppp+tunnel interfaces per context 32 IPv6 IP tunnel interfaces per context Upto 511 GRE tunnels/context and 2048 GRE tunnels/chassis 256 loopback interfaces per context	512 ethernet+ppp+tunnel interfaces per context 32 IPv6 IP tunnel interfaces per context Upto 511 GRE tunnels/context and 2048 GRE tunnels/chassis 256 loopback interfaces per context Note the information about "Demux on MIO Cards" at the end of this section.	512 ethernet+ppp+tunnel interfaces per context 32 IPv6 IP tunnel interfaces per context Upto 511 GRE tunnels/context and 2048 GRE tunnels/chassis 256 loopback interfaces per context Note the information about "Demux on MIO Cards" at the end of this section.
Number of LDP sessions	16 per context	16 per context	16 per context
Number of NEMO prefixes/Framed routes	256,000 per chassis	512,000 per chassis	512,000 per chassis
Number of OSPF neighbors	Up to 200 per chassis	Up to 200 per chassis	Up to 200 per chassis
Number of OSPF routes	Up to 10,000 per context 64,000 per chassis	Up to 10,000 per context 64,000 per chassis	Up to 10,000 per context 64,000 per chassis
Number of pool explicit host routes	5000 per context (6000 per chassis)	5000 per context (6000 per chassis)	5000 per context (6000 per chassis) in 17.x and 18.[1234] 24000 per context (24000 per chassis) in 18.5 and above
Number of pool routes	6000 per context (6000 per chassis)	6000 per context (6000 per chassis)	6000 per context (6000 per chassis)
Number of routes (excluding framed-routes)	64,000 per context	64,000 per context	64,000 per context
Number of secondary addresses per interface	16	16	16
Number of Static routes	1200 per context	1200 per context	1200 per context
Number of VLANs	4000 per chassis	4000 per chassis	4000 per chassis

Parameter	Scaling Number (Release 12.x, 14.x)	Scaling Number (Release 15.x, 16.x)	Scaling Numbrer (Release 17.x, 18.x, 19.x, 20.x +)
Number of VRFs	250 per context 2048 per chassis APN limit is 1024/chassis and does not match the VRF limit.	300 per context 2048 per chassis **Note** • VRF Limits and APN limits are assumed to be identical. • Note the "Demux on MIO Cards" section at the end of this section.	300 per context 2048 per chassis Note: VRF Limits and APN limits are assumed to be identical. Note the "Demux on MIO Cards" section at the end of this section.
Number of routes (all kinds of routes including framed routes)	64,000 per context	64,000 per context	64,000 per context
Route maps	64 per context	64 per context	64 per context
Static pool addresses	32 million per context 96 million per chassis	32 million per context 96 million per chassis	32 million per context 96 million per chassis

Demux on MIO Cards

When enabling Demux on MIO cards, VPN resources are combined on the MIO cards with the controller processes, thus reducing the resources available for all VPN tasks. This results in reducing some of the limits (mentioned in the previous section) when MIO cards are demux-enabled.

APPENDIX B

StarOS Tasks

This appendix describes system and subsystem tasks running under StarOS on an ASR 5500 and virtualized platforms.

Important This appendix is not a comprehensive list of all StarOS tasks. It simply provides general descriptions of the primary tasks and subsystems within StarOS.

It includes the following sections:

Overview

For redundancy, scalability and robust call processing, StarOS supports a series of tasks that perform specific functions. These tasks communicate with each other as needed to share control and data signals. As a result, processes can be distributed across multiple tasks thus reducing the overall work-load on any given task and improving system performance. This distributed design provides fault containment that greatly minimizes the impact to processes or sessions due to a failure.

The Exec mode **show task** command displays snapshots of running processes within StarOS. For detailed information about this command, see the *Command Line Interface Reference* and *Statistics and Counters Reference*.

The following sections describe the primary tasks that are implemented by StarOS:

Primary Task Subsystems

The individual tasks that run on the CPUs are divided into subsystems. Following is a list of the primary subsystems responsible for call session processing:

- **System Initiation Task (SIT):** This subsystem starts tasks and initializes the system. This includes starting a set of initial tasks at system startup time (static tasks), and starting individual tasks on demand at arbitrary times (dynamic tasks).
- **High Availability Task (HAT):** With the Recovery Control Task (RCT) subsystem, the HAT subsystem maintains the operational state of the system. HAT monitors the various software and hardware components of the system. If there are unusual activities, such as the unexpected termination of another task, the HAT subsystem takes a suitable course of action, such as triggering an event to the RCT subsystem to take corrective action or to report the status. The primary function of the HAT task is to minimize service impacts.
- **Recovery Control Task (RCT):** This subsystem executes a recovery action for any failure that occurs in the system. The RCT subsystem receives signals from the HAT subsystem (and in some cases from the NPU subsystem) and determines what recovery actions are needed.
- **Shared Configuration Task (SCT):** This subsystem provides a facility to set, retrieve, and receive notification of system configuration parameters. The SCT is mainly responsible for storing configuration data for the applications that run on the system.

  The SCT subsystem runs only on the active management card and synchronizes the information it contains with the SCT subsystem on the standby management card.
- **Resource Management (RM):** This subsystem assigns resources, such as CPU loading and memory, for every system task upon start-up. The RM subsystem monitors resource use to verify that allocations are as specified. RM also monitors all sessions and communicates with the Session Controller to enforce capacity licensing limits.
- **Virtual Private Network (VPN):** This subsystem manages the administrative and operational aspects of all VPN-related entities in the system. The functions performed by the VPN subsystem include:
  - Creating separate VPN contexts
  - Starting the IP services within a VPN context
  - Managing IP pools and subscriber IP addresses, and distributing the IP flow information within a VPN context.

  All IP operations within StarOS are done within specific VPN contexts. In general, packets are not forwarded across different VPN contexts. The only exception currently is the Session subsystem.
- **Network Processing Unit** (npusim on ASR 5500, and iftask or knpusim on VPC-DI and VPC-SI)[1]: This subsystem is responsible for the following:
  - Using the database to match address and port numbers to destination tasks for fast-path forwarding of dataframes
  - Receiving and transmitting user data frames to/from various physical interfaces
  - IP forwarding decisions (both unicast and multicast)
  - Per-interface packet filtering
  - Traffic management and traffic engineering
  - Passing user data frames to/from packet processing CPUs

[1] knpusim runs instead of iftask on VPC VMs that do not have Intel DPDK supported configurations.

- Modifying, adding, or stripping datalink/network layer headers
- Recalculating checksums
- Maintaining statistics
- Managing external Ethernet interfaces

- **Card/Slot/Port (CSP):** Coordinates the events that occur when any card is inserted, locked, unlocked, removed, shutdown, or migrated. CSP also performs auto-discovery and configures ports on a newly-inserted interface card. It determines how interface cards map to packet processing cards.

  The CSP subsystem runs only on the active management card and synchronizes the information it contains with the SCT subsystem on the standby management card. It is started by the SIT subsystem and monitored by the HAT subsystem.

- **Session Manager (SM):** Performs high-touch processing of mobile subscribers' packet-oriented data session flows. High-touch user data processing consists of the following:
  - Payload transformation
  - Filtering and scheduling
  - Statistics collection
  - Policing

Controllers and Managers

Many of the primary subsystems are composed of controller tasks called Controllers, and subordinated tasks called Managers.

Controllers serve several purposes:

- Monitor the state of their Managers and allow communication between Managers within the same subsystem.
- Enable inter-subsystem communication since they can communicate with the controllers of other subsystems.
- Mask the distributed nature of the software from the user allowing for ease of management.

Managers manage resources and mappings between resources. In addition, some managers are directly responsible for call processing.

For information about the primary subsystems that are composed of critical, controller, and /or manager tasks, see Subsystem Tasks, on page 451.

Subsystem Tasks

The following subsections list and briefly describe StarOS tasks for various subsystems:

- System Initiation Subsystem, on page 452

System Initiation Subsystem

Table 30: System Initiation Subsystem Tasks

Task	Description	Function
SITMAIN	System Initiation Task – Main	Initiated at system start-up.
		Reads and provides startup configuration to other SIT components.
		Starts SITREAP sub-function.
		Maintains CPU state information.
SITPARENT	SIT Parent Sub-function	Starts management cards in either active or standby mode.
		Registers tasks with HAT task.
		Notifies CSP task of CPU startup completion.
		Brings up packet processing cards in standby mode.
SITREAP	SIT Reap Sub-function	Shuts down tasks as required.

High Availability Subsystem

Table 31: High Availability Subsystem Tasks

Task	Description	Function
hatcpu	High Availability Task CPU	Performs device initialization and control functions based on the CPUs hardware capabilities.
		Reports the loss of any task on its CPU to hatsystem sub-function.
		Controls the LEDs on the packet processing cards.
		Initializes and monitors the dedicated hardware on packet processing cards. (ASR 5500 only)
		Collects CPU monitoring information periodically and reports to the master hatcpu sub-function.
		Reports the loss of any task on its CPU to the master hatcpu sub-function.
		Performs device initialization and control functions because of the CPU's hardware capabilities.
		Reports the loss of any task on its CPU to hatsystem sub-function.
		Controls the LEDs on the management card. (ASR 5500 only)
		Initializes and monitors the dedicated hardware on the management card. (ASR 5500 only)
hatsystem	High Availability Task System Controller	Controls all the HAT sub-function tasks in the system. It is initiated on system start-up.
		Initializes system components (such as the Gigabit Ethernet switches and switch fabric).
		Monitors system components such as fans for state changes.
		Triggers actions for redundancy in the event of fault detection.
		The HAT subsystem on the redundant management card mirrors the HAT subsystem on the active management card.

Resource Manager Subsystem

Table 32: Resource Manager (RM) Subsystem Tasks

Task	Description	Function
rmctrl	Resource Manager Controller	Started by the sitparent task on StarOS startup, and monitored by HAT for a failure.
		Initializes resources such as CPUs and memory.
		Requests updated card status from the CSP subsystem and updates the system card table.
		Communicates with all rmctrls to request their most recent set of resource data.
rmmgr	Resource Manager Managers	Started by the sitparent task, and monitored by HATs for failures.
		Initializes the local resource data and local resource scratch space.
		Communicates with the SIT task on the local CPU to get its entire task table and the resources associated with each task.
		Gathers current resource utilization for each task.
		Sends the resource data to the rmctrl task.

Virtual Private Networking Subsystem

Table 33: Virtual Private Networking (VPN) Subsystem Tasks

Task	Description	Function
vpnctrl	VPN Controller	Created at system start-up.
		Initiates the VPN Manager for each context.
		Informs the Session Controller task when there are additions or changes to contexts. Only one Session Controller operates at any time.
		Routes context specific operation information to the appropriate VPN Manager.
		Performs VPN Manager recovery and saves all VPN-related configuration information in SCT.
vpnmgr	VPN Manager	Started by the VPN Controller for each configured context (one is always present for the local context).
		Performs IP address pool and subscriber IP address management.
		Performs all context specific operations including but not limited to: UCM services, IP interfaces, the Address Resolution Protocol (ARP), IP address pool management, slow path forwarding, NPU flows, port Access Control Lists (ACLs), and logging.
		Provides IP interface address information for each context to the Session Controller.
bgp	Border Gateway Protocol	Created by the VPN Manager for each context that has enabled the BGP routing protocol (**router bgp** Context Configuration mode CLI command).
		Responsible for learning and redistributing routing information via the BGP protocol.
		Maintains the BGP peering connections.
		Applies any defined BGP routing policy.

Task	Description	Function
ospf	Open Shortest Path First	Created by VPN Manager for each context that has enabled the OSPF routing protocol (**router ospf** Context Configuration mode CLI command).
		Responsible for learning and redistributing routing information via the OSPF protocol.
		Maintains the OSPF neighboring relationship.
		Maintains the LSA database.
		Performs SPF calculations.
		Applies any defined OSPF routing policy
ospfv3	Open Shortest Path First	Created by VPN Manager for each context that has enabled the OSPFv3 routing protocol (**router ospfv3** Context Configuration mode CLI command)
		Responsible for learning and redistributing routing information via the OSPFv3 protocol.
		Maintains the OSPFv3 neighboring relationship.
		Maintains the LSA database.
		Performs OSPFv3 SPF calculations.
		Applies any defined OSPFv3 routing policy.
rip	Routing Information Protocol	Created by VPN Manager for each context that has enabled the RIP routing protocol (**router rip** Context Configuration mode CLI command)
		Responsible for learning and redistributing routing information via the RIP protocol.
		Maintains the RIP database.
		Sends periodic RIP update messages.
		Applies any defined RIP routing policy.

Task	Description	Function
zebos	L2 and L3 Switching	Created by VPN Manager for each context.
		Maintains the routing table (RIB and FIB) for the context.
		Performs static routing.
		Interfaces to the kernel for routing & interface updates.
		Redistributes routing information to dynamic routing protocols.
		Calculates nexthop reachability.

Network Processing Unit Subsystem

Table 34: Network Processing Unit (NPU) Subsystem Tasks

Task	Description	Function
iftask	Internal Forwarder Task (Intel DPDK) [VPC-DI, VPC-SI]	Created at StarOS start up.
		Provides port configuration services to the CSP task.
		Provides interface binding and forwarding services to the VPN Manager.
		Provides flow insertion and removal services to Session Manager and AAA Manager tasks.
knpusim	Kernel-based NPU Simulator [VPC-DI, VPC-SI]	Created at StarOS start up.
		Provides port configuration services to the CSP task.
		Provides interface binding and forwarding services to the VPN Manager.
		Provides flow insertion and removal services to Session Manager and AAA Manager tasks.
		Provides recovery services to the NPU Controller.

Task	Description	Function
npuctrl	NPU Controller	Created at StarOS start-up. Only one NPU Controller operates in the system at any time.
		Monitors the state of NPU Managers in the system.
		Registers to receive notifications when NPU Manager crashes.
		Controls recovery operation.
		Provides a centralized location for CLI commands related to NPU Manager state.
npumgr	NPU Manager	Created for every packet processing card that is installed and started.
		Provides port configuration services to the CSP task.
		Provides interface binding and forwarding services to the VPN Manager.
		Provides flow insertion and removal services to Session Manager and AAA Manager tasks.
		Provides recovery services to the NPU Controller.
npusim	NPU Simulator [ASR 5500]	Created for every DPC installed and started.
		Provides port configuration services to the CSP task
		Provides interface binding and forwarding services to the VPN Manager.
		Provides flow insertion and removal services to Session Manager and AAA Manager tasks.
		Provides recovery services to the NPU Controller.

Session Subsystem

Table 35: Session Subsystem Tasks

Task	Description	Function
sessctrl	Session Controller	Created at StarOS start-up. Only one Session Controller instantiated in the system at any time.
		Acts as the primary point of contact for the Session Subsystem. Since it is aware of the other subsystems running within the system, the Session Controller acts as a proxy for the other components, or tasks, that make up the subsystem.
		Starts, configures, and coordinates the efforts of the Session Processing Subsystem sub-managers.
		Works with Resource Manager to start new Session Managers when all existing Session Managers exceed their capacity.
		Receives context information from VPN Managers.
		Distributes IP interface address information to other Session Processing Subsystem sub-managers.
		Manages Enhanced Charging Service (ECS), Content Filtering and URL Blacklisting services.

Task	Description	Function
sessmgr	Session Manager	Created by the Session Controller.
		Provides a subscriber processing system that supports multiple session types.
		Multiple Session Managers can run on a single CPU and/or can be distributed throughout any CPU present in the system.
		A single Session Manager can service sessions from multiple A11 Managers, and from multiple contexts.
		Processes protocols for A10/A11, GRE, R3, R4, R6, GTPU/GTPC, PPP, and Mobile IP.
		Manages Enhanced Charging Service, Content Filtering and URL Blacklisting services.
		Session Managers are paired with AAA Managers.
		Limitation: Any frequent AAAMGR crashes leads to change of AAAmgr instance id and thus a sessmgr can increment to >512. For example, for calls connecting to sessmgr instance 512 and above, the encoding will be 10 bit (during GUTI based MME attach) and decoding is 9 bit (during SGSN-CONTEXT-REQUEST), leading to context-request landing on incorrect sessmgr and failure of lookup.
a11mgr	A11 Manager	Created by the Session Controller for each context in which a PDSN service is configured.
		Receives the R-P sessions from the PCF and distributes them to different Session Manager tasks for load balancing.
		Maintains a list of current Session Manager tasks to aid in system recovery.
		The A11 Manager task is also known as the Signaling De-multiplexing task (SDT).
		With session recovery (SR) enabled, this demux manager is usually established on one of the CPUs on the first active packet processing card.

Task	Description	Function
aaamgr	Authorization, Authentication, and Accounting (AAA) Manager	Paired with Session Managers.
		Performs all AAA protocol operations and functions for subscribers and administrative users within the system.
		Acts as a AAA client to AAA servers.
		Manages GTP Prime (GTP') messaging with charging gateway functions (CGFs).
		Multiple AAA Managers can run on a single CPU and/or can be distributed throughout any CPU present in the system.
		AAA operations for the CLI are done through a AAA Manager running on the active management card.
aaaproxy	Authorization, Authentication, and Accounting (AAA) Proxy Manager	Starts whenever the Global Configuration mode **gtpp single-source** command is configured. When GTPP single-sourcing is enabled, aaaproxy generates requests to the accounting server using a single UDP source port number, instead of having each AAA Manager generate independent requests with unique UDP source port numbers.
		Runs on a demux card when session recovery is enabled. If session recovery is not enabled, the Global Configuration mode **require demux card** command starts aaaproxy on the designated demux card.
		Writes CDRs to a file in its VRAM-disk. The enqueued CDRs are then periodically synchronized with a HDD for transfer.

Task	Description	Function
acsctrl	Active Charging System (ACS) Controller	Active Charging service is defined at the global level and can be utilized through CSS commands from any VPN context. Enable via the Global Configuration mode **active-charging service** CLI command.
		The ACS controller runs on the primary packet processing card and is responsible for managing the ACS service.
		Reads and writes ACS configuration information into SCT.
		The ACS Controller monitors the ACS Manager's recovery process and performs cleanup when redundancy is enabled.
acsmgr	Active Charging System (ACS) Controller	Created by ACS Controller to perform IP session processing for a specific number of flows.
		Sends and receives data through Session Managers.
		Active/Standby acsmgr tasks are created when session recovery (SR) is enabled.
cdrmod	Charging Detail Record Module	Responsible for receiving EDR/UDR records from different ACSMGR instances in the system.
		Responsible for writing the received EDR/UDR records in files using the configured file naming conventions.
dgmbmgr	Diameter Gmb interface Application Manager	Provides Multimedia Broadcast/Multicast Service (MBMS) feature support for GGSN. It is instantiated when an MBMS policy CLI is configured in the GGSN Service configuration mode. dgmbmgr
		Maintains the MBMS UE and bearer contexts.
		Handles the Gmb interface over a Diameter connection to a BMSC Server for MBMS bearer sessions. dgmbmgr recovers by polling all sessmgrs for MBMS session states and recreating the MBMS UE and MBMS bearer context information.

Task	Description	Function
diamproxy	Diameter Proxy	Created by diactrl (which runs as part of vpnctrl) and the number of diamproxy tasks spawned is based on the configuration to use "multiple" or "single" proxies. In instances that a single proxy is configured, only one diamproxy task is spawned for the entire chassis and runs on demux packet processing cards. When multiple proxies are configured, one diamproxy task is run per packet processing card.
		Maintains Diameter base connections to all peers configured in the system.
		Informs applications about any change in the connection status.
		Acts as a pass-through to the messages from application to the Diameter server.
		Just acts as a forwarding agent (does not maintain any queues).
		A single Diameter proxy is used to service multiple Diameter applications.
egtpemgrr	Enhanced GPRS Tunneling Protocol Egress Manager	Created by the Session Controller for each context in which an egtp-service of interface type sgw-egress or MME is configured.
		Handles certain EGTP messages from SGW, PGW.
		Maintains list of current EGTP sessions.
		Maintains list of current Session Manager tasks which aids in session recovery.
		Handles GTP Echo messaging.
		With session recovery (SR) enabled, this demux manager is usually established on one of the CPUs on the first active packet processing card.

Task	Description	Function
egtpimgr	Enhanced GPRS Tunneling Protocol Ingress Manager	Created by Session Controller for each context in which an egtp-service of interface type sgw-ingress or pgw-ingress is configured.
		Receives EGTP sessions from MME/S4 SGSN/SGW and distributes them to different Session Manager tasks for load balancing.
		Maintains list of current EGTP sessions.
		Maintains list of current Session Manager tasks which aids in session recovery.
		Handles GTP Echo messaging.
		With session recovery (SR) enabled, this demux manager is usually established on one of the CPUs on the first active packet processing card.
gtpcmgr	GPRS Tunneling Protocol Control (GTP-C) Message Manager	Created by the Session Controller for each context in which a GGSN service is configured.
		Receives the GTP sessions from the SGSN and distributes them to different Session Manager tasks for load balancing.
		Maintains a list of current Session Manager tasks to aid in system recovery.
		Verifies validity of GTPC messages.
		Maintains a list of current GTPC sessions.
		Handles GTPC Echo messaging to/from SGSN.

Task	Description	Function
gtpumgr	GPRS Tunneling Protocol User (GTP-U Manager	Created by the Session Controller for each context in which a GTPU service is configured. Supported for both GTPUv0 and GTPUv1
		Maintains a list of the GTPU-services available within the context and performs load-balancing (of only Error-Ind) for them.
		Supports GTPU Echo handling.
		Provides Path Failure detection on no response for GTPU echo.
		Receives Error-Ind and demuxes it to a particular Session Manager.
		Serves as the Default GTPU listener. GTPUMGR will process GTPU packets with invalid TEID.
		With session recovery (SR) enabled, this demux manager is usually established on one of the CPUs on the first active packet processing card.
hamgr	Home Agent (HA) Manager	Created by the Session Controller for each context in which an HA service is configured.
		Receives Mobile IP sessions from the Foreign Agents (FAs) and distributes them to different Session Manager tasks.
		Maintains a list of current Session Manager tasks that aids in system recovery.
		Functions as the DemuxMgr – handles all the PMIP signaling packets.
		Functions as the Demuxmgr for MIPv6/MIPv4 HA.
		With session recovery (SR) enabled, this demux manager is usually established on one of the CPUs on the first active packet processing card.

Task	Description	Function
hnbdemux	Home NodeB (HNB) Demux Manager	Started as part of HNB-GW service creation procedure. There is only one hnbdemux in the chassis.
		Distributes incoming Iuh connections to HNB Managers in the system.
		Remains aware of all the active HNB-GW services in the system.
		With session recovery (SR) enabled, this demux manager is usually established on one of the CPUs on the first active packet processing card.
hnbmgr	Home NodeB (HNB) Manager	Starts when an HNB-GW service configuration is detected. There can be multiple instances of this task for load sharing. All HNB Managers have all the Active HNB-GW Services configured and be identical in configuration and capabilities.
		Runs the SCTP protocol stack.
		Handles the SCTP associations.
		Maintains Home-NodeB databases.
		Provides nodal functions for Iuh interface on SCTP protocol.
		With session recovery (SR) enabled, this manager is usually established on one of the CPUs on the first active packet processing card.

Task	Description	Function
imsimgr	International Mobile Subscriber Identity Manager for MME	Starts when an MME service configuration is detected. There is only one instance of this task:
		Selects which SessMgr to use for new subscriber sessions.
		Maintains and reports MME-related demux statistics on events like Attach by IMSI, Attach by GUTI, etc.
		Can interact with the following tasks in the system: - Session Controller - MME Manager - Session Manager
		With session recovery (SR) enabled, this demux manager is usually established on one of the CPUs on the first active packet processing card.
imsimgr	International Mobile Subscriber Identity Manager for SGSN	Started by the Session Controller.
		Selects SessMgr, when not done by linkmgr or sgtpcmgr tasks, for calls sessions based on IMSI/P-TMSI.
		Load-balances across SessMgrs to select one to which a subscriber will be assigned.
		Maintains records for all subscribers on the system.
		Maintains mapping between the IMSI/P-TMSI and SessMgrs.
		With session recovery (SR) enabled, this demux manager is usually established on one of the CPUs on the first active demux packet processing card.
ipsgmgr	IP Services Gateway Manager	Created by the Session Controller.
		In Server mode, acts as a RADIUS server, and supports Proxy functionality.
		In Snoop mode supports snooping RADIUS Accounting messages.
		Load balances requests among different SessMgrs.
		Activates and deactivates sessions.

Task	Description	Function
l2tpdemux	L2TP Demultiplexor Task	Created by the Session Controller when an LNS service is created. Only one L2TPDemux task is invoked for the entire system.
		De-multiplexes and forwards new incoming tunnel create requests to L2TPMgrs.
		Maintains information about current active tunnels in all L2TPMgrs.
		Load balances requests among L2TPMgrs.
		With session recovery (SR) enabled, this demux manager is usually established on one of the CPUs on the first active packet processing card.
l2tpmgr	Layer 2 Tunneling Protocol Manager	Created by the Session Controller for each context in which a LAC or LNS service is configured. Additional managers are created as needed depending on loading.
		Responsible for all aspects of L2TP processing.
		Maintains protocol state machines for all L2TP sessions and tunnels.
		Triggers IPSec encryption for new L2TP tunnels as needed.
		Works with Session Managers to gracefully bring down tunnels.
		With session recovery (SR) enabled, this demux manager is usually established on one of the CPUs on the first active packet processing card.
linkmgr	SS7 Link Manager	Created by the Session Controller when the first SS7RD (routing domain) is activated.
		Multi-instanced for redundancy and scaling purposes.
		Provides SS7 and Gb connectivity to the platform.
		Routes per subscriber signalling across the SS7 (including Iu) and Gb interfaces to the SessMgr.

Task	Description	Function
magmgr	Mobile Access Gateway (MAG) Manager	Created by the Session Controller when the first MAG service is created in a context.
		Sends and receives PMIP control messages (PBU/PBA).
		Adds an NPU flow to receive MIPv6 PBA packets. This flow is identical to the flow used in the HAMgr.
		Maintains the Binding Update List used to keep track of the mobile node's bindings.
		Originates PBU-based on trigger received from the Session Manager during error conditions.
		Receives PBA and forwards it to Session Manager.
		Supports debugging facility – "magmgr" and "mobile-ipv6".
mmgr	SGSN Master Manager	Created upon provisioning of SS7RDs/SCCP-NWs/etc. The Session Controller provides the initial system configuration which includes a detailed description of each distributed protocol layer, its resources sets, and a list of its service user protocol layers and service provider protocol layers.
		Runs as a single instance.
		Handles nodal SS7, Iu, and Gb functionality.
		Implements master linkmgr functionality for SS7 route status aggregation.
		Implements master linkmgr functionality for RNC and BSC status aggregation.
		With session recovery (SR) enabled, this demux manager is usually established on one of the CPUs on the first active demux packet processing card.

Task	Description	Function
mmedemux	Mobility Management Entity Demux Manager	Started as part of MME service creation procedure. There is only one mmedemux in the chassis.
		Distributes incoming S1-MME SCTP connections to mmemgr tasks in the system.
		Remains aware of all the active MME services in the system.
		With session recovery (SR) enabled, this demux manager is usually established on one of the CPUs on the first active packet processing card.
mmemgr	Mobility Management Entity Manager	Starts when an MME service configuration is detected. There can be multiple instances of this task for load sharing. All mmemgrs will have all the Active MME Services configured and will be identical in configuration and capabilities.
		Runs the SCTP protocol stack.
		Handles the SCTP associations.
		Maintains TA List.
		Manage eNodeB databases.
		Provides nodal functions for S1-MME protocol.
		With session recovery (SR) enabled, this demux manager is usually established on one of the CPUs on the first active packet processing card.

Task	Description	Function
pccdemux	Policy and Charging Control Bindmux Manager	Started as part of PCC service creation procedure. There is only one instance of BindMux MGR in the chassis.
		Handles multiplexing of the sessions across the available pccmgrs along with the session binding functions
		Monitors load on pccmgrs.
		Distributes incoming IP-CAN connections across pccmgrs in the system.
		Performs session binding; binds IP-CAN/Gateway session with the AF-Session.
		Ensures all messaging for an IMSI across various interfaces is directed towards the selected pccmgr.
		Remains aware of all the active PCC services in the system.
		With session recovery (SR) enabled, this demux manager is usually established on one of the CPUs on the first active packet processing card.
pccmgr	Policy and Charging Control Bindmux Manager	pccmgr is part of a Session Manager instance.
		Handles all PCRF service sessions.
		Interfaces with PCC-Core while processing different events associated with individual subscriber sessions.
		Maintains subscriber information while applying business logic.
		Creates calline and corresponding APN session for each subscriber.

Task	Description	Function
sgtpcmgr	SGSN GPRS Tunneling Protocol Control message Manager	Created by the Session Controller for each VPN context in which an SGSN service is configured.
		Terminates Gn/Gp and GTP-U interfaces from peer GGSNs and SGSNs for SGSN Services.
		Terminates GTP-U interfaces from RNCs for IuPS Services.
		Controls standard ports for GTP-C and GTP-U.
		Processes and distributes GTP-traffic received from peers on these ports.
		Performs all node level procedures associated with Gn/Gp interface.
		With session recovery (SR) enabled, this demux manager is usually established on one of the CPUs on the first active demux packet processing card.
srb	Standard Routing Database	Eight srbs are created by the Session Controller when Content Filtering in the Enhanced Charging Service is enabled. A minimum of two packet processing cards are required to initiate these eight tasks.
		Receives the static database from the session controller. Each srb task loads two database volumes (one primary and one secondary). The srb task also stores the static DB.
		Rates and categorizes the URL based on the DB volumes and CSI (Category Set Index) stored on it.
		Performs peer loading in case its peer fails. If both the srb task and its peer fail, the session controller performs the loading.

Platform Processes

Table 36: Platform Process Tasks

Task	Description	Function
afctrl	ASR 5500 Fabric Controller	Responsible for the overall management of the system fabric. Manages the pool of Rendezvous Destinations and coordinates fabric recovery by the afmgr proclets after a fault. A single afctrl instance runs on the active MIO/UMIO only.
afmgr	ASR 5500 Fabric Manager	Responsible for the management of fabric resources on a particular card. There is one afmgr on every CPU that is responsible for one or more fabric access processors (FAPs) or fabric elements (FEs). afmgr supports recovery but not migration.
afio	ASR 5500 Fabric I/O Driver	Responsible for the direct configuration of the fabric chipset. afio supports non-messenger interprocess communication (IPC) with the local afmgr and with other local afio instances
connproxy	TCP/SCTP Connection proxy	Allows applications on any card to share the same TCP/SCTP connection to the same remote endpoint instead of opening a new connection for each application on the card.
cspctrl	Card-Slot-Port Controller	Manages physical chassis components.
cssctrl	Content Server Selection (CSS) Controller	Maintains all global CSS properties which include a list of CSS servers that can be bound to a service in a context.
		CSS defines how traffic will be handled based on the "content" of the data presented by or sent to a mobile subscriber. CSS encompasses features such as load balancing, NAT, HTTP redirection, DNS redirection. The content server (services) can be either external to the platform or integrated within the platform. External CSS servers are configured via the Context Configuration mode **css server** command.
		The CSS Controller does not create CSS Managers. CSS Managers are stopped and started by VPN Managers. A CSS Manager is automatically created for each context.

Task	Description	Function
cssmgr	Content Server Selection (CSS) Manager	Spawned by the VPN Manager within a StarOS context.
		Manages the keepalives to a CSS server within the specific VPN context.
		Fetches the CSS related information for a subscriber
		If a CSS server goes down, the cssmgr task reprograms the NPUs to by-pass the service or redistribute the data among the rest of the servers in the service.
dcardctrl	Daughter Card Controller [ASR 5500 only]	Spawns daughter card managers during system initialization and monitors daughter card managers during system steady state execution. It also spawns daughter card managers whenever a daughter card manager task fails.
dcardmgr	Daughter Card Manager [ASR 5500 only]	Responsible for managing IPSec Security Associations for AH- and ESP-based sessions.
		Interfaces with the on-board hardware accelerated cryptographic chip which executes cryptographic algorithms associated with the given IPSec Security Associations.
dhmgr	Distributed Host Manager	Started automatically on each CPU by SITPARENT.
		Coordinates establishment of locally terminated TCP, SCTP, and UDP connections on behalf of multi-instanced tasks such as Diameter endpoints among sessmgr tasks.
drvctrl	Driver Controller	Centralizes access to many of the system device drivers. It also performs temperature and voltage monitoring.
hdctrl	Hard Drive Controller	Controls and manages the drive array spanning the management cards.
hwctrl	Hardware Controller	The hwctrl task has several timers that manage polling loops for hardware sensor readings, sensor threshold monitoring, and fan tray monitoring.

Task	Description	Function
hwmgr	Hardware Manager	The hwmgr task runs on all cards in the chassis to read local accessible hardware sensors and report them back to the hwctrl.
inetd	InterNET Service Daemon	The subsystem responsible for starting most of the network services.
		Listens for requests from connecting clients, such as FTP, SFTP, and telnet. When a TCP packet or UDP packet arrives with a particular destination port number, inetd launches the appropriate server program to handle the connection. **Note:** In release 20.0 and higher Trusted StarOS builds, FTP and Telnet are not supported.
ipsecctrl	IPSec Controller	Started by SIT on system startup regardless of configuration.
		Starts ipsecmgr tasks based on configuration and maintains its list for task recovery.
		Receives and maintains user configuration for IPSec.
		Manages the configured IPSec crypto maps and its assignment to ipsecmgrs.
		Interfaces with the vpnmgr task for required IPSec configuration parameters such as IP Access Lists, IP pools, interface addresses, and interface state notifications.
ipsecmgr	IPSec Manager	Created by the Session Controller, establishes and manages secure IKEv1, IKEv2 and IPSec data tunnels.
kvctrl	Key Value Controller	Central key value store (kvstore) function that runs on the management card. Its primary function is to support recovery and distribution functions.
lagmgr	Link Aggregation Group Manager [ASR 5500 only]	Started by npuctrl on the demux card's primary MIO (ASR 5500) with a facility level between CSP and npumgr to receive configuration/status notification from npumgr and build global LAG database.
		Exchanges control packets (LACP and Marker) over configured physical ports with peers to reach agreement on an aggregation of links.

Task	Description	Function
msgd	Messenger Daemon	Implements the Name Service and related functions for the internal message passing system.
msgproxy	Message Proxy	The Messenger Proxy process handles broadcast messages send from any single application (referred to as a client) to any facility which has one instance per thread (referred to as the Target Facility).
		One msgproxy task runs on each CPU complex on the DPCs (ASR 5500) and SF Virtual Machine (VPC-DI).
		Processes incoming broadcast messages from the Client processes, such as sessctrl, distributes them to the correct Target Facility, such as sessmgr, creates the correct responses and sends them back to the correct Client.
nscontrol	Name Service Controller	As part of the Messenger process, provides a reliable channel for tasks to send control messages to the Messenger Daemon.
ntpd	Network Time Protocol (NTP) Daemon	Maintains the system time in synchronization with time servers using NTP. Enabled when one or more NTP servers have been configured via the NTP Configuration mode **ntp server** CLI command.
rct	Recovery Control Task	Monitors tasks/managers/facilities across the system and performs recovery in the event of a failure.
sct	Shared Configuration Task	Performs the redundant storage of configuration information and other state information in an in-memory database.
sft	Switch Fabric Task	Monitors the switch fabric and the gigabit Ethernet control plane.
sshd	Secure SHell Daemon	Supports secure login to the StarOS CLI. Enabled via the Context Configuration mode **server sshd** CLI command.

Task	Description	Function
ucm	Utilities Configuration Manager	DHCPD, DNS, FTPD, INETD, NTPD, PING, RLOGIN, SFTPD, SFTP-SERVER, SNMPD, SSH, SSHD, TELNET, TELNETD, TFTPD, TRACEROUTE **Note:** In release 20.0 and higher Trusted StarOS builds, FTP and Telnet are not supported.

Management Processes

Table 37: Management Process Tasks

Task	Description	Function
bulkstat	Bulk Statistics Manager	Periodically polls and gathers bulk statistics and transfers this data to external management systems.
evlogd	Event Log Daemon	Handles event logging functions including the interface to external syslogd servers and the internal event logs.
orbs	ORBEM Service [ASR 5500 only]	The orbs task is also known as the ORB Element Manager (ORBEM). An Element Management System (EMS) requests orbs to perform Element Management Functions on the system using secure IIOP. ORBS then interacts with concerned Controller Tasks to execute the function. The response/errors from the execution are interpreted, formulated into an EMF response, and handed off to EMS servers.
orbns	ORBEM Notification Service [ASR 5500 only]	Notifies the EMS servers of event occurrences.
		Registers such EMS servers and subscribes them to associated event types.
		As the events occur, the concerned Controller Task notifies orbs (ORBEM), which then notifies the subscribing EMS servers.

Task	Description	Function
sesstrc	Session Trace Collection Task	Implements the standards-based session trace functionality.
		Manages both CLI and signaling-based subscriber traces. It collects messages to be traced and generates trace files as needed. It uploads trace files to the Trace Collection Entity as needed.
snmp	Simple Network Management Protocol	Handles inboard SNMP operations if configured, and sends SNMP notifications (traps) if enabled.
threshold	Threshold Server	Handles monitoring of threshold crossing alerts, if configured. Polls the needed statistics/variables, maintains state, and generates log messages/SNMP notification of threshold crossings.

APPENDIX C

NETCONF and ConfD

This chapter describes NETCONF and the StarOS process called ConfD manager.

It contains the following sections:

Overview

StarOS provides a northbound NETCONF interface that supports a YANG data model for transferring configuration and operational data with the Cisco Network Service Orchestrator (NSO). It also incorporates a ConfD engine to communicate with the NSO management console.

NETCONF (Network Configuration Protocol) is a network management protocol developed and standardized by the IETF (RFC 6241). It provides mechanisms to install, manipulate, and delete the configuration of network devices. Its operations are realized on top of a simple remote procedure call (RPC) layer. The NETCONF protocol uses XML-based data encoding for the configuration data as well as the protocol messages. The protocol messages are exchanged on top of a secure transport protocol.

ConfD is an on-device management framework that provides a set of interfaces to manage a device. The ConfD framework automatically renders all the management interfaces from a data-model. ConfD implements the full NETCONF specification and runs over SSH with content encoded in XML.

ConfD is configured to allow only authenticated/authorized access through external authentication. ConfD Manager provides a standalone CLI module for ConfD to invoke when authenticating/authorizing any new users. ConfD is configured to allow only authorized access through StarOS authentication. Upon authentication, the user is given a privilege level (0-15) which is mapped to StarOS *secure admin*, *admin*, *operator*, and *inspector*, as defined in the YANG model. StarOS logs CLI authentication event/status messages for each ConfD authentication request.

On the southbound side, ConfD communicates with a StarOS process called via a set of APIs provided by the ConfD management agent. The ConfD Configuration Database (CDB) is used by ConfD to store objects. StarOS accesses the database through the ConfD-supplied APIs. Once the ConfD configuration database is

populated, StarOS continues to allow CLI access to modify the overall configuration. There are no automatic updates to the CDB as a result. The CDB only receives updates via the NETCONF interface. In order to keep the CDB and the StarOS configuration databases in sync, all changes made via CLI access (external to NETCONF) to YANG model supported configuration objects must be applied to the CDB manually.

YANG is a data modeling language for the NETCONF network configuration protocol. It can be used to model both configuration data as well as state data of network elements. YANG can also be used to define the format of event notifications emitted by network elements and it allows data modelers to define the signature of remote procedure calls that can be invoked on network elements via the NETCONF protocol (RFC 6020). The YANG file is compiled as part of StarOS and incorporates a subset of the existing StarOS supported CLI commands.

In this release, the YANG model supports a limited set of ECS configuration commands via NSO. For additional information, refer to NSO user documentation and Supported StarOS ECS Configuration Commands, on page 487.

Configuring ConfD

To enable NETCONF protocol in StarOS, you must enable **server confd** and enter the NETCONF Protocol Configuration mode. The NETCONF Protocol Configuration mode supports optional configuration commands.

SSH Key Requirement

NETCONF-ConfD support requires that a V2-RSA SSH key be configured on the local context.

If an SSH key is not available, StarOS generates an error message.

```
Failure: The ConfD (NETCONF) server requires an RSA key on the local context
```

You can run the **show ssh key** command to verify the existence of an SSH key on the system.

If an SSH key is not available, see the *Configuring SSH Options* section of the *Getting Started* chapter in this guide.

NETCONF Protocol Configuration Mode

The NETCONF protocol is enabled via the Context Configuration mode **server conf** command. This command is restricted to the local context only.

```
[local]host_name# configure

[local]host_name(config)# context local

[local]host_name(config-ctx)# server confd

[local]host_name(config-confd)# ?
  autosave-config      - Automatically saves current configuration to the URL specified
                         whenever a change is applied through ConfD interfaces

  confd-user           - Configures the default login user with full administrator rights
                         for the ConfD server.

  end                  - Exits configuration mode and returns to Exec Mode

  exit                 - Exits current configuration mode, returning to previous mode

  no                   - Enables/Disables the followed option
```

The **autosave-config** and **confd-user** keywords are optional.

To disable NETCONF protocol, run the Global Configuration mode **no server confd** command.

For additional information, see the *NETCONF Protocol Configuration Mode Commands* chapter of the *Command Line Interface Reference*.

autosave-config

This NETCONF Protocol Configuration mode command automatically saves the current ConfD configuration to a specified URL whenever a change is applied by NSO through the ConfD interface in the CLI based model. By default this command is disabled.

The command syntax is: **autosave-config** <*url*>, where **[file:]{/flash | /usb1 | /hd-raid | /sftp} [** /<*directory*>**]** /<*filename*>.

confd-user

This NETCONF Protocol Configuration mode command associates a username for all CLI operations via NETCONF. The user will be authenticated with verifiable credentials. This username is used for CLI logging purposes only.

The command syntax is: **confd-user** <*username*>, where <*username*> is an alphanumeric string of 1 to 144 characters.

Important The NETCONF or RESTful session must still be established with verifiable credentials.

Seeding and Synchronizing the CDB

After enabling **server confd** you may need to initially seed the CDB with a local copy of the configuration database (CDB) managed by ConfD on StarOS. The seeding procedure creates a CDB used by ConfD on the StarOS platform that contains all CLI based YANG model supported configuration commands.

Important
- If you manually modify a managed object via the StarOS CLI, you must resynchronize the running configuration with the NSO by repeating the procedure described below.

Step 1 Run Exec mode **save configuration** <*url*> **confd** to save the ConfD supported StarOS configuration data to a file on the /flash device.

Step 2 Run Exec mode **show configuration error** to validate the saved configuration. Correct any errors before applying the configuration. Otherwise, ConfD will reject the entire configuration.

Step 3 Run Exec mode **configure confd** <*url*> to apply the ConfD configuration. Once the ConfD configuration is applied, the device is ready to establish NETCONF connections to the NSO management service.

Step 4 Synchronize the device with your NSO. Refer to NSO user documentation for detailed information on the synchronization process.

Sample Configuration

The following command sequence establishes a COnfD configuration in support of NETCONF protocol.

A type v2-RSA SSH key is required for enabling **server confd**.

```
configure
  context local
      ssh key
<encrypted key text>
len 938 type v2-rsa
    server confd
      confd-user NETCONF
      autosave-config /flash/config.cfg
    #exit
    subscriber default
    exit
    aaa group default
    #exit
    gtpp group default
    #exit
  #exit
end
```

Notes:

- **confd-user** and **autosave-config** are optional. Just configuring **server confd** enables NETCONF support.

Verifying the Configuration

There are two Exec mode **show** commands that display information about the NETCONF-ConfD configuration.

show confdmgr Command

This command displays information about the StarOS ConfD Manager (confdmgr) process.

The syntax for this command is:

show confdmgr { confd { cdb | netconf | state } | subscriptions }

Notes:

See below for a sample output for **show confdmgr**:

- The **confd** keyword displays the following options:
  - **cdb** displays ConfD CDB information
  - **netconf** displays NETCONF state information
  - **state** displays current ConfD state information
- The **subscriptions** keyword displays ConfD CDB subscription information.

```
[local]<host_name># show confdmgr

State Information
-----------------
State               Started
```

```
Subscriptions       2
Last successful id  1461-704882-705350
Last failed id      None
Autosave url        Not configured
Username            Not configured

Statistics
----------
Triggers                  1
Notifications             2
Successful notifications  2
Failed notifications      0
Unexpected                0
[local]<host_name>#
```

The Statistics portion of this output includes the following information:

- **Triggers** – Number of times confdmgr has requested ConfD to dump the CDB contents back into confdmgr which results in a config synchronization by SCT (Shared configuration Task).
- **Notifications** – Number of times ConfD has sent an update to confdmgr.
- **Successful Notifications** – Number of times an update received from ConfD was successfully processed.
- **Failed Notifications** – Number of times an update received from ConfD was not processed successfully. The number of successes and failures should always equal the total number of notifications.
- **Unexpected** – Number of times an unexpected condition was encountered. An error log is generated for each case.

See below for a sample output for **show confdmgr confd cbd**:

```
[local]<host_name># show confdmgr confd cdb
active-charging service acs
exit
context local
 server confd
  autosave-config /tmp/ut_confdmgr_config.txt
  confd-user      confd_user
 exit
exit
nacm read-default permit
nacm groups group admin
!
nacm groups group inspector
!
nacm groups group operator
!
nacm groups group secure_admin
!
nacm rule-list secure_admin
!
 group [ secure_admin ]
 rule any-access
  action permit
!
 rule secure_admin_server_confd
  module-name       cisco-staros-cli-config
  path              /context/server/confd
  access-operations create,read,update
  action            permit
  |
  |
  V
```

```
nacm rule-list inspector
 group [ inspector ]
 rule any-access
  access-operations read
  action            permit
 !
!
[local]<host_name>#
```

See below for a sample output for **show confdmgr confd netconf**:

```
[local]<host_name># show confdmgr confd netconf
netconf-state capabilities capability urn:ietf:params:netconf:base:1.0
netconf-state capabilities capability urn:ietf:params:netconf:base:1.1
netconf-state capabilities capability urn:ietf:params:netconf:capability:writable-running:1.0
netconf-state capabilities capability urn:ietf:params:netconf:capability:candidate:1.0
 |
 |
 V
netconf-state statistics netconf-start-time 2016-03-30T17:09:49-04:00
netconf-state statistics in-bad-hellos 0
netconf-state statistics in-sessions 0
netconf-state statistics dropped-sessions 0
netconf-state statistics in-rpcs 0
 |
 |
 V
netconf-state datastores datastore candidate
NAME         CREATOR  CREATED                    CONTEXT
-----------------------------------------------------------
/rollback0   admin    2016-04-26T17:08:02-00:00  noaaa
/rollback1   admin    2016-04-26T17:07:57-00:00  noaaa
/rollback2   admin    2016-04-26T17:07:49-00:00  noaaa
/rollback3   admin    2016-04-26T17:07:49-00:00  noaaa
/rollback4   admin    2016-04-26T17:07:47-00:00  noaaa
 |
 |
 V
/cli-history/admin.hist
/cli-history/root.hist
/global.data

[local]<host_name>#
```

See below for a sample output for **show confdmgr confd state**:

```
[local]<host_name># show confdmgr confd state
Monday June 24 10:58:49 EDT 2019
confd-state version 7.1
confd-state epoll false
confd-state daemon-status started
confd-state loaded-data-models data-model acs-config
 revision        2016-10-31
 namespace       http://www.cisco.com/usp/nfv/acs-config
 prefix          acs-config
 exported-to-all
confd-state loaded-data-models data-model cisco-staros-bulkstats
 revision        2016-12-14
 namespace       http://www.cisco.com/staros-bulkstats
 prefix          staros_bulkstats
 exported-to-all
confd-state loaded-data-models data-model cisco-staros-cli-config
 revision        2016-12-14
 namespace       http://www.cisco.com/staros-cli-config
 prefix          staros_cli
```

```
 exported-to-all
confd-state loaded-data-models data-model cisco-staros-config
 revision        2016-12-14
 namespace       http://www.cisco.com/staros-config
 prefix          staros_config
 exported-to-all
confd-state loaded-data-models data-model cisco-staros-exec
 revision        2016-12-14
 namespace       http://www.cisco.com/staros-exec
 prefix          staros_exec
 exported-to-all
confd-state loaded-data-models data-model cisco-staros-kpi
 revision        2017-10-31
 namespace       http://www.cisco.com/staros-kpi
```

See below for a sample output for **show confdmgr subscriptions**:

```
[local]<host_name># show confdmgr subscriptions

Subscriptions:
Path                                Index  Namespace
--------------------------------------------------
/active-charging                        6     staros
/context                                7     staros
[local<host_name>#
```

Subscriptions are configuration points defined in the Yang model for which confdmgr wants to be notified when a change occurs. In this release there are two subscriptions: “/active-charging” and “/context”.

show configuration confd Command

The **confd** keyword filters the output of the **show configuration** command to display only configuration commands that are supported by the YANG model.

```
show configuration confd
```

A sample output appears below.

```
[local]<host_name># show configuration confd
config
  context local
    server confd
    #exit
  active-charging service ecs
    ruledef rd1
      tcp any-match = TRUE
    #exit
    rulebase default
    #exit
  #exit
end
[local]<host_name>#
```

clear confdmgr statistics

This command clears everything listed in the "Statistics" section of the output of the **show confdmgr** command, including:

- Triggers
- Notifications
- Successful notifications
- Failed notifications
- Unexpected

Show Support Details (SSD)

The output of all **show confdmgr** commands has been added to the SSD.

CDB Maintenance

A local copy of the ConfD Configuration Database (CDB) is managed by ConfD on StarOS.

You can show and save all ConfD supported StarOS configuration commands to a URL. The **confd** keyword has been added to the **show configuration** and **save configuration** commands for these purposes.

After saving a ConfD-supported configuration to a URL, you can apply it directly to the CDB via the Exec mode **configure confd** <*url*> command. This command applies the contents of the file at the *url* to the running configuration of ConfD.

Additional detail regarding the above commands is provided below.

clear confdmgr confd cdb

This Exec mode command erases the configuration in the ConfD Configuration Database (CDB) which is used by ConfD to store configuration objects. StarOS accesses the database via ConfD-supplied APIs.

Note The CDB cannot be erased unless the Context Configuration mode **no server confd** command is run in the local context to disable ConfD and NETCONF protocol support.

The following is a sample command sequence for clearing the CDB:

```
[local]host_name# config
[local]host_name(config)# context local
[local]host_name(config-ctx)# no server confd
[localhost_name(config-ctx)# end
[local]host_name# clear confdmgr confd cdb
About to delete the ConfD configuration database
The running configuration is NOT affected.
```

```
Are you sure? [Yes|No]: y
[local]host_name#
```

Caution Clearing the CDB is a terminal operation. The CDB must be repopulated afterwards.

configure confd <url>

This Exec mode command applies the contents of the configuration script specified by the URL to the current ConfD configuration database (CDB).

A sample command sequence is provided below.

```
[local]host_name# save configuration /flash/confd.config confd
[local]host_name# configure confd /flash/confd.config
Info: #!$$ StarOS V20.2 Chassis 52767e9ff9e207bed12c76f7f8a5352c
Info: config
Info:   active-charging service acs
Info:     rulebase default
Info:     #exit
Info:   #exit
Info: end
[local]host_name#
```

save configuration <url> confd

The keyword **confd** is added to the Exec mode **save configuration** command. This keyword filters the saved configuration commands to contain only configuration commands that are supported by the YANG model.

The command syntax for this process is:

```
[local]host_name# save configuration <url> confd
```

The output of the YANG model subset of configuration commands can be viewed via the **show file url** <*url*> command, where <*url*> is the pathname used to save the configuration. The saved configuration file can then be applied to the CDB using the **configure confd** command.

Supported StarOS ECS Configuration Commands

For this release, the following StarOS ECS commands are supported for the CLI based YANG model:

- ruledef <*ruledef_name*>
  - ip server-ip-address = *
  - tcp-ether-port = *
  - udp ether-port = *
  - tcp either-port-range = *
  - udp ether-port range = *
  - tcp-any-match = *

    - udp any-match = *
    - http url = *
    - httpcookie = *
    - http x-header = *
- group-of-ruledefs <*ruledefs_group_name*>
    - add-ruledef priority = *
- qos-group-of-ruledefs <*group_name*>
    - add-group-of-ruledef <*group_of_ruledef_name*>
- charging-action <*charging_action_name*>
    - flow-idle-timeout <*seconds*>
    - content-id 1
    - service-identifier <*service_id*>
    - billing-action egcdr
- rulebase <*rulebase_name*>
    - action priority <*priority_number*> group-of-ruledefs <*ruledefs_group_name*> charging-action <*charging_action_name*>

Note "= *" indicates support for every option following the prior keyword/value.

ConfD Upgrade Support

APPENDIX D

ICSR Checkpointing

This appendix lists and describes macro- and micro-checkpoints employed by the Interchassis Session Recovery framework. Checkpoints are exchanged between the active and standby ICSR chassis via the Service Redundancy Protocol (SRP).

The following topics are discussed:

Overview of Checkpointing

Interchassis Session Recovery (ICSR) provides a framework for sessmgr instance-level checkpointing within an ICSR framework. A checkpoint is a snapshot of the status of an application. Checkpointing can be used by sessmgr to push instance level information to the peer chassis.

Instance-level checkpointing sends messages to specific sessmgr instances. Each application, such as GGSN, PDSN, P-GW, S-GW or SGSN, is responsibility for encoding and decoding the checkpoint message. The ICSR framework provides the APIs for transport of the instance-level checkpoint information and associated statistics.

Macro-checkpoints contain full session information and micro-checkpoints contain only a few variables. Macro-checkpoints are sent initially from the active chassis to the standby chassis on power up and reload, and periodically thereafter. When a standby chassis receives macro-checkpoints, it clears any existing CRR (Call Recovery Record) or CLP (Call Line Pointer) related to that session, and creates a new CRR or CLP. Macro-checkpoints are also known as full checkpoints (FCs).

To conserve processing cycles and memory, dynamic and periodic updates from an active chassis to a standby chassis are done using micro-checkpoints.

The output of the Exec mode **show srp info** command displays a complete list of SRP checkpoints.

Macro-checkpoints

This section lists and briefly describes ICSR macro-checkpoints.

GGSN_APN ID MAPPING

This macro-checkpoint is sent from the active to the standby chassis to map APN names on the standby chassis.

- **Time based:** No
- **Frequency:** N/A
- **Event based:** Yes
- **Events:** Occurs whenever a TCP connection is established between the sessmgrs and they move to READY_STATE.
- **Accounting:** No
- **Delta/Cumulative:** N/A
- **Related CLI command: show session subsystem facility sessmgr instance** <*instance no*> **debug-info** and **show srp micro-checkpoint statistics**

INSTANCE LEVEL CHECKPOINT

This macro-checkpoint is generated by ECS (Enhanced Charging System) to send new rules to the standby chassis. It is also used by ECS to delete or modify a rule on the standby chassis.

- **Time based:** Yes
- **Frequency:** 30 minutes
- **Event based:** Yes
- **Events:** Occurs:
  1. When a new rule is added or deleted on the active chassis.
  2. Every 30 minutes if the ECS is registered for periodic micro-checkpointing.
- **Accounting:** —
- **Delta/Cumulative:** —
- **Related CLI command: show session subsystem facility sessmgr instance** <*instance no*> **debug-info** and **show srp micro-checkpoint statistics**

SERVICE_ID MAPPING

This macro-checkpoint is sent from the active to the standby chassis to map Service IDs on the standby chassis.

- **Time based:** No
- **Frequency:** N/A
- **Event based:** Yes
- **Events:** Occurs whenever a TCP connection is established between the sessmgrs and they move to READY_STATE.
- **Accounting:** No
- **Delta/Cumulative:** N/A
- **Related CLI command: show session subsystem facility sessmgr instance** <*instance no*> **debug-info**

VPNMGR_ID MAPPING

This macro-checkpoint is sent from the active to the standby chassis to map VPNs on the standby chassis.

- **Time based:** No

- **Frequency:** N/A
- **Event based:** Yes
- **Events:** Occurs whenever a TCP connection is established between the sessmgrs and they move to READY_STATE.
- **Accounting:** No
- **Delta/Cumulative:** N/A
- **Related CLI command: show session subsystem facility sessmgr instance** <*instance no*> **debug-info**

Micro-checkpoints

This section lists and briefly describes the characteristics of micro-checkpoints by application category.

Micro-checkpoints are listed in alphabetical order under the following categories:

Uncategorized

SESS_UCHKPT_CMD_INVALIDATE_CRR

This micro-checkpoint is sent to the standby chassis to clear a deleted call. It carries the Call ID and other information that must be deleted on the standby chassis.

- **Time based:** No
- **Frequency:** N/A
- **Event based:** Yes
- **Events:** Occurs when a call is deleted on the active chassis.
- **Accounting:** No
- **Delta/Cumulative:** N/A
- **CMD-ID:** 1
- **Related CLI command:** None

SESS_UCKKPT_CMD_UPDATE_CLPSTATS

This micro-checkpoint sends VoLTE data statistics.

- **Time based:** Yes

- **Frequency:** —
- **Event based:** Yes
- **Events:** Occurs during ICSR background checkpointing. A chassis switchover triggers the sending of VoLTE data stats.
- **Accounting:** —
- **Delta/Cumulative:** —
- **CMD-ID:** 4
- **Related CLI command:** None

SESS_UCHKPT_CMD_UPDATE_IDLESECS

This micro-checkpoint sends remaining number of seconds before idle timeout.

- **Time based:** Yes
- **Frequency:** —
- **Event based:** No
- **Events:** Occurs during ICSR background checkpointing.
- **Accounting:** No
- **Delta/Cumulative:** N/A
- **CMD-ID:** 2
- **Related CLI command:** None

DCCA Category

SESS_UCHKPT_CMD_DCCA_SESS_INFO

This micro-checkpoint sends Credit Control (CC) related information.

- **Time based:** Yes
- **Frequency:** 18 seconds for GR micro-checkpoint
- **Event based:** Yes
- **Events:** Sent along with the macro-checkpoint/CCA/Assume-positive-state-transitions.
- **Accounting:** Yes
- **Delta/Cumulative:** Cumulative
- **CMD-ID:** 19
- **Related CLI command:** None

ECS Category

SESS_UCHKPT_CMD_ACS_CALL_INFO

This micro-checkpoint sends critical ECS call level data.

- **Time based:** Yes
- **Frequency:** —
- **Event based:** Yes
- **Events:** Occurs whenever ECS call level information is created or modified.
- **Accounting:** No

- **Delta/Cumulative:** N/A
- **CMD-ID:** 179
- **Related CLI command:** None

SESS_UCHKPT_CMD_ACS_GX_LI_INFO

This micro-checkpoint sources lawful intercept (LI) related information maintained by ECS.

- **Time based:** Yes
- **Frequency:** —
- **Event based:** Yes
- **Events:** Occurs whenever LI information is created or modified.
- **Accounting:** No
- **Delta/Cumulative:** N/A
- **CMD-ID:** 75
- **Related CLI command:** None

SESS_UCHKPT_CMD_ACS_SESS_INFO

This micro-checkpoint sends ECS-level bearer-related data

- **Time based:** Yes
- **Frequency:** —
- **Event based:** Yes
- **Events:** Occurs whenever ECS bearer information is created or modified.
- **Accounting:** No
- **Delta/Cumulative:** N/A
- **CMD-ID:** 33
- **Related CLI command:** None

SESS_UCHKPT_CMD_DEL_ACS_CALL_INFO

This micro-checkpoint notifies that a Release Call event has occurred.

- **Time based:** No
- **Frequency:** N/A
- **Event based:** Yes
- **Events:** Occurs whenever an ECS Release Call message is processed.
- **Accounting:** No
- **Delta/Cumulative:** N/A
- **CMD-ID:** 188
- **Related CLI command:** —

SESS_UCHKPT_CMD_DEL_ACS_SESS_INFO

This micro-checkpoint notifies that a Release Bearer event has occurred.

- **Time based:** No
- **Frequency:** N/A
- **Event based:** Yes
- **Events:** Occurs whenever an ECS Release Bearer message is processed.

- **Accounting:** No
- **Delta/Cumulative:** N/A
- **CMD-ID:** 187
- **Related CLI command:** None

SESS_UCHKPT_CMD_DYNAMIC_CHRG_CA_INFO

This micro-checkpoint sends dynamic charging action information maintained by ECS.

- **Time based:** Yes
- **Frequency:** —
- **Event based:** Yes
- **Events:** Occurs whenever dynamic charging action information is created or modified.
- **Accounting:** No
- **Delta/Cumulative:** N/A
- **CMD-ID:** 141
- **Related CLI command:** None

SESS_UCHKPT_CMD_DYNAMIC_CHRG_DEL_CA_INFO

This micro-checkpoint notifies that a dynamic charging action has been deleted.

- **Time based:** No
- **Frequency:** N/A
- **Event based:** Yes
- **Events:** Occurs whenever a dynamic charging action has been deleted.
- **Accounting:** No
- **Delta/Cumulative:** N/A
- **CMD-ID:** 183
- **Related CLI command:** None

SESS_UCHKPT_CMD_DYNAMIC_CHRG_DEL_QG_INFO

This micro-checkpoint notifies that a dynamic QoS group has been deleted.

- **Time based:** No
- **Frequency:** N/A
- **Event based:** Yes
- **Events:** Occurs whenever a dynamic QoS group has been deleted.
- **Accounting:** No
- **Delta/Cumulative:** N/A
- **CMD-ID:** 182
- **Related CLI command:** None

SESS_UCHKPT_CMD_DYNAMIC_CHRG_QG_INFO

This micro-checkpoint sends dynamic QoS group related information maintained by ECS.

- **Time based:** Yes
- **Frequency:** —
- **Event based:** Yes

- **Events:** Occurs whenever dynamic QoS group information is created or modified.
- **Accounting:** No
- **Delta/Cumulative:** N/A
- **CMD-ID:** 140
- **Related CLI command:** None

SESS_UCHKPT_CMD_DYNAMIC_RULE_DEL_INFO

This micro-checkpoint notifies that a dynamic rule has been deleted.

- **Time based:** No
- **Frequency:** —
- **Event based:** Yes
- **Events:** Occurs whenever a dynamic rule has been deleted.
- **Accounting:** No
- **Delta/Cumulative:** N/A
- **CMD-ID:** 178
- **Related CLI command:** None

SESS_UCHKPT_CMD_DYNAMIC_RULE_INFO

This micro-checkpoint sources predefined and dynamic rule related information maintained by ECS.

- **Time based:** Yes
- **Frequency:** —
- **Event based:** Yes
- **Events:** Occurs whenever a dynamic rule is created or modified.
- **Accounting:** No
- **Delta/Cumulative:** N/A
- **CMD-ID:** 43
- **Related CLI command:** None

ePDG Category

SESS_UCHKPT_CMD_DELETE_EPDG_BEARER

This micro-checkpoint synchronizes deleted ePDG bearers between the active and standby chassis.

- **Time based:** No
- **Frequency:** N/A
- **Event based:** Yes
- **Events:** N/A
- **Accounting:** Yes
- **Delta/Cumulative:** Cumulative
- **CMD-ID:** 110
- **Related CLI command: show srp micro-checkpoint statistics debug-info**

SESS_UCHKPT_CMD_UPDATE_EPDG_BEARER

This micro-checkpoint synchronizes ePDG bearers between the active and standby chassis.

- **Time based:** No
- **Frequency:** N/A
- **Event based:** No
- **Events:** N/A
- **Accounting:** Yes
- **Delta/Cumulative:** Cumulative
- **CMD-ID:** 110
- **Related CLI command: show srp micro-checkpoint statistics debug-info**

SESS_UCHKPT_CMD_UPDATE_EPDG_PEER_ADDR

This micro-checkpoint synchronizes ePDG peer addresses between the active and standby chassis.

- **Time based:** No
- **Frequency:** N/A
- **Event based:** Yes
- **Events:** —
- **Accounting:** Yes
- **Delta/Cumulative:** Cumulative
- **CMD-ID:** 110
- **Related CLI command: show srp micro-checkpoint statistics debug-info**

SESS_UCHKPT_CMD_UPDATE_EPDG_REKEY

This micro-checkpoint synchronizes ePDG rekey statistics between the active and standby chassis.

- **Time based:** Yes
- **Frequency:** 30 seconds
- **Event based:** No
- **Events:** N/A
- **Accounting:** Yes
- **Delta/Cumulative:** Cumulative
- **CMD-ID:** 110
- **Related CLI command: show srp micro-checkpoint statistics debug-info**

SESS_UCHKPT_CMD_UPDATE_EPDG_STATS

This micro-checkpoint synchronizes session statistics between the active and standby chassis.

- **Time based:** Yes
- **Frequency:** 30 seconds
- **Event based:** No
- **Events:** N/A
- **Accounting:** Yes
- **Delta/Cumulative:** Cumulative
- **CMD-ID:** 110

- **Related CLI command: show srp micro-checkpoint statistics debug-info**

Firewall/ECS Category

SESS_UCHKPT_CMD_SFW_DEL_RULE_INFO

This micro-checkpoint is sent when a ruledef is deleted for a bearer.

- **Time based:** No
- **Frequency:** N/A
- **Event based:** Yes
- **Events:** Occurs whenever PCRF sends a command to disable the predefined stateful firewall access rules.
- **Accounting:** No
- **Delta/Cumulative:** N/A
- **CMD-ID:** 186
- **Related CLI command:** None

SESS_UCHKPT_CMD_SFW_RULE_INFO

This micro-checkpoint notifies the addition of dynamically enabled stateful firewall (SFW) access rules.

- **Time based:** No
- **Frequency:** N/A
- **Event based:** Yes
- **Events:** Occurs whenever PCRF sends a command to enable the predefined SFW access rules.
- **Accounting:** Yes
- **Delta/Cumulative:** Cumulative
- **CMD-ID:** 185
- **Related CLI command:** None

GGSN Category

SESS_UCHKPT_CMD_GGSN_DELETE_SUB_SESS

This micro-checkpoint sends an update when a secondary bearer is deleted.

- **Time based:** No
- **Frequency:** N/A
- **Event based:** Yes
- **Events:** Occurs upon secondary bearer deletion
- **Accounting:** —
- **Delta/Cumulative:** —
- **CMD-ID:** 117
- **Related CLI command:** None

SESS_UCHKPT_CMD_GGSN_UPDATE_RPR

If RPR (Resilient Packet Ring) is configured in the GGSN service, an RPR timer is started during secondary bearer creation. This checkpoint is sent upon expiry of this timer.

- **Time based:** Yes
- **Frequency:** RPR timer
- **Event based:** Yes
- **Events:** Occurs when the secondary bearer creation RPR timer expires.
- **Accounting:** —
- **Delta/Cumulative:** —
- **CMD-ID:** 118
- **Related CLI command:** —

SESS_UCHKPT_CMD_GGSN_UPDATE_SESSION

This micro-checkpoint is sent in a Network or UE initiated update procedure except for updates that result in the following scenarios:

- Creation or deletion of the beare
- TFT change or inter-RAT handovers
- Gn-Gp handoff

Parameters associated with this micro-checkpoint are shown below.

- **Time based:** No
- **Frequency:** N/A
- **Event based:** Yes
- **Events:** Occurs for a network initiated or UE initiated update.
- **Accounting:** No
- **Delta/Cumulative:** N/A
- **CMD-ID:** 171
- **Related CLI command: show srp checkpoint statistics active verbose**, and **show session subsystem facility sessmgr instance** <*instance_number*> **debug-info**.

SESS_UCHKPT_CMD_GGSN_UPDATE_STATS

This micro-checkpoint periodically sends session statistics.

- **Time based:** Yes
- **Frequency:** Every five minutes
- **Event based:** No
- **Events:** N/A
- **Accounting:** Yes
- **Delta/Cumulative:** Cumulative
- **CMD-ID:** 116
- **Related CLI command:** None

SESS_UCHKPT_CMD_UPDATE_COA_PARAMS

This micro-checkpoint updates input and output ACL parameters.

- **Time based:** —
- **Frequency:** —
- **Event based:** Yes
- **Events:** COA (Change of Authorization) response
- **Accounting:** —
- **Delta/Cumulative:** —
- **CMD-ID:** 83
- **Related CLI command:** None

Gx Interface Category

SESS_UCHKPT_CMD_ACS_VOLUME_USAGE

This micro-checkpoint sends volume usage over Gx accounting buckets.

- **Time based:** Yes
- **Frequency:** 4 seconds for aamgr micro-checkpoint and 18 seconds for GR micro-checkpoint
- **Event based:** No
- **Events:** Send along with macro-checkpoint
- **Accounting:** Yes
- **Delta/Cumulative:** Cumulative
- **CMD-ID:** 79
- **Related CLI command:** —None

SESS_UCHKPT_CMD_UPDATE_SGX_INFO

This micro-checkpoint sends Gx session-related information.

- **Time based:** No
- **Frequency:** N/A
- **Event based:** Yes
- **Events:** Triggered on receiving CCA-I/U or RAR from PCRF.
- **Accounting:** Yes
- **Delta/Cumulative:** Cumulative
- **CMD-ID:** 137
- **Related CLI command:** None

NAT Category

SESS_UCHKPT_CMD_GR_UPDATE_NAT_REALM_PORT_INFO1

This micro-checkpoint is sent when a port chunk is allocated or deallocated for a subscriber sharing a NAT IP address with other subscribers. The port chunk is allocated or deallocated while data is being received for that subscriber.

- **Time based:** No
- **Frequency:** N/A
- **Event based:** Yes

- **Events:** Triggered when a new NAT port chunk is allocated or deleted.
- **Accounting:** No
- **Delta/Cumulative:** N/A
- **CMD-ID:** 105
- **Related CLI command:** None

SESS_UCHKPT_CMD_GR_UPDATE_NAT_REALMS

This micro-checkpoint is sent when a NAT IP address is allocated to or deallocated from a subscriber.

For an on-demand case, it is triggered when the first packet matching a particular NAT realm is received and the NAT IP address is allocated to the subscriber.

If this is not an on-demand case, the NAT IP address is allocated during call setup and this micro-checkpoint is sent.

- **Time based:** No
- **Frequency:** N/A
- **Event based:** Yes
- **Events:** Triggered when a NAT IP address is allocated to or deallocated from a subscriber.
- **Accounting:** No
- **Delta/Cumulative:** N/A
- **CMD-ID:** 45
- **Related CLI command:** None

SESS_UCHKPT_CMD_NAT_SIP_ALG_CALL_INFO

This micro-checkpoint is sent when a new SIP flow is created or deleted for a subscriber (while SIP data is being passed via the subscriber).

- **Time based:** No
- **Frequency:** N/A
- **Event based:** Yes
- **Events:** Triggered when a new SIP flow is created or deleted.
- **Accounting:** No
- **Delta/Cumulative:** N/A
- **CMD-ID:** 98
- **Related CLI command:** None

SESS_UCHKPT_CMD_NAT_SIP_ALG_CONTACT_PH_INFO

This micro-checkpoint is sent when a received SIP packet is analyzed and pinholes are created in the NAT firewall.

- **Time based:** No
- **Frequency:** N/A
- **Event based:** Yes
- **Events:** Triggered when a SIP packet creates pinholes in the NAT firewall.
- **Accounting:** No
- **Delta/Cumulative:** N/A
- **CMD-ID:** 97

- **Related CLI command:** None

SESS_UCHKPT_CMD_UPDATE_DSK_FLOW_CHKPT_INFO

This micro-checkpoint is sent when a new NAT flow is created or deleted for a subscriber (while data is being passed via the subscriber).

This checkpoint is sent from a timer but it is not timer based. The timer is used to pace (10 micro-checkpoints) whenever the timer fires (granularity = 2 sec). This only occurs if there are new flows that need to be micro-checkpointed. Otherwise, no micro-micro-checkpoints are sent.

- **Time based:** No
- **Frequency:** See explanation above.
- **Event based:** Yes
- **Events:** Triggered when a new NAT flow is created or deleted.
- **Accounting:** No
- **Delta/Cumulative:** N/A
- **CMD-ID:** 96
- **Related CLI command:** None

SESS_UCHKPT_CMD_UPDATE_NAT_BYPASS_FLOW_INFO

This micro-checkpoint is sent when NAT is enabled for a subscriber but bypass-nat (no NATing) is configured for this flow (based on a rule-match), and a new bypass flow is created.

This checkpoint is sent when the flow is both added and deleted.

- **Time based:** No
- **Frequency:** N/A
- **Event based:** Yes
- **Events:** Triggered when a new flow with bypass-nat enabled is created or deleted.
- **Accounting:** No
- **Delta/Cumulative:** N/A
- **CMD-ID:** 60
- **Related CLI command:** None

P-GW Category

SESS_UCHKPT_CMD_PGW_DELETE_SUB_SESS

Reserved for future use.

SESS_UCHKPT_CMD_PGW_OVRCHRG_PRTCTN_INFO

This micro-checkpoint indicates that the S-GW has set the Overcharging Protection bit in the MBR.

- **Time based:** No
- **Frequency:** N/A
- **Event based:** Yes
- **Events:** Triggered when the S-GW sets the Over Charging Protection Bit.
- **Accounting:** No

- **Delta/Cumulative:** N/A
- **CMD-ID:** 159
- **Related CLI command:** None

SESS_UCHKPT_CMD_PGW_SGWRESTORATION_INFO

This micro-checkpoint indicates the interval that a call will remain up when the S-GW is down.

- **Time based:** No
- **Frequency:** N/A
- **Event based:** Yes
- **Events:** Triggered when the S-GW goes into Restoration mode.
- **Accounting:** No
- **Delta/Cumulative:** N/A
- **CMD-ID:** 158
- **Related CLI command:** None

SESS_UCHKPT_CMD_PGW_UBR_MBR_INFO

This micro-checkpoint is sent at the end of a UBR (Update Bearer Request) or MBR (Modify Bearer Request) except when the UBR /MBR procedure results in the following scenarios:

- TFT change
- Bearer updat or modification for a collapsed call
- Pure P to collapsed or collapsed to Pure P change
- Inter-technology handoff, for example, WiFi to LTE

Parameters associated with this micro-checkpoint are show below.

- **Time based:** No
- **Frequency:** N/A
- **Event based:** yes
- **Events:** Occurs as a result of a UBR or MBR procedure.
- **Accounting:** No
- **Delta/Cumulative:** N/A
- **CMD-ID:** 193
- **Related CLI command: show srp checkpoint statistics active verbose** and **show session subsystem facility sessmgr instance** <*instance_number*> **debug-info**.

SESS_UCHKPT_CMD_PGW_UPDATE_APN_AMBR

Reserved for future use.

SESS_UCHKPT_CMD_PGW_UPDATE_INFO

Reserved for future use.

SESS_UCHKPT_CMD_PGW_UPDATE_LI_PARAM

This micro-checkpoint indicates the state of Lawful Intercept (LI) for this call.

- **Time based:** No

- **Frequency:** N/A
- **Event based:** Yes
- **Events:** Triggered when there is a change in the LI state for this call.
- **Accounting:** No
- **Delta/Cumulative:** N/A
- **CMD-ID:** 151
- **Related CLI command:** None

SESS_UCHKPT_CMD_PGW_UPDATE_PDN_COMMON_PARAM

Reserved for future use.

SESS_UCHKPT_CMD_PGW_UPDATE_QOS

Reserved for future use.

SESS_UCHKPT_CMD_PGW_UPDATE_SGW_CHANGE

Reserved for future use.

SESS_UCHKPT_CMD_PGW_UPDATE_STATS

This micro-checkpoint periodically sends session statistics.

- **Time based:** Yes
- **Frequency:** Every five minutes
- **Event based:** No
- **Events:** N/A
- **Accounting:** Yes
- **Delta/Cumulative:** Cumulative
- **CMD-ID:** 65
- **Related CLI command:** None

Rf Interface Category

SESS_UCHKPT_CMD_ACS_ACCOUNTING_TYPE_QCI_RF

This micro-checkpoint indicates a change in the SDF+QCI-based Rf accounting buckets.

- **Time based:** Yes
- **Frequency:** 4 seconds for aamgr checkpoint and 18 seconds for GR checkpoint
- **Event based:** No
- **Events:** N/A
- **Accounting:** Yes
- **Delta/Cumulative:** Cumulative
- **CMD-ID:** 126
- **Related CLI command:** None

SESS_UCHKPT_CMD_ACS_ACCOUNTING_TYPE_QCI_RF_WITH_FC

This micro-checkpoint indicates complete SDF+QCI-based Rf accounting buckets.

- **Time based:** Yes
- **Frequency:** 4 seconds for aamgr checkpoint and 18 seconds for GR checkpoint
- **Event based:** No
- **Events:** Sent along with macro-checkpoint.
- **Accounting:** Yes
- **Delta/Cumulative:** Cumulative
- **CMD-ID:** 164
- **Related CLI command:** None

SESS_UCHKPT_CMD_ACS_ACCOUNTING_TYPE_RATING_GROUP_RF

This micro-checkpoint indicates a change in the SDF-based Rf accounting buckets.

- **Time based:** Yes
- **Frequency:** 4 seconds for aamgr checkpoint and 18 seconds for GR checkpoint
- **Event based:** No
- **Events:** N/A
- **Accounting:** Yes
- **Delta/Cumulative:** Cumulative
- **CMD-ID:** 125
- **Related CLI command:** None

SESS_UCHKPT_CMD_ACS_ACCOUNTING_TYPE_RATING_GROUP_RF_WITH_FC

This micro-checkpoint indicates complete SDF-based Rf accounting buckets.

- **Time based:** Yes
- **Frequency:** 4 seconds for aamgr checkpoint and 18 seconds for GR checkpoint;
- **Event based:** No
- **Events:** Sent along with macro-checkpoint.
- **Accounting:** Yes
- **Delta/Cumulative:** Cumulative
- **CMD-ID:** 163
- **Related CLI command:** None

S6b Interface Category

SESS_UCHKPT_CMD_S6B_INFO

This micro-checkpoint sends the Restoration Priority Indicator when reauthorization occurs over the S6b interface.

- **Time based:** No
- **Frequency:** N/A
- **Event based:** Yes

- **Events:** Occurs when an Sb6 reauthorization results in a change in value of the Restoration Priority Indicator.
- **Accounting:** No
- **Delta/Cumulative:** N/A
- **CMD-ID:** 202
- **Related CLI command:** None

SaMOG Category

SESS_UCHKPT_CMD_CGW_DELETE_BEARER

Reserved for future use.

SESS_UCHKPT_CMD_CGW_DELETE_PDN

This micro-checkpoint indicates a PDN connection has been deleted.

- **Time based:** No
- **Frequency:** N/A
- **Event based:** Yes
- **Events:** Occurs whenever SaMOG sends a Delete-Session-Req or upon receiving a Delete-Bearer-Request.
- **Accounting:** No
- **Delta/Cumulative:** N/A
- **CMD-ID:** 169
- **Related CLI command: show subscriber samog-only full**

SESS_UCHKPT_CMD_CGW_UPDATE_BEARER_QOS

This micro-checkpoint indicates a QoS update for the bearer.

- **Time based:** No
- **Frequency:** N/A
- **Event based:** Yes
- **Events:** Occurs when a change in Bearer QoS is received from the P-GW due to a reauthorization (AAR Received from AAA Server) or Update-Bearer-Request.
- **Accounting:** No
- **Delta/Cumulative:** N/A
- **CMD-ID:** 167
- **Related CLI command: show subscriber samog-only full**

SESS_UCHKPT_CMD_CGW_UPDATE_PDN

This micro-checkpoint indicates a PDN update for a change in APN-AMBR.

- **Time based:** No
- **Frequency:** N/A
- **Event based:** Yes
- **Events:** Occurs when a change in APN-AMBR is received from the P-GW due to a reauthorization (AAR Received from AAA Server) or Update-Bearer-Request.

- **Accounting:** No
- **Delta/Cumulative:** N/A
- **CMD-ID:** 168
- **Related CLI command: show subscriber samog-only full**

SESS_UCHKPT_CMD_CGW_UPDATE_STATS

Reserved for future use.

SESS_UCHKPT_CMD_CGW_UPDATE_UE_PARAM

Reserved for future use.

SESS_UCHKPT_CMD_SAMOG_ACCT_INTERIM_INFO

This micro-checkpoint is sent for a SaMOG session on receipt of an Accounting Req (INTERIM-UPDATE) from the WLC

- **Time based:** No
- **Frequency:** N/A
- **Event based:** Yes
- **Events:** Occurs on receipt of an Accounting Req (INTERIM-UPDATE) from the WLC.
- **Accounting:** No
- **Delta/Cumulative:** N/A
- **CMD-ID:** 177
- **Related CLI command: show subscriber samog-only full**

SESS_UCHKPT_CMD_SAMOG_ACCT_START_INFO

This micro-checkpoint is sent for a SaMOG session on receipt of an Accounting Req (START) from the WLC (Wireless LAN Controller).

- **Time based:** No
- **Frequency:** N/A
- **Event based:** Yes
- **Events:** Occurs when a Account Req (START) request is received from the WLC.
- **Accounting:** No
- **Delta/Cumulative:** N/A
- **CMD-ID:** 174
- **Related CLI command: show subscriber samog-only full**

SESS_UCHKPT_CMD_SAMOG_EOGRE_TUNNEL_INFO

This micro-checkpoint is sent for an Inter-RG handoff for EoGRE subscriber sessions. This checkpoint updates the VMAC Address and WLC EoGRE tunnel end-point address.

- **Time based:** No
- **Frequency:** N/A
- **Event based:** Yes
- **Events:** Occurs whenever a DHCP-Discover message is received over a different EoGRE tunnel.
- **Accounting:** No

- **Delta/Cumulative:** N/A
- **CMD-ID:** 201
- **Related CLI command: show subscriber samog-only full**

SESS_UCHKPT_CMD_SAMOG_GTPV1_UPDATE_PDN_INFO

This micro-checkpoint is sent for a SaMOG session upon receipt of an Update-PDP-Context-Req from the GGSN to update the PDN information.

- **Time based:** No
- **Frequency:** N/A
- **Event based:** Yes
- **Events:** Occurs after successful SaMOG processing of an Update-PDP-Context-Req from the GGSN.
- **Accounting:** No
- **Delta/Cumulative:** N/A
- **CMD-ID:** 191
- **Related CLI command: show subscriber samog-only full**

SESS_UCHKPT_CMD_SAMOG_HANDOFF_AUTHEN_INFO

This micro-checkpoint is sent for a SaMOG session that is Re-authenticating the subscriber while the subscriber session is in Handoff state.

- **Time based:** No
- **Frequency:** N/A
- **Event based:** Yes
- **Events:** Occurs on completion of Re-Authentication for an existing SaMOG subscriber session currently in Handoff state.
- **Accounting:** No
- **Delta/Cumulative:** N/A
- **CMD-ID:** 176
- **Related CLI command: show subscriber samog-only full**

SESS_UCHKPT_CMD_SAMOG_HANDOFF_INIT_INFO

This micro-checkpoint is sent for a SaMOG session on receipt of an Accounting Req (STOP) from the WLC (Wireless LAN Controller).

SaMOG will delay handoff as it expects an Accounting Req (START) from the subscriber.

- **Time based:** No
- **Frequency:** N/A
- **Event based:** Yes
- **Events:** Occurs when a Account Req (STOP) request is received from the WLC.
- **Accounting:** No
- **Delta/Cumulative:** N/A
- **CMD-ID:** 175
- **Related CLI command: show subscriber samog-only full**

SESS_UCHKPT_CMD_SAMOG_LI_PROV_INFO

This micro-checkpoint is sent for a SaMOG session that is on lawful intercept (LI) Active-Camp-on mode.

- **Time based:** No
- **Frequency:** N/A
- **Event based:** Yes
- **Events:** Occurs after an LI trigger is received after SaMOG session has been created.
- **Accounting:** No
- **Delta/Cumulative:** N/A
- **CMD-ID:** 189
- **Related CLI command: show subscriber samog-only full**

SESS_UCHKPT_CMD_SAMOG_MIPV6_TIMER_INFO

This micro-checkpoint updates the Binding Cache Life timer and MIPv6 biding status for a SaMOG session.

- **Time based:** No
- **Frequency:** N/A
- **Event based:** Yes
- **Events:** Occurs whenever a PMIPv6 PBU is received with a lifetime of zero from the WLC.
- **Accounting:** No
- **Delta/Cumulative:** N/A
- **CMD-ID:** 190
- **Related CLI command: show subscriber samog-only full**

SESS_UCHKPT_CMD_SAMOG_MULTI_ROUND_AUTHEN_INFO

This micro-checkpoint is sent for a SaMOG session when SaMOG is waiting on the UE after sending an Access-Challenge while Re-authenticating the subscriber session.

- **Time based:** No
- **Frequency:** N/A
- **Event based:** Yes
- **Events:** Occurs after SaMOG sends an Access-Challenge for an existing SaMOG subscriber session during Re-authentication.
- **Accounting:** No
- **Delta/Cumulative:** N/A
- **CMD-ID:** 184
- **Related CLI command: show subscriber samog-only full**

SESS_UCHKPT_CMD_SAMOG_REAUTHEN_INFO

This micro-checkpoint is sent for a SaMOG session when subscriber Re-authentication is completed.

- **Time based:** No
- **Frequency:** N/A
- **Event based:** Yes
- **Events:** Occurs on completion of Re-Authentication for an existing SaMOG subscriber session.
- **Accounting:** No
- **Delta/Cumulative:** N/A

- **CMD-ID:** 172
- **Related CLI command: show subscriber samog-only full**

SESS_UCHKPT_CMD_SAMOG_REAUTHOR_INFO

This micro-checkpoint is sent for a SaMOG session when subscriber Re-authorization is completed.

- **Time based:** No
- **Frequency:** N/A
- **Event based:** Yes
- **Events:** Occurs on receiving and successfully processing AAR from the AAA Server to re-authorize the subscriber
- **Accounting:** No
- **Delta/Cumulative:** N/A
- **CMD-ID:** 173
- **Related CLI command: show subscriber samog-only full**

SESS_UCHKPT_CMD_SAMOG_REAUTHOR_INFO

APPENDIX E

VPC-DI SDR CLI Command Strings

This appendix identifies the StarOS CLI command strings that can be entered for a record section via the **support record section** command in the Global Configuration Mode. The string must be entered within double quotation marks (" ") to be recognized. This table also indicates default and nondefault strings.

For detailed command string information, refer to the *Command Line Interface Reference* or the online Help for the command.

This table also indicates default and nondefault strings. It reflects the output sequence of the **show support collection definitions** command.

Table 38: VPC-DI SDR CLI Command Strings

No.	Default SDR	Command String
0	Enabled	"show version verbose"
1	Enabled	"show clock"
2	Enabled	"show clock universal"
3	Enabled	"show configuration"
4	Enabled	"show_profile"
5	Enabled	"show context"
6	Enabled	"show boot"
7	Enabled	"show boot initial-config"
8	Enabled	"show system uptime"
9	Disabled	"show license information"
10	Disabled	"show license history"
11	Disabled	"show hardware inventory"
12	Disabled	"show hardware version"
13	Disabled	"show card hardware"
25	Disabled	"show card info"
26	Enabled	"show card diag

No.	Default SDR	Command String
27	Enabled	"show card table all"
28	Enabled	"show port table all"
29	Enabled	"show port info"
30	Enabled	"show port utilization table"
31	Enabled	"show data-path congestion"
32	Disabled	"show npu details"
33	Disabled	"show lagmgr details"
34	Enabled	"show fans"
36	Enabled	"show power chassis"
37	Enabled	"show temperature"
39	Disabled	"show alarm audible"
40	Disabled	"show alarm central-office"
41	Disabled	"show alarm outstanding"
42	Disabled	"show alarm statistics"
43	Enabled	"show cpu table"
44	Disabled	"show cpu info verbose"
45	Enabled	"show cpu errors verbose"
46	Enabled	"show cpu performance verbose"
47	Disabled	"show resources"
48	Disabled	"show task table"
49	Disabled	"show task memory"
50	Disabled	"show task memory max"
51	Disabled	"show task resources"
52	Disabled	"show task resources max"
53	Enabled	"show crash list"
54	Enabled	"show crash all"
55	Disabled	"show persistdump list"
56	Disabled	"show persistdump display"
57	Enabled	"show snmp trap history verbose"
58	Disabled	"show snmp trap statistics verbose"
59	Enabled	"show logs"
63	Disabled	"show messenger settings"

No.	Default SDR	Command String
64	Enabled	"show messenger nameservice"
65	Enabled	"show messenger statistics"
66	Enabled	"show messenger bounces"
67	Disabled	"debug limits checkup detailed"
68	Disabled	"show plugin"
69	Disabled	"show module"
70	Disabled	"show ppp statistics"
71	Disabled	"show rsvp statistics"
72	Enabled	"show session disconnect-reasons verbose"
73	Disabled	"show apn statistics all"
74	Disabled	"show ipsg statistics"
75	Disabled	"show pdsn-service all"
76	Disabled	"show hsgw-service all"
77	Disabled	"show hsgw-service statistics all"
78	Disabled	"show epdg-service all counters"
79	Disabled	"show epdg-service statistics"
80	Disabled	"show fa-service all"
81	Disabled	"show ha-service all"
82	Disabled	"show mag-service all"
83	Disabled	"show mipv6ha-service all"
84	Disabled	"show lma-service all"
85	Disabled	"show dhcp-service all"
86	Disabled	"show sgsn-service all"
87	Disabled	"show sgsn sessmgr all memory statistics"
88	Disabled	"show operator-policy all"
89	Disabled	"show call-control-profile all"
90	Disabled	"show apn-profile all"
91	Disabled	"show imei-profile all"
92	Disabled	"show gprs-service all"
93	Disabled	"show iups-service all"
94	Disabled	"show sgtp-service all"

No.	Default SDR	Command String
95	Disabled	"show map-service all"
96	Disabled	"show gs-service all"
97	Disabled	"show ggsn-service all"
98	Disabled	"show ggsn-service sgsn-table"
99	Disabled	"show cscf service all"
100	Disabled	"show cscf service diameter policy-control statistics"
101	Disabled	"show cscf service diameter location-info statistics"
102	Disabled	"show cscf service li-packet-cable statistics"
103	Disabled	"show cscf peer-servers full"
104	Disabled	"show demux-mgr statistics cscfmgr all"
105	Disabled	"show lac-service all"
106	Disabled	"show lns-service all"
107	Disabled	"show pdsnclosedrp-service all"
108	Enabled	"show subscriber summary"
109	Enabled	"show connproxy sockets all"
110	Disabled	"show session progress"
111	Disabled	"show session subsystem data-info verbose"
112	Disabled	"show session subsystem full data-info"
113	Disabled	"show session subsystem facility sessmgr all debug-info"
114	Disabled	"show sessctrl config-reconciliation statistics"
115	Disabled	"show rp statistics"
116	Disabled	"show mipfa statistics"
117	Disabled	"show mipha statistics"
118	Disabled	"show mipv6ha statistics"
119	Disabled	"show lma-service statistics"
120	Disabled	"show mag-service statistics"
121	Disabled	"show cli configuration-monitor"

No.	Default SDR	Command String
122	Enabled	"show srp info"
123	Enabled	"show srp checkpoint statistics"
124	Disabled	"show srp checkpoint statistics verbose"
125	Disabled	"show srp checkpoint statistics sessmgr all"
126	Disabled	"show srp checkpoint statistics ipsecmgr all"
127	Enabled	"show srp checkpoint statistics sessmgr all write-list-stats"
128	Disabled	"show srp monitor"
129	Enabled	"show srp monitor all"
130	Disabled	"show srp monitor diameter debug"
131	Enabled	"show srp statistics"
132	Disabled	"show srp call-loss statistics"
133	Disabled	"show srp audit-statistics"
134	Disabled	"show gtpc statistics verbose"
135	Enabled	"show gtpu statistics verbose"
136	Enabled	"show gtpu debug-info"
137	Enabled	"show gmm-sm statistics verbose"
138	Enabled	"show sgtpc statistics verbose"
139	Enabled	"show sgtpu statistics"
140	Disabled	"show ss7-routing-domain all sctp asp all status peer-server all peer-server-process all verbose"
141	Enabled	"show ss7-routing-domain all sctp asp all statistics gen"
142	Disabled	"show ss7-routing-domain all m3ua status peer-server all"
143	Disabled	"show ss7-routing-domain all m3ua statistics peer-server all peer-server-process all"
144	Disabled	"show ss7-routing-domain all qsaal statistics linkset all link all"
145	Disabled	"show ss7-routing-domain all sscf statistics linkset all link all"

No.	Default SDR	Command String
146	Disabled	"show ss7-routing-domain all mtp3 status linkset all link all"
147	Disabled	"show ss7-routing-domain all mtp3 statistics gen"
148	Disabled	"show ss7-routing-domain all mtp3 statistics linkset all link all"
149	Disabled	"show ss7-routing-domain all routes"
150	Disabled	"show sccp-network all status all"
151	Disabled	"show global-title-translation association"
152	Disabled	"show global-title-translation address-map"
153	Enabled	"show egtpc peers"
154	Disabled	"show egtpc statistics interface mme"
155	Enabled	"show egtpc statistics interface sgsn"
156	Enabled	"show egtpc statistics interface sgw-ingress"
157	Enabled	"show egtpc statistics interface sgw-egress"
158	Enabled	"show egtpc statistics interface pgw-ingress"
159	Enabled	"show egtpc statistics interface cgw-egress"
160	Enabled	"show egtpc statistics interface epdg-egress"
161	Disabled	"show egtp-service all"
162	Disabled	"show gtpu-service all"
163	Disabled	"show pgw-service all"
164	Disabled	"show sgw-service all"
165	Disabled	"show saegw-service all"
166	Disabled	"show henbgw-access-service statistics"
167	Disabled	"show henbgw-network-service statistics"
168	Disabled	"show mme-service all"

No.	Default SDR	Command String
169	Disabled	"show mme-service enodeb-association full all"
170	Disabled	"show mme-service statistics debug"
171	Disabled	"show mme-service db statistics"
172	Disabled	"show sgs-service all"
173	Disabled	"show sgs-service vlr-status full"
174	Disabled	"show sgs-service statistics all"
175	Enabled	"show sgw-service statistics all"
176	Disabled	"show saegw-service statistics all verbose"
177	Disabled	"show saegw-service statistics all function sgw verbose"
178	Disabled	"show saegw-service statistics all function pgw verbose"
179	Enabled	"show pgw-service statistics all"
180	Disabled	"show sccp statistics"
181	Disabled	"show tcap statistics"
182	Disabled	"show map statistics"
183	Disabled	"show sms statistics"
184	Disabled	"show pdg-service statistics"
185	Disabled	"show hnbgw sessmgr all memory statistics"
186	Disabled	"show hnbgw sessmgr all internal statistics"
187	Disabled	"show hnbgw disconnect-reasons"
188	Disabled	"show cs-network statistics"
189	Disabled	"show ps-network statistics"
190	Disabled	"show hnbgw statistics"
191	Disabled	"show hnbgw counters"
192	Disabled	"show demux-mgr statistics hnbmgr full"
193	Disabled	"show demuxmgr statistics bngmgr all"
194	Disabled	"show alcap statistics"

No.	Default SDR	Command String
195	Disabled	"show pdg-service statistics micro-tunnel"
196	Disabled	"show pdg-service statistics transport"
197	Disabled	"show demuxmgr statistics a11mgr all"
198	Disabled	"show demuxmgr statistics famgr all"
199	Disabled	"show demuxmgr statistics hamgr all"
200	Disabled	"show demuxmgr statistics l2tpmgr all"
201	Disabled	"show demuxmgr statistics ipsgmgr all"
202	Enabled	"show demuxmgr statistics sgtpcmgr all"
203	Disabled	"show demuxmgr statistics imsimgr all"
204	Enabled	"show demuxmgr statistics gtpcmgr all"
205	Enabled	"show demuxmgr statistics egtpinmgr all"
206	Disabled	"show demuxmgr statistics egtpegmgr all"
207	Disabled	"show demuxmgr statistics pdgdmgr all"
208	Enabled	"show demuxmgr statistics gtpumgr all"
209	Disabled	"show bcmcs statistics all"
210	Enabled	"show linkmgr all parser statistics all"
211	Disabled	"show gtpp accounting servers"
212	Disabled	"show gtpp statistics verbose"
213	Disabled	"show gtpp counters all"
214	Disabled	"show gtpp storage-server"
215	Disabled	"show gtpp storage-server statistics verbose"

No.	Default SDR	Command String
216	Disabled	"show gtpp storage-server local file statistics verbose" **Important** After you configure this command, you must save the configuration and then reload the chassis for the command to take effect. For information on saving the configuration file and reloading the chassis, refer to the *System Administration Guide* for your deployment.
217	Disabled	"show gtpp storage-server local file counters all" **Important** After you configure this command, you must save the configuration and then reload the chassis for the command to take effect. For information on saving the configuration file and reloading the chassis, refer to the *System Administration Guide* for your deployment.
218	Disabled	"show gtpp storage-server streaming file statistics verbose"
219	Disabled	"show gtpp storage-server streaming file counters all"
220	Disabled	"show gtpp group all"
221	Enabled	"show hd-storage-policy statistics all verbose"
222	Enabled	"show hd-storage-policy counters all verbose"
223	Disabled	"show dhcp statistics verbose"
224	Disabled	"show npu table"
225	Disabled	"show npu sf hw-info"

No.	Default SDR	Command String
229	Disabled	"show l2tp statistics"
230	Enabled	"show fabric asr5500"
231	Enabled	"show vpn subsystem facility vpnmgr"
232	Enabled	"show session recovery status verbose"
233	Enabled	"show clock all"
234	Disabled	"show sntp statistics verbose"
235	Disabled	"show llc statistics verbose"
236	Disabled	"show bssgp statistics verbose"
237	Disabled	"show bssap+ statistics verbose"
238	Disabled	"show network-service-entity ip-config"
239	Disabled	"show network-service-entity fr-config"
240	Disabled	"show gprsns statistics sns-msg-stats"
241	Disabled	"show radius authentication servers detail"
242	Disabled	"show radius accounting servers detail"
243	Enabled	"show radius counters all"
245	Enabled	"show rct stats"
246	Enabled	"show heartbeat stats hatcpus"
247	Disabled	"show ntp associations all"
248	Disabled	"show npu details"
249	Disabled	"show active-charging service all"
250	Disabled	"show active-charging tcp-proxy statistics all verbose debug-info"
251	Disabled	"show active-charging edr-udr-file flow-control-counters verbose debug-only"
252	Disabled	"show active-charging service statistics"
253	Disabled	"show active-charging analyzer statistics"

No.	Default SDR	Command String
254	Disabled	"show active-charging dns-learnt-ip-addresses statistics sessmgr all verbose"
255	Disabled	"show active-charging analyzer statistics name ip verbose"
256	Disabled	"show active-charging analyzer statistics name ipv6 verbose"
257	Disabled	"show active-charging analyzer statistics name tcp verbose"
258	Disabled	"show active-charging analyzer statistics name http verbose"
259	Disabled	"show active-charging charging-action statistics"
260	Disabled	"show active-charging rulebase statistics"
261	Disabled	"show active-charging ruledef statistics all charging"
262	Enabled	"show active-charging ruledef statistics all firewall wide"
263	Disabled	"show active-charging regex status all"
264	Disabled	"show active-charging regex statistics memory summary"
265	Disabled	"show active-charging regex statistics ruledef summary"
266	Disabled	"show active-charging edr-format statistics"
267	Disabled	"show active-charging subsystem all debug-only"
268	Disabled	"debug acsmgr show flow-stats cumulative all-flows"
269	Disabled	"debug acsmgr show flow-stats cumulative http"
270	Disabled	"debug acsmgr show flow-stats cumulative ip"
271	Disabled	"debug acsmgr show flow-stats cumulative tcp"
272	Disabled	"debug acsmgr show flow-stats cumulative udp"

No.	Default SDR	Command String
273	Disabled	"debug acsmgr show flow-stats max-simultaneous-flows all-flows"
274	Disabled	"debug acsmgr show flow-stats max-simultaneous-flows http"
275	Disabled	"debug acsmgr show flow-stats max-simultaneous-flows ip"
276	Disabled	"debug acsmgr show flow-stats max-simultaneous-flows tcp"
277	Disabled	"debug acsmgr show flow-stats max-simultaneous-flows udp"
278	Disabled	"debug acsmgr show flow-stats duration-based all-flows"
279	Disabled	"debug acsmgr show flow-stats duration-based tcp"
280	Disabled	"debug acsmgr show flow-stats duration-based udp"
281	Disabled	"debug acsmgr show flow-stats lifetime-based all-flows"
282	Disabled	"debug acsmgr show p2p detection-params sct"
283	Disabled	"debug acsmgr show rule-optimization-information"
284	Disabled	"debug sessmgr charging-service show-stats all"
285	Disabled	"debug acsmgr show memory usage"
286	Disabled	"debug aaamgr show memory usage"
287	Disabled	"show active-charging credit-control statistics debug-info"
288	Disabled	"show active-charging credit-control session-states"
289	Disabled	"show active-charging credit-control statistics"
290	Disabled	"show diameter endpoints all"
291	Disabled	"show diameter endpoints all debug-info"
292	Disabled	"show diameter route table debug-info"

No.	Default SDR	Command String
293	Disabled	"show diameter peers full debug"
294	Disabled	"show diameter aaa-statistics"
295	Disabled	"show diameter aaa-statistics all"
296	Disabled	"show diameter aaa-statistics debug-info"
297	Disabled	"show diameter accounting servers debug-info"
298	Disabled	"show diameter authentication servers debug-info"
299	Disabled	"show diameter statistics"
300	Disabled	"show diameter statistics debug-info"
301	Disabled	"show diameter statistics proxy"
302	Disabled	"show diameter statistics proxy debug-info"
303	Disabled	"show diameter dynamic-dictionary all contents"
304	Disabled	"show active-charging edr-udr-file statistics"
305	Disabled	"show active-charging firewall statistics debug-info"
306	Disabled	"show active-charging nat statistics"
307	Disabled	"show demuxmgr statistics asngwmgr all"
308	Disabled	"show asngw-service all"
309	Disabled	"show asngw-service statistics verbose"
310	Disabled	"show demuxmgr statistics asnpcmgr all"
311	Disabled	"show asnpc-service all"
312	Disabled	"show asnpc-service statistics verbose"
313	Disabled	"show demuxmgr statistics phsgwmgr all"
314	Disabled	"show phsgw-service all"
315	Disabled	"show phsgw-service statistics verbose"

No.	Default SDR	Command String
316	Disabled	"show demuxmgr statistics phspcmgr all"
317	Disabled	"show phspc-service all"
318	Disabled	"show phspc-service statistics verbose"
319	Disabled	"show demuxmgr statistics magmgr all"
320	Disabled	"show active-charging content-filtering category policy-id all"
321	Disabled	"show content-filtering category database all verbose"
322	Disabled	"show content-filtering category database facility srdbmgr all verbose"
323	Disabled	"show content-filtering category statistics"
324	Disabled	"show content-filtering category statistics facility srdbmgr all"
325	Disabled	"show active-charging content-filtering category statistics"
326	Disabled	"show active-charging content-filtering server-group statistics verbose"
327	Disabled	"show active-charging url-blacklisting statistics"
328	Disabled	"show url-blacklisting database all"
329	Disabled	"show url-blacklisting database facility acsmgr all"
330	Disabled	"show active-charging tethering-detection database"
331	Disabled	"show active-charging tethering-detection database sessmgr all"
332	Disabled	"show active-charging tethering-detection statistics"
333	Disabled	"show ims-authorization service statistics"
334	Disabled	"show ims-authorization policy-control statistics"

No.	Default SDR	Command String
335	Disabled	"show ims-authorization policy-control statistics debug-info"
336	Disabled	"show local-policy statistics summary"
337	Disabled	"show rohc statistics"
338	Disabled	"show dns client statistics"
339	Disabled	"show hss-peer-service service all"
340	Disabled	"show ipms status all"
341	Disabled	"show ipms status debug-info"
342	Disabled	"show kvstore"
343	Disabled	"show kvstore verbose"
344	Disabled	"show kvstore kvclient"
345	Disabled	"show kvstore kvmgr"
346	Disabled	"show pcc-service all"
347	Disabled	"show pcc-service statistics all"
348	Disabled	"show pcc-policy service all"
349	Disabled	"show pcc-policy service statistics all"
350	Disabled	"show pcc-quota service all"
351	Disabled	"show pcc-quota service statistics all"
352	Disabled	"show pcc-af service all"
353	Disabled	"show pcc-af service statistics all"
354	Disabled	"show pcc-sp-endpoint all"
355	Disabled	"show pcc-sp-endpoint statistics all"
356	Disabled	"show event-notif server all"
357	Disabled	"show event-notif statistics"
358	Disabled	"show demux-mgr statistics bindmux all"
359	Disabled	"show congestion-control configuration"
360	Disabled	"show congestion-control statistics mme full"

No.	Default SDR	Command String
361	Disabled	"show congestion-control statistics imsimgr all full"
365	Disabled	"show cli history"

Notes:

- Enabled = Included in default record section
- Disabled = Not included in default record section

APPENDIX F

VPC Commands

This chapter details commands that were introduced or changed on the VPC since StarOS Release 19.1.

access-type

This command is used to configure the NB-IoT RAT per TAI database.

Product

MME

Privilege

Administrator

Command Modes

Exec > Global Configuration > LTE Policy Configuration > LTE TAI Management Database Configuration

configure > lte-policy > tai-mgmt-db *db_name*

Entering the above command sequence results in the following prompt:

`[local]`*`host_name`*`(tai-mgmt-db)#`

Syntax Description

`[ no ] access-type nb-iot`

no

Removes the configured access type for the TAI database.

nb-iot

Configures the access type as NB-IoT for a TAI database.

Usage Guidelines

The LTE TAI Management Database Configuration Mode is used to create and manage the LTE Tracking Area Identifier (TAI) management database on the system. Enter the TAI Management Database Configuration Mode for an existing or newly defined database. This command is also used to remove an existing database. Use this command to configure the access type of a TAC or group of TACs as NB-IoT RAT. As per 3GPP standards, the same TAC cannot belong to both EUTRAN and NB-IoT RATs. This command is not enabled by default. The default RAT is WB-EUTRAN.

Example

The following command is used to configure the access type as NB-IoT:

`access-type nb-iot`

access-type

This command is used to configure the NB-IoT RAT per TAI object.

Product

MME

Privilege

Administrator

Command Modes

Exec > Global Configuration > LTE Policy Configuration > LTE TAI Management Database Configuration > LTE TAI Management Object Configuration

configure > **lte-policy** > **tai-mgmt-db** *db_name* > **tai-mgmt-obj** *obj_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(tai-mgmt-obj)#
```

Syntax Description

[no] access-type nb-iot

no

Removes the configured access type for the TAI object.

nb-iot

Configures the access type as NB-IoT for a TAI object.

Usage Guidelines

The LTE TAI Management Object Configuration Mode is used to create and manage the LTE Tracking Area Identifiers for the TAI database. This mode is used to create, remove or modify the existing LTE Tracking Area Identifier (TAI) object configurations. Use this command to configure the access type of a TAC or group of TACs as NB-IoT RAT. As per 3GPP standards, the same TAC cannot belong to both EUTRAN and NB-IoT RATs. This command is not enabled by default. The default RAT is WB-EUTRAN.

Example

The following command is used to configure the access type as NB-IoT:

access-type nb-iot

bfd

Configures Bidirectional Forwarding Detection (BFD) interface parameters.

Product

All

Privilege

Security Administrator, Administrator

Command Modes

Exec > Global Configuration > Context Configuration > Ethernet Interface Configuration

configure > **context** *context_name* > **interface** *interface_name* **broadcast**

Entering the above command sequence results in the following prompt:

```
[context_name]host_name(config-if-eth)#
```

Syntax Description

```
[no] bfd { echo [echo-interval interval_num] | interval interval_num }
   min_rx milliseconds multiplier value
```

no

Disables the specified option on this interface.

echo

Enables BFD echo mode.

BFD echo mode works with asynchronous BFD. Echo packets are sent by the forwarding engine and forwarded back along the same path in order to perform detection—the BFD session at the other end does not participate in the actual forwarding of the echo packets. The echo function and the forwarding engine are responsible for the detection process, therefore the number of BFD control packets that are sent out between two BFD neighbors is reduced.

Since the forwarding engine is testing the forwarding path on the remote (neighbor) system without involving the remote system, there is an opportunity to improve the interpacket delay variance, thereby achieving quicker failure detection times than when using BFD Version 0 with BFD control packets for the BFD session.

echo-interval *interval_num*

Specifies the transmit interval between BFD echo packets. The default interval is 150 ms. The range is from 0 to 999 ms. (VPC only)

interval *interval_num*

Specifies the transmit interval (in milliseconds) between BFD packets.

- *For releases prior to 17.0*, *interval_num* is an integer from 50 through 999. (Default 50)
- *For release 17.0 onwards*, *interval_num* is an integer from 50 through 10000. (Default 50)

min_rx *milliseconds*

Specifies the receive interval in milliseconds for control packets.

- *For releases prior to 17.0*, *milliseconds* is an integer from 50 through 999. (Default 50)
- *For release 17.0 onwards*, *milliseconds* is an integer from 50 through 10000. (Default 50)

multiplier *value*

Specifies the value used to compute the hold-down time as a number from 3 to 50.

Usage Guidelines

Specify BFD parameters including echo mode and the transmit interval between BFD packets.

Example

To apply enable echo mode on this interface, use the following command:

```
bfd echo
```

The following command sets BFD interval parameters:

```
bfd interval 3000 min_rx 300 multiplier 3
```

ciot-optimisation

This command is used to configure Control Plane (CP) CIoT optimization for an UE.

Product

MME

Privilege

Security Administrator, Administrator

Command Modes

Exec > Global Configuration > Call Control Profile Configuration

configure > **call-control-profile** *profile_name*

Entering the above command sequence results in the following prompt:

`[local]`*`host_name`*`(config-call-control-profile-`*`profile_name`*`)#`

Syntax Description

```
ciot-optimisation { cp-optimisation { access-type { all | nb-iot |
wb-eutran } |  ciot-capable-ue } | eps-attach-wo-pdn access-type { all |
 nb- iot | wb-eutran } }
remove ciot-optimisation cp-optimisation  ciot-capable-ue
remove ciot-optimisation eps-attach-wo-pdn access-type { all | nb-iot |
wb-eutran }
```

remove

The keyword remove deletes the existing configuration.

cp-optimisation

Use this keyword to enable Control Plane optimization for an UE.

access-type

Use this keyword to specify the access type extension on which control plane optimization should be enabled. Control plane optimization and EPS attach without PDN can be enabled on both NB-IoT and WB-EUTRAN RATs or on either of them.

ciot-capable-ue

Uses only the ue-nw-capability to determine whether CP optimization or not.

all

Use this keyword to enable control plane optimization on both RAT types WB-EUTRAN and NB-IOT. This keyword is provided to the operator for the ease of configuring. Both NB-IoT and WB-EUTRAN will be considered as two independent access types for all functions.

nb-iot

Use this keyword to enable control plane optimization on the RAT type NB-IoT.

wb-eutran

Use this keyword to enable control plane optimization on the RAT type WB-EUTRAN.

eps-attach-wo-pdn

Use this keyword to enable EPS attach without PDN support for an UE.

Usage Guidelines

Use this command to configure the control plane optimization on the RAT type and to configure EPS attach without PDN support for UE. This command is not enabled by default. The call-control-profile can be associated with the operator-policy or with IME-TAC group, therefore it is possible to either enable or disable CIoT optimization on a per subscriber (IMSI) basis or on a group of subscribers or on per group of IMEI basis. CIoT optimization can be enabled on both NB-IoT and WB-EUTRAN RATs or on either of them. Enabling one RAT type does not disable the other RAT type.

Example

Use the following command to configure control plane optimization by specifying the access type as NB-IoT:

```
ciot-optimisation cp-optimisation access-type nb-iot
```

Use the following command to configure EPS attach without PDN support for UE, specify the access type as WB-EUTRAN:

```
ciot-optimisation eps-attach-wo-pdn access-type wb-eutran
```

debug bfd

Enables or disables the debug options for Bidirectional Forwarding Detection BFD) debugging. If logging is enabled, results are sent to the logging system.

Product

All

Privilege

Security Administrator, Administrator, Operator

Command Modes

Exec

The following prompt is displayed in the Exec mode:

```
[local]host_name#
```

Syntax Description

```
[ no ] debug bfd [all | events ipc-error | ipc-events | nsm | packet |
session]
    [level-1 | level-2 | level-3]
```

no

Indicates the IP debugging is to be disabled for the IP interfaces/function specified.

bfd ***name***

Specifies which IP interfaces/function to debug.

all: enables debug for all BFD items.

events: enables debug for BFD events.

ipc-error: enables debug for BFD Inter-process communication (IPC) errors.

ipc-events: enables debug for BFD Inter-process communication (IPC) events.

nsm: enables debug for BFD Network Service Manager messages.

packet: enables debug for BFD packets.

session: enables debug for BFD sessions.

level-1 | level-2 | level-3

Optionally specifies the amount of information provided by the debug command:

- Level-1 debugging shows errors, warnings, and some critical one time events. Level-1 is the default.
- Level-2 debugging shows errors, warnings, and all events.
- Level-3 debugging shows errors, warnings, all events and is much more verbose.

Usage Guidelines

The **debug** command is valuable when troubleshooting network problems with BFD-enabled BGP routers. The debugging is stopped by using the **no** keyword.

Caution Issuing this command could negatively impact system performance depending on system configuration and/or loading.

Example

The following commands enable/disable debugging for BFD.

```
debug bfd
no debug bfd
```

delay-tolerant-pdn

Configures Delay Tolerant behavior for PDN connection to support UE in Power Saving Mode.

Product

P-GW

S-GW

SAEGW

Privilege

Security Administrator, Administrator

Command Modes

Exec > Global Configuration > Context Configuration > APN Configuration

configure > **context** *context_name* > **apn** *apn_name*

Entering the above command sequence results in the following prompt:

```
[context_name]host_name(config-apn)#
```

Syntax Description

delay-tolerant-pdn max-control-signal-buffer *1-4*
no delay-tolerant-pdn

no

Removes and restores the configuration to its default value.

max-control-signal-buffer ***1-4***

Configures maximum number of P-GW initiated control signaling messages to be buffered (range 1 to 4) when the UE is in Power Saving Mode (PSM).

Usage Guidelines

When the CLI is configured, it indicates that the PDN supports delay tolerant behavior. Also, the number of control signals that can be buffered is indicated by **max-control-signal-buffer**. When a new Rule is sent to update/create bearer, the number of transactions that will be buffered gets restricted to 4.

By default, the command is disabled and eDRX support is not applicable.

This CLI command takes effect during new call set-up or during handoff procedure to S5/S8 interface.

Example

The following command configures *3* P-GW initiated control signaling messages to be buffered when UE is in Power Saving mode.

delay-tolerant-pdn max-control-signal-buffer *3*

diameter

This command configures the diameter interface for an SCEF service.

Product

MME

Privilege

Administrator

Command Modes

Exec > Global Configuration > Context Configuration > SCEF Service Configuration

configure > **context** *context_name* > **scef-service** *service_name*

Entering the above command sequence results in the following prompt:

```
[context_name]host_name(config-scef-service)#
```

Syntax

```
diameter { dictionary standard | endpoint endpoint_name }
[ no ] diameter endpoint
[ default ] diameter dictionary
```

no

The prefix no disables the configuration.

default

The prefix default assigns or restores the default vale for the selected parameters.

endpoint *endpoint_value*

This command configures the diameter endpoint.

endpoint_name must be for the Diameter server expressed as an alphanumeric string of 1 through 63 characters.

dictionary standard

This command configures the dictionary to be used for the interface. The above configuration can be used to configure the transfer of Non-IP data over SCEF at the T6a diameter interface

Usage Guidelines

Use this command to configure the diameter interface. The above mentioned commands can be used to configure the transfer of Non-IP data over SCEF at the T6a diameter interface.

A diameter endpoint name must be specified. It is not recommended to remove the diameter endpoint when there are active calls on the system. Hence, please adhere to the 'Method of Procedure' to remove the endpoint. Otherwise, the system behavior would be undefined.

Example

The following command configures the diameter with an endpoint t6a-endpoint:

```
diameter endpoint t6a-endpoint
```

Example

The following command configures the diameter standard dictionary:

```
diameter dictionary standard
```

edrx

This command enables Extended Discontinuous Reception (eDRX) and configures its respective parameters, on the MME.

Product

MME

Privilege

Administrator

Command Modes

Exec > Global Configuration > Call Control Profile Configuration

configure > **call-control-profile** *profile_name*

Entering the above command sequence results in the following prompt:

`[local]host_name(config-call-control-profile-profile_name)#`

Syntax

edrx { ptw *ptw_value* **edrx-cycle** *cycle_length_value* **| ue-requested } [dl-buf-duration [packet-count** *packet_count_value* **]]**
remove edrx

remove

The keyword **remove**disables the eDRX configuration on the MME.

ptw *ptw_value*

This keyword is used to configure the PTW value.

In releases prior to 21.2: The *ptw_value* is an integer ranging from "0" up to "20".

In 21.2 and later releases: The *ptw_value* is an integer ranging from "0" up to "15".

ue-requested

The keyword **ue-requested**specifies the UE requested values of the Paging Time Window (PTW) and the eDRX cycle length received from the UE in the Attach Request/TAU Request message be accepted.

edrx-cycle *cycle_length_value*

The keyword **edrx-cycle**is used to configure the eDRX cycle length. The *cycle_length_value* is an integer value from " 512" up to "262144". It is a multiple of 2 starting from 512 up to 262144 (for example: 512, 1024, 2048, and so on).

dl-buf-duration

The keyword **dl-buf-duration** is used to send downlink buffer duration in DDN ACK when unable to page UE.

packet-count*packet_count_value*

The keyword **packet-count** is used to send 'DL Buffering Suggested Packet Count' in DDN ACK when unable to page UE. The *packet_count_value* is an integer value from "0" up to "65535". If the *packet_count_value* is not configured locally, the subscription provided value for the *packet_count_value* is used. The subscription value can be "0" in which case packet count IE will not be sent for that subscriber even if it is configured locally.

Usage Guidelines

Use this command to enable eDRX on the MME. This command is configured as part of the eDRX feature for MME - it allows UEs to connect to the network on a need basis. With eDRX, a device can remain inactive or in sleep mode for minutes, hours or even days based on the H-SFN synchronization time (UTC Time). The H-SFN synchronization time for eDRX is configured at an MME-Service level. See *MME Service Configuration*

Mode Commands chapter for configuration information on H-SFN synchronization. This command is not enabled by default.

Example

The following command is used to configure the PTW and eDRX cycle length. The command is also used to send the downlink buffer duration in the DDN ACK along with a suggested packet count:

```
edrx ptw 10 edrx-cycle 512 dl-buf-duration packet-count 10
```

gtpc

Configure the GPRS Tunneling Protocol Control (GTP-C) plane settings for this service.

Product

ePDG

MME

P-GW

S-GW

SAEGW

SaMOG

SGSN

Privilege

Administrator

Command Modes

Exec > Global Configuration > Context Configuration > eGTP Service Configuration

configure > **context** *context_name* > **egtp-service** *service_name*

Entering the above command sequence results in the following prompt:

```
[context_name]host_name(config-egtp-service)#
```

Syntax Description

gtpc { allow-on-congestion { apn-name *apn_name* **| arp** *priority_level* **} | bind { ipv4-address** *ipv4_address* **[ipv6-address** *ipv6_address* **] | ipv6-address** *ipv6_address* **[ipv4-address** *ipv4_address* **] } | command-messages { dual-ip-stack-support } | disable cause-source | echo-interval** *seconds* **[dynamic [smooth-factor** *multiplier* **]] echo-max-retransmissoins** *number* **| echo-retransmission-timeout** *seconds* **| error-response-handling | peer-salvation | ip qos-dscp {** *forwarding_type* **| max-remote-restart-counter-change** *integer* **} | max-retransmissions** *num* **| node-feature { cellular-iot network-triggered-service-restoration | pgw-restart-notification } | path-failure detection-policy { echo | control-restart-counter-change | echo-restart-counter-change } | private-extension overcharge-protection | reject s2b-ho-no-context | retransmission-timeout** *seconds* **| retransmission-timeout-ms** *milliseconds* **}**
no gtpc { allow-on-congestion { apn-name *apn_name* **| arp** *priority_level* **} | bind { ipv4-address** *ipv4_address* [ipv6-address *ipv6_address* **] | ipv6-address** *ipv6_address* **[ipv4-address** *ipv4_address*] **} | command-messages { dual-ip-stack-support } | disable cause-source | echo-interval |**

```
error-response-handling | node-feature {
cellular-iotnetwork-triggered-service-restoration |
pgw-restart-notification } | path-failure detection-policy |
private-extension overcharge-protection | reject s2b-ho-no-context }
default gtpc  disable cause-source |{ echo-interval |
echo-max-retransmissons | echo-retransmission-timeout disable cause-source|
 ip qos-dscp | max-retransmissions | node-feature { cellular-iot
network-triggered-service-restoration | pgw-restart-notification } |
path-failure detection-policy | retransmission-timeout |
retransmission-timeout-ms }
```

no

Disables or removes the configured GTP-C setting.

default

Resets the specified parameter to its default value.

allow-on-congestion { apn-name *apn_name* | arp *priority_level* }

Important ***P-GW, SAEGW, and S-GW only***. This functionality requires that a valid VoLTE license key be installed. Contact your Cisco account representative for information on how to obtain a license.

Enables the prioritized handling for calls under congestion conditions for the specified APN/ARP(s).

- If prioritized APN/ARP handling is enabled, and if the APN/ARP received in a CSReq at the EGTP demux matches any of the configured prioritized APN/ARP values, any valid CSReq will not be rejected at EGTP demux because of congestion control.
- This feature impacts only CSReq handling for new incoming calls.
- P-GW initiated dedicated bearer creation/updating is not changed due to this configuration.

apn-name *apn_name*: Configures the gateway to allow calls for this Access Point Name (APN), even under congestion. *apn_name* is an alphanumeric string of 1 through 64 characters. A maximum of 3 APNs can be configured.

arp *priority_level*: Configures the gateway to allow calls for this ARP, even under congestion. *priority_level* sets the priority value as an integer from 1 to 15. A maximum of 8 ARP values can be configured.

Important There is no APN-to-ARP mapping.

bind { ipv4-address *ipv4_address* [ipv6-address *ipv6_address*] | ipv6-address *ipv6_address* [ipv4-address *iv4p_address*] }

Binds the service to an interface with IPv4 address, IPv6 address, or both.

ipv4-address *ipv4_address* **[ipv6-address** *ipv6_address* **]**: Binds this service to the IPv4 address of a configured interface. Optionally, bind the service to a configured interface with an IPv6 address.

ipv4_address must be entered using IPv4 dotted-decimal notation.

ipv6_address must be entered using IPv6 colon-separated hexadecimal notation.

ipv6-address *ipv6_address* **[ipv4-address** *ipv4_address* **]**: Binds this service to the IPv6 address of a configured interface. Optionally, bind the service to a configured interface with an IPv4 address.

ipv6_address must be entered using IPv6 colon-separated hexadecimal notation.

ipv4_address must be entered using IPv4 dotted-decimal notation.

Important For binding GTP-C service on S2b interface, either IPv6 or IPv4 bind address shall be used. Binding both IPv4 and IPv6 address is not supported on ePDG.

The **ipv6-address** *ipv6_address* **[ipv4-address** *ipv4_address* **]** option is not currently supported on the SGSN.

cellular-iot

Enables the Cellular IoT features supported for eGTP Service.

command-messages dual-ip-stack-support

command-messages: Configuration related to MBC/DBC/BRC messages on S-GW and P-GW.

dual-ip-stack-support: Enables to handle command messages on both IPv4/IPv6 transport if supported. By default feature is enabled.

disable cause-source

disable: Disables functionality at eGTPC level.

cause-source: Disables cause source Bit in Cause IE.

echo-interval *seconds* [dynamic [smooth-factor *multiplier*]]

Configures the duration (in seconds) between the sending of echo request messages. *seconds* is an integer from 60 to 3600.

Default: 60

dynamic: Enables the dynamic echo timer for the eGTP service. The dynamic echo timer uses a calculated round trip timer (RTT) to support variances in different paths to peer nodes.

smooth-factor *multiplier*: Introduces a multiplier into the dynamic echo timer. *multiplier* is an integer from 1 to 5.

Default: 2

max-remote-restart-counter-change *integer*

Specifies the counter change after which the P-GW will detect a peer restart. Note that a peer restart will be detected only if the absolute difference between the new and old restart counters is less than the value configured. For example, if the **max-remote-restart-counter-change** is 10 and the current peer restart counter is 251, then eGTP will detect a peer restart only if the new restart counter is 252 through 255 or 0 through 5. Similarly, if the stored restart counter is 1, eGTP will detect a peer restart only if the new restart counter is 2 through 11.

Valid settings are from 1 to 255.

The recommended setting is 32.

The default setting is 255.

echo-max-retransmissions *number*

Configures the maximum retries for GTP Echo requests. *number* is an integer from 0 to 15. If **echo-max-retransmissions** option is not configured, then the **max-retransmissions** configuration will be used for maximum number of echo retries.

Default: 4

echo-retransmission-timeout *seconds*

Configures the echo retransmission timeout, in seconds, for the eGTP service. *seconds* is an integer ranging from 1 to 20.

If dynamic echo is enabled (**gtpc echo-interval dynamic**) the value set in this command serves as the dynamic minimum (if the RTT multiplied by the smooth factor is less than the value set in this command, the service uses this value).

Default: 3

error-response-handling

Enables error-response-handling on the S-GW. If this command is enabled in the eGTP service, then on receiving a bad response from the peer instead of dropping the message while doing validation eGTP-C informs the S-GW about the bad response received. The S-GW uses this notification from eGTP-C that a bad response is received to send a proper response to the other peer.

peer-salvation

Enables peer salvation for inactive GTPv2 peers for EGTP services in this context. When enabled, this functionality is enabled at the specific egtp-service level.

This functionality should be enabled at the context level if it is enabled at the egtp-service level. The configuration sequence is not dependent on enabling this functionality.

The parameter configured at the context level is used when peer-salvation is enabled. Ensure that peer-salvation is configured at all the configured services of a product. For example, sgw-services (egtp-service).

Note

- The parameter configured at the context level is used when peer-salvation is enabled. Ensure that peer-salvation is configured at all the configured services of a product. For example, sgw-services (egtp-service).
- All the information (peer statistics/recovery counter and so on) of the particular peer is lost after it is salvaged.
- The context level configuration is applied to egtpinmgr and egtpegmgr separately.

ip qos-dscp { *forwarding_type* }

Specifies the IP QoS DSCP per-hop behavior (PHB) to be marked on the outer header of signalling packets originating from the LTE component. This is a standards-based feature (RFC 2597 and RFC 2474).

Note that CS (class selector) mode options below are provided to support backward compatibility with the IP precedence field used by some network devides. CS maps one-to-one to IP precedence, where CS1 is IP precedence value 1. If a packet is received from a non-DSCP aware router that used IP precedence markings, then the DSCP router can still understand the encoding as a Class Selector code point.

The following forwarding types are supported:

- **af11**: Designates the use of Assured Forwarding 11 PHB.

  This is the default setting.
- **af12**: Designates the use of Assured Forwarding 12 PHB.
- **af13**: Designates the use of Assured Forwarding 13 PHB.
- **af21**: Designates the use of Assured Forwarding 21 PHB.
- **af22**: Designates the use of Assured Forwarding 22 PHB.
- **af23**: Designates the use of Assured Forwarding 23 PHB.
- **af31**: Designates the use of Assured Forwarding 31 PHB.
- **af32**: Designates the use of Assured Forwarding 32 PHB.
- **af33**: Designates the use of Assured Forwarding 33 PHB.
- **af41**: Designates the use of Assured Forwarding 41 PHB.
- **af42**: Designates the use of Assured Forwarding 42 PHB.
- **af43**: Designates the use of Assured Forwarding 43 PHB.
- **be**: Designates the use of Best Effort forwarding PHB.
- **cs1**: Designates the use of Class Selector code point "CS1".
- **cs2**: Designates the use of Class Selector code point "CS2".
- **cs3**: Designates the use of Class Selector code point "CS3".
- **cs4**: Designates the use of Class Selector code point "CS4".
- **cs5**: Designates the use of Class Selector code point "CS5".
- **cs6**: Designates the use of Class Selector code point "CS6".
- **cs7**: Designates the use of Class Selector code point "CS7".
- **ef**: Designates the use of Expedited Forwarding PHB typically dedicated to low-loss, low-latency traffic.

The assured forwarding behavior groups are listed in the table below.

	Class 1	Class 2	Class 3	Class 4
Low Drop	AF11	AF21	AF31	AF41

	Class 1	Class 2	Class 3	Class 4
Medium Drop	AF12	AF22	AF32	AF42
High Drop	AF13	AF23	AF33	AF43

Traffic marked with a higher class is given priority during congestion periods. If congestion occurs to traffic with the same class, the packets with the higher AF value are dropped first.

max-retransmissions *num*

Configures the maximum number of retries for packets as an integer from 0 through 15.

After maximum retransmissions is reached, the path is considered to be failed.

Default: 4

node-feature pgw-restart-notification

Enables P-GW Restart Notification functionality. Node will start announcement of new supported features to peer nodes in echo as soon as configuration is added.

From release 17.0 onwards, the S4-SGSN and MME support receiving/advertising the P-GW Restart Notification (PRN). This command option must be configured in order to inform S-GW that S4-SGSN and/or MME supports receiving/advertising the PRN in eGTPC echo request/response messages.

Default: Disabled

node-feature network-triggered-service-restoration

This keyword applies to MME and S-GW only.

Enables Network Triggered Service Restoration (NTSR) functionality as per 3GPP TS 23.007 Release 11 for this eGTP service.

Upon receipt of a Downlink Data Notification (DDN) message including an IMSI, the MME will accept the request and initiate paging including the IMSI in order to force the UE to re-attach. IMSI-based DDN requests contain a zero TEID. Since the UE is not attached, the UE will be paged over the whole MME coverage area.

A different MME may be selected by the eNodeB to service the attach request. Since the MME that serviced the DDN will not be aware that the UE has responded with the attach request, it will stop paging upon a timeout.

path-failure detection-policy echo

Enables session cleanup upon path failure detected via ECHO timeout toward a peer.

Default: Enabled

If disabled, there is no session cleanup upon path failure detected via ECHO timeout toward a peer; however, SNMP trap/logs will continue to indicate path failure.

path-failure detection-policy control-restart-counter-change

Enables path failure detection policy when the restart counter in Echo Request/Echo Response messages changes. Used in conjunction with the **max-remote-restart-counter-change** command.

path-failure detection-policy echo-restart-counter-change

Enables path failure detection policy when the restart counter in Control Request/Control Response messages changes. Used in conjunction with the **max-remote-restart-counter-change** command.

private-extension overcharge-protection

Important From StarOS 19.0 and later releases, this command is obsolete.

Important Use of Overcharging Protection requires that a valid license key be installed. Contact your Cisco account representative for information on how to obtain a license.

Controls whether the PDU will contain overcharge-protection related data in the Indication information element or in the private extension.

- If this keyword is enabled in the eGTP service, then eGTP-C will encode/decode overcharge-protection related data in/from the private extension instead of the Indication IE.
- If this option is disabled in the eGTP service, then the eGTP-C layer will encode/decode overcharge-protection related data in the Indication IE.
- By default, this option is disabled.

reject s2b-ho-no-context

Allows handoff call on S2b interface, even when eGTP-C does not have a UE context.

retransmission-timeout *seconds*

Important In 17.3 and later releases, this option has been deprecated. Use the **retransmission-timeout-ms** option.

Configures GTPv2 control packets (non-echo) retransmission timeout (in seconds) as an integer from 1 to 20.

Default: 5

retransmission-timeout-ms *milliseconds*

Configures the control packet retransmission timeout in GTP, in milliseconds <in steps of 100>, ranging from 1000 to 20000.

Default: 5000

Usage Guidelines

Use this command to configure GTP-C settings for the current service.

This interface assumes the characteristics of an S11 reference point on the S-GW or MME.

For communication between the S4-SGSN and LTE S-GW, the interface assumes the characteristics of an S4 reference point on the S4-SGSN. Before using the **gtpc** command on the S4-SGSN, a new or existing service must be created or entered using the **egtp-service** command in the *Context Configuration Mode*. Once the

eGTP service is configured, the service must be associated with the configured 2G and/or 3G services on the S4-SGSN using the **associate** command in the *SGSN Service Configuration Mode* and/or *GPRS Service Configuration Mode.*

Important If you modify this command, the parent service (service within which the eGTP/GTP-U service is configured) will automatically restart. Service restart results in dropping of active calls associated with the parent service.

Important For ePDG, IPv6 bind address must be used as ePDG supports IPv6 as transport on the S2b interface.

Example

The following command binds the service to a GTP-C interface with an IPv4 address of *112.104.215.177*:

```
gtpc bind ipv4-address 112.104.215.177
```

gtpp attribute

Allows the specification of the optional attributes to be present in the Call Detail Records (CDRs) that the GPRS/PDN/UMTS access gateway generates. It also defines that how the information is presented in CDRs by encoding the attribute field values.

Product

GGSN

SGSN

P-GW

SAEGW

Privilege

Security Administrator, Administrator

Command Modes

Exec > Global Configuration > Context Configuration

configure > **context** *context_name*

Entering the above command sequence results in the following prompt:

```
[context_name]host_name(config-ctx)#
```

Syntax Description

```
gtpp attribute { apn-ambr [ include-for-all-bearers |
include-for-default-bearer | include-for-non-gbr-bearers ] | apn-ni |
apn-selection-mode | charging-characteristic-selection-mode | camel-info
 | cell-plmn-id |  { ciot-cp-optind | ciot-unipdu-cponly } | diagnostics
 [ abnormal-release-cause ] | direct-tunnel | duration-ms | dynamic-flag
 | dynamic-flag-extension | furnish-charging-information | imei |
imsi-unauthenticated-flag | lapi last-ms-timezone | last-uli |
local-record-sequence-number | losdv | ms-timezone | msisdn | node-id |
```

```
node-id-suffix STRING | pdn-connection-id | pdp-address | pdp-type |
pgw-ipv6-addr | pgw-plmn-id | plmn-id | qos max-length | rat |
recordextension | record-extensions rat | record-type { sgsnpdprecord |
sgwrecord } | served-mnai | served-pdp-pdn-address-extension |
served-pdp-pdn-address-prefix-length | sgsn-change |  sms {
destination-number | recording-entity | service-centre } | sgw-ipv6-addr
 | sna-ipv6-addr | sponsor-id | start-time | stop-time | twanuli | uli |
 user-csg-information } +
default gtpp attribute { apn-ambr [ include-for-all-bearers |
include-for-default-bearer | include-for-non-gbr-bearers ] | apn-ni |
apn-selection-mode | charging-characteristic-selection-mode | camel-info
 | cell-plmn-id |  { ciot-cp-optind | ciot-unipdu-cponly } | diagnostics
 [ abnormal-release-cause ] | direct-tunnel | duration-ms | dynamic-flag
 | dynamic-flag-extension | furnish-charging-information | imei |
imsi-unauthenticated-flag | lapi last-ms-timezone | last-uli |
local-record-sequence-number | losdv | ms-timezone | msisdn | node-id |
node-id-suffix STRING | pdn-connection-id | pdp-address | pdp-type |
pgw-ipv6-addr | pgw-plmn-id | plmn-id | qos max-length | rat |
recordextension | record-extensions rat | record-type { sgsnpdprecord |
sgwrecord } | served-mnai | served-pdp-pdn-address-extension |
served-pdp-pdn-address-prefix-length | sgsn-change |  sms {
destination-number | recording-entity | service-centre } | sgw-ipv6-addr
 | sna-ipv6-addr | sponsor-id | start-time | stop-time | twanuli | uli |
 user-csg-information } +
no gtpp attribute { apn-ambr [ include-for-all-bearers |
include-for-default-bearer | include-for-non-gbr-bearers ] | apn-ni |
apn-selection-mode | charging-characteristic-selection-mode | camel-info
 | cell-plmn-id |  { ciot-cp-optind | ciot-unipdu-cponly } | diagnostics
 [ abnormal-release-cause ] | direct-tunnel | duration-ms | dynamic-flag
 | dynamic-flag-extension | furnish-charging-information | imei |
imsi-unauthenticated-flag | lapi last-ms-timezone | last-uli |
local-record-sequence-number | losdv | ms-timezone | msisdn | node-id |
node-id-suffix STRING | pdn-connection-id | pdp-address | pdp-type |
pgw-ipv6-addr | pgw-plmn-id | plmn-id | qos max-length | rat |
recordextension | record-extensions rat | record-type { sgsnpdprecord |
sgwrecord } | served-mnai | served-pdp-pdn-address-extension |
served-pdp-pdn-address-prefix-length | sgsn-change |  sms {
destination-number | recording-entity | service-centre } | sgw-ipv6-addr
 | sna-ipv6-addr | sponsor-id | start-time | stop-time | twanuli | uli |
 user-csg-information } +
```

default

Sets the default GTPP attributes in the generated CDRs. It also sets the default presentation of attribute values in generated CDRs.

no

Removes the configured GTPP attributes from the CDRs.

apn-ambr [include-for-all-bearers | include-for-default-bearer | include-for-non-gbr-bearers]

Default: Disabled

This keyword controls the inclusion of the optional field "apn-ambr" in the PGW-CDRs in the custom24 GTPP dictionary.

Important This keyword option will be available only if a valid license is installed. For more information, contact your Cisco account representative.

The APN Aggregate Maximum Bit Rate (AMBR) is a subscription parameter stored per APN. It limits the aggregate bit rate that can be expected to be provided across all non-GBR bearers and across all PDN connections of the same APN. Each of these non-GBR bearers potentially utilize the entire APN AMBR, e.g. when the other non-GBR bearers do not carry any traffic. The APN AMBR is present as part of QoS information.

In 15.0 and later releases, this CLI command should be configured along with the following additional options to support APN-AMBR reporting in SGW-CDRs in all GTPP dictionaries.

- **include-for-all-bearers**: Includes the APN-AMBR information in SGW-CDRs for all bearers (GBR and NON-GBR)
- **include-for-default-bearer**: Includes APN-AMBR information in SGW-CDRs only for default bearer.
- **include-for-non-gbr-bearers**: Includes APN-AMBR information for non-gbr-bearers.

This feature is required to enable post-processing of CDRs to verify MVNO subscribers actual QoS against invoicing systems.

Important This CLI command and the associated options are not available for products other than S-GW and P-GW. The option "**non-gbr-bearers-only**" is available in S-GW and P-GW but the other options are available in S-GW only.

In the P-GW implementation, if the CLI command "**gtpp attribute apn-ambr**" is configured, it will be treated as "**gtpp attribute apn-ambr non-gbr-bearers-only**". In case of S-GW/P-GW combo if any of the options is configured, it will be considered that the attribute is available.

apn-ni

Default: Enabled

This keyword controls the inclusion of the optional field "APN" in the x-CDRs.

apn-selection-mode

Default: Enabled

This keyword controls the inclusion of the optional field "APN Selection Mode" in the x-CDRs.

camel-info

SGSN only

Enter this keyword to include CAMEL-specific fields in SGSN CDRs. Default: Disabled

cell-plmn-id

SGSN only

Enter this keyword to enable the system to include the Cell PLMN ID field in the M-CDR. Default: Disabled

charging-characteristic-selection-mode

Default: Enabled

This keyword controls the inclusion of the optional field "Charging Characteristic Selection Mode" in the x-CDRs.

ciot-cp-optind

Includes optional field "CP CIoT EPS optimisation indicator" in the CDR.

ciot-unipdu-cponly

Includes optional field "UNI PDU CP Only Flag" in the CDR.

diagnostics [abnormal-release-cause]

Default: Disabled

Enables the system to include the Diagnostic field in the CDR that is created when PDP contexts are released. The field will include one of the following values:

- **26** - For GGSN: if the GGSN sends "delete PDP context request" for any other reason (e.g., the operator types "clear subscribers" on the GGSN). For SGSN: The SGSN includes this cause code in the S-CDR to indicate that a secondary PDP context activation request or a PDP context modification request has been rejected due to insufficient resources.
- **36** - For GGSN: this cause code is sent in the G-CDR to indicate the PDP context has been deactivated in the GGSN due to the SGSN having sent a "delete PDP context request" to the GGSN. For SGSN, this cause code is used to indicate a regular MS or network-initiated PDP context deactivation.
- **37** - when the network initiates a QoS modification, the SGSN sends in the S-CDR to indicate that the MS initiation deactivate request message has been rejected with QoS not accepted as the cause.
- **38** - if the GGSN sends "delete PDP context request" due to GTP-C/GTP-U echo timeout with SGSN. If the SGSN sends this cause code, it indicates PDP context has been deactivated due to path failure, specifically GTP-C/GTP-U echo timeout.
- **39** - SGSN only - this code indicates the network (GGSN) has requested a PDP context reactivation after a GGSN restart.
- **40** - if the GGSN sends "delete PDP context request" due to receiving a RADIUS Disconnect-Request message.

abnormal-release-cause: This keyword controls the inclusion of abnormal bearer termination information in diagnostics field of SGW-CDR. Note that the CLI command "**gtpp attribute diagnostics**" will disable **abnormal-release-cause** and enable the **diagnostics** field. The **no gtpp attribute diagnostics** command will disable both **abnormal-release-cause** and **diagnostics** field.

Important The Abnormal Bearer Termination feature is currently applicable only to custom34 and custom35 GTPP dictionaries. That is, the bearer termination cause is populated in SGW-CDR for custom34 and custom35 dictionaries, and PGW-CDRs for custom35 GTPP dictionary when the cause for record closing is "Abnormal Release".

direct-tunnel

Default: Disabled

Includes the Direct Tunnel field in PGW-CDR/eG-CDRs.

This keyword is applicable for GGSN, P-GW and S-GW only.

duration-ms

Specifies that the information contained in the mandatory Duration field be reported in milliseconds instead of seconds (as the standards require). Default: Disabled

dynamic-flag

Default: Enabled

This keyword controls the inclusion of the optional field "Dynamic Flag" in the x-CDRs.

dynamic-flag-extension

Default: Enabled

This keyword controls the inclusion of the optional field "Dynamic Address Flag Extension" in the x-CDRs.

This field is seen in the CDR when the IPv4 address is dynamically assigned for a dual PDP context. This extension field is required in the 3GPP Release 10 compliant CDRs so that the Dual Stack Bearer support is available.

furnish-charging-information

Default: Disabled

This keyword controls the inclusion of the optional field "pSFurnishChargingInformation" in the eG-CDRs and PGW-CDRs.

Important The Furnish Charging Information (FCI) feature is applicable to all GTPP dictionaries compliant to 3GPP Rel.7 and 3GPP Rel.8 except custom43 dictionary. This keyword option will be available only if a valid license is installed. For more information, contact your Cisco account representative.

PGW-CDR and eG-CDR will contain FCI only if it is enabled at command level, i.e. using the **gtpp attribute furnish-charging-information** command in GTPP Server Group Configuration mode.

Whenever FCI changes, a new Free-Format-Data (FFD) value is either appended to existing FFD or overwritten on the existing FDD depending on Append-Free-Format-Data (AFFD) flag. CDR is not generated upon FCI change.

FCI is supported in main CDR as well as in LOSDV. Whenever a trigger (volume, time, RAT, etc.) happens current available FFD at command level is added to the main body of the CDR. The same FFD at command level is added to the main body of the next CDRs until it is not appended or overwritten by next Credit-Control-Answer message at command level.

In the case of custom43 dictionary, the FCI implementation will be as follows:

- Whenever FCI changes PGW-CDR will generate CDR i.e close old bucket and will have old FCI details in the generated CDR.
- Translation for the PS-Free-Format-Data in CDR will be conversion of hexadecimal values in ASCII format (for numbers 0 to 9) to decimal values as integers.
- PS-Append-Free-Format-Data always OVERWRITE.

imei

Default: Disabled

For SGSN: includes the IMEI value in the S-CDR.

For GGSN: includes the IMEISV value in the G-CDR.

imsi-unauthenticated-flag

Default: Enabled

This keyword controls the inclusion of the optional field "IMSI Unauthenticated Flag" in the x-CDRs.

When the served IMSI is not authenticated, this field "IMSI Unauthenticated Flag" if configured, will be present in the P-GW CDR record for custom35 dictionary. This field is added per 3GPP TS 32.298 v10.7.

lapi

Default: Disabled

Includes the Low Access Priority Indicator (LAPI) field in the CDRs. This field is required to support MTC feature.

When UE indicates low priority connection, then the "lowPriorityIndicator" attribute will be included in the CDR.

last-ms-timezone

Default: Disabled

Sets the "Last MS-Timezone" in the CDR field. This option would be disabled when the default option is used.

last-uli

Default: Disabled

Sets the "Last ULI" in the CDR field. This option would be disabled when the default option is used.

local-record-sequence-number

Default: Disabled

This keyword provides both the local record sequence number and the Node ID. In the x-CDRs, this field indicates the number of CDRs generated by the node and is unique within the session manager.

The Node ID field is included in the x-CDR for any of several reasons, such as when PDP contexts are released or if partial-CDR is generated based on configuration. The field will consist of a AAA Manager identifier automatically appended to the name of the SGSN or GGSN service.

The name of the SGSN or GGSN service may be truncated, because the maximum length of the Node ID field is 20 bytes. Since each AAA Manager generates CDRs independently, this allows the Local Record Sequence Number and Node ID fields to uniquely identify a CDR.

Important If the **gtpp single-source centralized-lrsn** is configured, the 'Node-ID' field consists of only the specified NodeID-suffix. If NodeID-suffix is not configured, GTPP group name is used. For default GTPP groups, GTPP context-name is used. If the **gtpp single-source centralized-lrsn** is not configured, then node-id format for CDRs generated by Sessmgr is as follows: <1-byte Sessmgr restartvalue><3-byte Sessmgr instance number> <node-id-suffix>. If the **gtpp single-source centralized-lrsn** is not configured, then node-id format for CDRs generated by ACSmgr is as follows: <1-byte ACSmgr restart-value> <3-byte ACSmgr instance number> <Active charging service-name>.

losdv

Default: Enabled

This keyword controls the inclusion of the optional field "List of Service Data" in the x-CDRs.

ms-timezone

Default: Enabled

This keyword controls the inclusion of the optional field "MS-Timezone" in the x-CDRs.

msisdn

Default: Enabled

This keyword controls the inclusion of the optional field "MSISDN" in the x-CDRs.

node-id

Default: Enabled

This keyword controls the inclusion of the optional field "Node ID" in the x-CDRs.

node-id-suffix *STRING*

Default: Disabled

Specifies the configured Node-ID-Suffix to use in the NodeID field of GTPP CDRs as an alphanumeric string of 1 through 16 characters. Each Session Manager task generates a unique NodeID string per GTPP context.

Important The NodeID field is a printable string of the *ndddSTRING* format: *n*: The first digit is the Sessmgr restart counter having a value between 0 and 7. *ddd*: The number of sessmgr instances. Uses the specified NodeID-suffix in all CDRs. The "Node-ID" field consists of sessMgr Recovery counter (1 digit) *n* + AAA Manager identifier (3 digits) *ddd* + the configured Node-Id-suffix (1 to 16 characters) *STRING*. If the centralized LRSN feature is enabled, the "Node-ID" field will consist of only the specified NodeID-suffix (NodeID-prefix is not included). If this option is not configured, then GTPP group name will be used instead (For default GTPP groups, context-name will be used).

Important If this **node-id-suffix** is not configured, the GGSN uses the GTPP context name as the Node-id-suffix (truncated to 16 characters) and the SGSN uses the GTPP group named as the node-id-suffix.

pdn-connection-id

Default: Enabled

This keyword controls the inclusion of the optional field "PDN Connection ID" in the x-CDRs.

pdp-address

Default: Enabled

This keyword controls the inclusion of the optional field "PDP Address" in the x-CDRs.

pdp-type

Default: Enabled

This keyword controls the inclusion of the optional field "PDP Type" in the x-CDRs.

pgw-ipv6-addr

Default: Disabled

Specifying this option allows to configure the P-GW IPv6 address.

Important This attribute can be controllably configured in custom24 and custom35 SGW-CDR dictionaries.

pgw-plmn-id

Default: Enabled

This keyword controls the inclusion of the optional field "PGW PLMN-ID" in the x-CDRs.

plmn-id [unknown-use]

Default: Enabled

For SGSN, reports the SGSN PLMN Identifier value (the RAI) in the S-CDR provided if the dictionary supports it.

For GGSN, reports the SGSN PLMN Identifier value (the RAI) in the G-CDR if it was originally provided by the SGSN in the GTP create PDP context request. It is omitted if the SGSN does not supply one.

Normally when SGSN PLMN-id information is not available, the attribute sgsnPLMNIdentifier is not included in the CDR. This keyword enables the inclusion of the sgsnPLMNIdentifier with a specific value when the SGSN PLMN-id is not available.

unknown-use *hex_num*: is aa hexadecimal number from 0x0 through 0xFFFFFF that identifies a foreign SGSN that has not provided a PLMN-id. For GGSN only.

qos max-length

Default: Disabled

Specifying this option will change the parameters related to QoS sent in S-CDR and SaMOG CDR. The **max-length** option is used to modify the length of QoS sent in CDR. The **qos_value** must be an integer from 4 through 24.

This feature is introduced to support Rel.7+ QoS formats.

rat

Default: Enabled

For SGSN: includes the RAT (identifies the radio access technology type) value in the S-CDR.

For GGSN: includes the RAT (identifies the radio access technology type) value in the G-CDR.

recordextension

Default: Disabled

This keyword controls the inclusion of the optional field "RecordExtension" in the x-CDRs.

record-extensions rat

Default: Disabled

Enables network operators and/or manufacturers to add their own recommended extensions to the CDRs according to the standard record definitions from 3GPP TS 32.298 Release 7 or higher.

record-type { sgsnpdprecord | sgwrecord }

Important This keyword is available only when the SaMOG Mixed Mode license (supporting both 3G and 4G) is configured.

Default: sgwrecord

Specifies the SaMOG CDR type to use.

For an SaMOG 3G license, this keyword will not be available. However, sgsnpdprecord type will be used as the default record type.

served-mnai

Default: Disabled

This keyword controls the inclusion of the optional field "Served MNAI" in the x-CDRs.

served-pdp-pdn-address-extension

Default: Disabled

In support of IPv4v6 dual-stack PDP address types, this keyword causes the service to include IPv4v6 address information in the CDR. The IPv4 address goes in the Served PDP PDN Address Extension field and the IPv6 address goes in the Served PDP Address or Served PDP PDN Address field.

Important This attribute will not be displayed if the GTPP dictionary is set to custom34.

Note For SGSN, on enabling **served-pdp-pdn-address-extension** all custom S-CDR dictionaries will support the CDR field "Served PDP/ PDN Address extension" except for the following dictionaries:

- custom17
- custom18
- custom23
- custom42
- custom41

served-pdp-pdn-address-prefix-length

Default: Enabled

In support of IPv6 prefix delegation, this keyword causes the service to include this field "Served PDP PDN Address" in the x-CDRs.

If this field is configured, the servedPDPPDNAddress field will support reporting the IPv6 prefix length as outlined in 3GPP 32.298. The prefix length will only be reported if:

- it is configured
- it is not the default length of 64
- it is an IPv6 or IPv4v6 call

sgsn-change

Default: Enabled

This keyword is specific to SGSN and is license restricted.

This keyword controls the inclusion of the S-CDR attribute "SGSN Change" in the S-CDRs. It is enabled by default and the attribute "SGSN Change" is included in the S-CDRs by default.

Note For SGSN specific custom33 dictionary, it is recommended to disable this keyword before an upgrade to prevent billing issues.

sgw-ipv6-addr

Default: Disabled

Specifying this option allows to configure the S-GW IPv6 address.

Important This attribute can be controllably configured in custom24 and custom35 SGW-CDR dictionaries.

sms { destination-number | recording-entity | service-centre }

This keyword is specific to the SGSN.

Entering this keyword causes the inclusion of an SMS-related field in the SMS-MO-CDR or SMS-MT-CDR.

destination-number: Includes the "destinationNumber" field in the SMS-MO-CDR or SMS-MT-CDR.

recording-entity: Includes the "recordingEntity" field in the SMS-MO-CDR or SMS-MT-CDR.

service-centre: Includes the "serviceCentre" field in the SMS-MO-CDR or SMS-MT-CDR.

sna-ipv6-addr

Default: Disabled

Specifying this option allows to configure the Serving Node IPv6 Address (SNAv6).

Important This attribute can be controllably configured in custom24 and custom35 SGW-CDR dictionaries.

sponsor-id

Default: Disabled

Includes the Sponsor ID and Application-Service-Provider-Identity fields in PGW-CDR.

Note that the "Sponsor ID" and "Application-Service-Provider-Identity" attributes will be included in PGW-CDR if the PCEF supports Sponsored Data Connectivity feature or the required reporting level is sponsored connectivity level as described in 3GPP TS 29.212.

This feature is implemented to be in compliance with Release 11 3GPP specification for CDRs. So, this behavior is applicable to all GTPP dictionaries that are Release 11 compliant, i.e. custom35.

start-time

Default: Enabled

This keyword controls the inclusion of the optional field "Start-Time" in the x-CDRs.

stop-time

Default: Enabled

This keyword controls the inclusion of the optional field "Stop-Time" in the x-CDRs.

twanuli

Default: Disabled

This keyword controls the inclusion of the optional field "TWAN User Location Information" in the CDRs.

uli

Default: Enabled

This keyword controls the inclusion of the optional field "User Location Information" in the x-CDRs.

user-csg-information

Default: Disabled

This keyword controls the inclusion of the optional field "User CSG Information" in the x-CDRs.

Important

Currently, UCI values are only supported for SGW-CDRs.

This attribute will not be displayed if the GTPP dictionary is set to custom11, custom34, or custom35.

+

Indicates that this command can be entered multiple times to configure multiple attributes.

Usage Guidelines

Use this command to configure the type of optional information fields to include in generated CDRs (M-CDRs, S-CDRs, S-SMO-CDR, S-SMT-CDR from SGSN and G-CDRs, eG-CDRs from GGSN) by the AGW (SGSN/GGSN/P-GW/SAEGW). In addition, it controls how the information for some of the mandatory fields are reported.

Fields described as optional by the standards but not listed above will always be present in the CDRs, except for Record Extensions (which will never be present).

Important

This command can be repeated multiple times with different keywords to configure multiple GTPP attributes.

Example

The following command configures the system to present the time provided in the Duration field of the CDR is reported in milliseconds:

```
gtpp attribute duration-ms
```

gtpp attribute

Enables the specification of some of the optional fields in the CDRs that the GSN (GGSN or SGSN) generates and/or how the information is to be presented. Many keywords are also applicable to S-GW and P-GW CDRs.

Product

GGSN

P-GW

SAEGW

SaMOG

SGSN

S-GW

Privilege

Security Administrator, Administrator

Command Modes

Exec > Global Configuration > Context Configuration > GTPP Server Group Configuration

configure > **context** *context_name* > **gtpp group** *group_name*

Entering the above command sequence results in the following prompt:

```
[context_name]host_name(config-gtpp-group)#
```

Syntax Description

```
gtpp attribute { apn-ambr [ include-for-all-bearers |
include-for-default-bearer | include-for-non-gbr-bearers ] | apn-ni |
apn-selection-mode | charging-characteristic-selection-mode | camel-info
 | cell-plmn-id |  { ciot-cp-optind | ciot-unipdu-cponly } | diagnostics[
 abnormal-release-cause ] | direct-tunnel | duration-ms | dynamic-flag |
 dynamic-flag-extension | extended-bitrate | furnish-charging-information
 | imei | imsi-unauthenticated-flag | lapi last-ms-timezone | last-uli |

| local-record-sequence-number | losdv | ms-timezone | msisdn | node-id
| node-id-suffix STRING packet-count | pco-nai | pdn-connection-id |
pdp-address | pdp-type | pgw-ipv6-addr | pgw-plmn-id | plmn-id | qos
max-length | rat | recordextension | record-extensions rat | record-type
 { sgsnpdprecord | sgwrecord } | served-mnai |
served-pdp-pdn-address-extension | served-pdp-pdn-address-prefix-length
| sgsn-change |  sms { destination-number | recording-entity |
service-centre } | sgw-ipv6-addr | sna-ipv6-addr | sponsor-id | start-time
 | stop-time | twanuli | ue-tun-ip-port | uwanuli | uli |
user-csg-information } +
default gtpp attribute { apn-ambr [ include-for-all-bearers |
include-for-default-bearer | include-for-non-gbr-bearers ] | apn-ni |
apn-selection-mode | charging-characteristic-selection-mode | camel-info
 | cell-plmn-id |  { ciot-cp-optind | ciot-unipdu-cponly } | diagnostics[
 abnormal-release-cause ] | direct-tunnel | duration-ms | dynamic-flag |
 dynamic-flag-extension | furnish-charging-information | imei |
imsi-unauthenticated-flag | lapi last-ms-timezone | last-uli |
| local-record-sequence-number | losdv | ms-timezone | msisdn | node-id
| node-id-suffix STRING | pdn-connection-id | pdp-address | pdp-type |
pgw-ipv6-addr | pgw-plmn-id | plmn-id | qos max-length | rat |
recordextension | record-extensions rat | record-type { sgsnpdprecord |
sgwrecord } | served-mnai | served-pdp-pdn-address-extension |
served-pdp-pdn-address-prefix-length | sgsn-change |  sms {
destination-number | recording-entity | service-centre } | sgw-ipv6-addr
 | sna-ipv6-addr | sponsor-id | start-time | stop-time | twanuli | uwanuli
 | uli | user-csg-information } +
no gtpp attribute { apn-ambr [ include-for-all-bearers |
```

```
include-for-default-bearer | include-for-non-gbr-bearers ] | apn-ni |
apn-selection-mode | charging-characteristic-selection-mode | camel-info
 | cell-plmn-id |  { ciot-cp-optind | ciot-unipdu-cponly } | diagnostics[
 abnormal-release-cause ] | direct-tunnel | duration-ms | dynamic-flag |
 dynamic-flag-extension | extended-bitrate | furnish-charging-information
 | imei | imsi-unauthenticated-flag | lapi last-ms-timezone | last-uli |

| local-record-sequence-number | losdv | ms-timezone | msisdn | node-id
| node-id-suffix STRING packet-count | pco-nai | pdn-connection-id |
pdp-address | pdp-type | pgw-ipv6-addr | pgw-plmn-id | plmn-id | qos
max-length | rat | recordextension | record-extensions rat | record-type
 { sgsnpdprecord | sgwrecord } | served-mnai |
served-pdp-pdn-address-extension | served-pdp-pdn-address-prefix-length
| af-record-info  | sgsn-change |  sms { destination-number |
recording-entity | service-centre } | sgw-ipv6-addr | sna-ipv6-addr |
sponsor-id | start-time | stop-time | twanuli | ue-tun-ip-port | uwanuli
 | uli | user-csg-information } +
```

default

Resets the default attribute values for this GTPP group configuration.

no

Disables the specified optional field so that the information will not be present in generated CDRs.

apn-ambr [include-for-all-bearers | include-for-default-bearer | include-for-non-gbr-bearers]

Default: Disabled

This keyword controls the inclusion of the optional field "apn-ambr" in the PGW-CDRs in the custom24 GTPP dictionary.

Important This keyword option will be available only if a valid license is installed. For more information, contact your Cisco account representative.

The APN Aggregate Maximum Bit Rate (AMBR) is a subscription parameter stored per APN. It limits the aggregate bit rate that can be expected to be provided across all non-GBR bearers and across all PDN connections of the same APN. Each of these non-GBR bearers potentially utilize the entire APN AMBR, e.g. when the other non-GBR bearers do not carry any traffic. The APN AMBR is present as part of QoS information.

In 15.0 and later releases, this CLI command should be configured along with the following additional options to support APN-AMBR reporting in SGW-CDRs in all GTPP dictionaries.

- **include-for-all-bearers**: Includes the APN-AMBR information in SGW-CDRs for all bearers (GBR and NON-GBR)
- **include-for-default-bearer**: Includes APN-AMBR information in SGW-CDRs only for default bearer.
- **include-for-non-gbr-bearers**: Includes APN-AMBR information for non-gbr-bearers.

This feature is required to enable post-processing of CDRs to verify MVNO subscribers actual QoS against invoicing systems.

Important This CLI command and the associated options are not available for products other than S-GW and P-GW. The option "**non-gbr-bearers-only**" is available in S-GW and P-GW but the other options are available in S-GW only.

In the P-GW implementation, if the CLI command "**gtpp attribute apn-ambr**" is configured, it will be treated as "**gtpp attribute apn-ambr non-gbr-bearers-only**". In case of S-GW/P-GW combo if any of the options is configured, it will be considered that the attribute is available.

apn-ni

Default: Enabled

This keyword controls the inclusion of the optional field "APN" in the x-CDRs.

apn-selection-mode

Default: Enabled

This keyword controls the inclusion of the optional field "APN Selection Mode" in the x-CDRs.

camel-info

SGSN only

Enter this keyword to include CAMEL-specific fields in SGSN CDRs. Default: Disabled

cell-plmn-id

SGSN only

Enter this keyword to enable the system to include the Cell PLMN ID field in the M-CDR. Default: Disabled

charging-characteristic-selection-mode

Default: Enabled

This keyword controls the inclusion of the optional field "Charging Characteristic Selection Mode" in the x-CDRs.

ciot-cp-optind

Includes optional field "CP CIoT EPS optimisation indicator" in the CDR.

ciot-unipdu-cponly

Includes optional field "UNI PDU CP Only Flag" in the CDR.

diagnostics [abnormal-release-cause]

Default: Disabled

Enables the system to include the Diagnostic field in the CDR that is created when PDP contexts are released. The field will include one of the following values:

- **26** - For GGSN: if the GGSN sends "delete PDP context request" for any other reason (e.g., the operator types "clear subscribers" on the GGSN). For SGSN: The SGSN includes this cause code in the S-CDR to indicate that a secondary PDP context activation request or a PDP context modification request has been rejected due to insufficient resources.
- **36** - For GGSN: this cause code is sent in the G-CDR to indicate the PDP context has been deactivated in the GGSN due to the SGSN having sent a "delete PDP context request" to the GGSN. For SGSN, this cause code is used to indicate a regular MS or network-initiated PDP context deactivation.
- **37** - when the network initiates a QoS modification, the SGSN sends in the S-CDR to indicate that the MS initiation deactivate request message has been rejected with QoS not accepted as the cause.
- **38** - if the GGSN sends "delete PDP context request" due to GTP-C/GTP-U echo timeout with SGSN. If the SGSN sends this cause code, it indicates PDP context has been deactivated due to path failure, specifically GTP-C/GTP-U echo timeout.
- **39** - SGSN only - this code indicates the network (GGSN) has requested a PDP context reactivation after a GGSN restart.
- **40** - if the GGSN sends "delete PDP context request" due to receiving a RADIUS Disconnect-Request message.

abnormal-release-cause: This keyword controls the inclusion of abnormal bearer termination information in diagnostics field of SGW-CDR. Note that the CLI command "**gtpp attribute diagnostics**" will disable **abnormal-release-cause** and enable the **diagnostics** field. The **no gtpp attribute diagnostics** command will disable both **abnormal-release-cause** and **diagnostics** field.

Important The Abnormal Bearer Termination feature is currently applicable only to custom34 and custom35 GTPP dictionaries. That is, the bearer termination cause is populated in SGW-CDR for custom34 and custom35 dictionaries, and PGW-CDRs for custom35 GTPP dictionary when the cause for record closing is "Abnormal Release".

direct-tunnel

Default: Disabled

Includes the Direct Tunnel field in PGW-CDR/eG-CDRs.

This keyword is applicable for GGSN, P-GW and S-GW only.

duration-ms

Specifies that the information contained in the mandatory Duration field be reported in milliseconds instead of seconds (as the standards require). Default: Disabled

dynamic-flag

Default: Enabled

This keyword controls the inclusion of the optional field "Dynamic Flag" in the x-CDRs.

dynamic-flag-extension

Default: Enabled

This keyword controls the inclusion of the optional field "Dynamic Address Flag Extension" in the x-CDRs.

This field is seen in the CDR when the IPv4 address is dynamically assigned for a dual PDP context. This extension field is required in the 3GPP Release 10 compliant CDRs so that the Dual Stack Bearer support is available.

extended-bitrate

Default: Disabled

This keyword controls the inclusion of extended bit-rate information in P-GW CDRs when the APN-AMBR, MBR, or GBR is greater than 4.2 Gbps.

furnish-charging-information

Default: Disabled

This keyword controls the inclusion of the optional field "pSFurnishChargingInformation" in the eG-CDRs and PGW-CDRs.

Important The Furnish Charging Information (FCI) feature is applicable to all GTPP dictionaries compliant to 3GPP Rel.7 and 3GPP Rel.8 except custom43 dictionary. This keyword option will be available only if a valid license is installed. For more information, contact your Cisco account representative.

PGW-CDR and eG-CDR will contain FCI only if it is enabled at command level, i.e. using the **gtpp attribute furnish-charging-information** command in GTPP Server Group Configuration mode.

Whenever FCI changes, a new Free-Format-Data (FFD) value is either appended to existing FFD or overwritten on the existing FDD depending on Append-Free-Format-Data (AFFD) flag. CDR is not generated upon FCI change.

FCI is supported in main CDR as well as in LOSDV. Whenever a trigger (volume, time, RAT, etc.) happens current available FFD at command level is added to the main body of the CDR. The same FFD at command level is added to the main body of the next CDRs until it is not appended or overwritten by next Credit-Control-Answer message at command level.

In the case of custom43 dictionary, the FCI implementation will be as follows:

- Whenever FCI changes PGW-CDR will generate CDR i.e close old bucket and will have old FCI details in the generated CDR.
- Translation for the PS-Free-Format-Data in CDR will be conversion of hexadecimal values in ASCII format (for numbers 0 to 9) to decimal values as integers.
- PS-Append-Free-Format-Data always OVERWRITE.

imei

Default: Disabled

For SGSN: includes the IMEI value in the S-CDR.

For GGSN: includes the IMEISV value in the G-CDR.

imsi-unauthenticated-flag

Default: Enabled

This keyword controls the inclusion of the optional field "IMSI Unauthenticated Flag" in the x-CDRs.

When the served IMSI is not authenticated, this field "IMSI Unauthenticated Flag" if configured, will be present in the P-GW CDR record for custom35 dictionary. This field is added per 3GPP TS 32.298 v10.7.

lapi

Default: Disabled

Includes the Low Access Priority Indicator (LAPI) field in the CDRs. This field is required to support MTC feature.

When UE indicates low priority connection, then the "lowPriorityIndicator" attribute will be included in the CDR.

last-ms-timezone

Sets the "Last MS-Timezone" in the CDR field. This option would be disabled when the default option is used.

last-uli

Sets the "Last ULI" in the CDR field. This option would be disabled when the default option is used.

local-record-sequence-number

Default: Disabled

This keyword provides both the local record sequence number and the Node ID. In the x-CDRs, this field indicates the number of CDRs generated by the node and is unique within the session manager.

The Node ID field is included in the x-CDR for any of several reasons, such as when PDP contexts are released or if partial-CDR is generated based on configuration. The field will consist of a AAA Manager identifier automatically appended to the name of the SGSN or GGSN service.

The name of the SGSN or GGSN service may be truncated, because the maximum length of the Node ID field is 20 bytes. Since each AAA Manager generates CDRs independently, this allows the Local Record Sequence Number and Node ID fields to uniquely identify a CDR.

Important If the **gtpp single-source centralized-lrsn** is configured, the 'Node-ID' field consists of only the specified NodeID-suffix. If NodeID-suffix is not configured, GTPP group name is used. For default GTPP groups, GTPP context-name is used. If the **gtpp single-source centralized-lrsn** is not configured, then node-id format for CDRs generated by Sessmgr is as follows: <1-byte Sessmgr restartvalue><3-byte Sessmgr instance number> <node-id-suffix>. If the **gtpp single-source centralized-lrsn** is not configured, then node-id format for CDRs generated by ACSmgr is as follows: <1-byte ACSmgr restart-value> <3-byte ACSmgr instance number> <Active charging service-name>.

losdv

Default: Enabled

This keyword controls the inclusion of the optional field "List of Service Data" in the x-CDRs.

ms-timezone

Default: Enabled

This keyword controls the inclusion of the optional field "MS-Timezone" in the x-CDRs.

msisdn

Default: Enabled

This keyword controls the inclusion of the optional field "MSISDN" in the x-CDRs.

node-id

Default: Enabled

This keyword controls the inclusion of the optional field "Node ID" in the x-CDRs.

node-id-suffix *STRING*

Default: Disabled

Specifies the configured Node-ID-Suffix to use in the NodeID field of GTPP CDRs as an alphanumeric string of 1 through 16 characters. Each Session Manager task generates a unique NodeID string per GTPP context.

Important The NodeID field is a printable string of the *ndddSTRING* format: *n*: The first digit is the Sessmgr restart counter having a value between 0 and 7. *ddd*: The number of sessmgr instances. Uses the specified NodeID-suffix in all CDRs. The "Node-ID" field consists of sessMgr Recovery counter (1 digit) *n* + AAA Manager identifier (3 digits) *ddd* + the configured Node-Id-suffix (1 to 16 characters) *STRING*. If the centralized LRSN feature is enabled, the "Node-ID" field will consist of only the specified NodeID-suffix (NodeID-prefix is not included). If this option is not configured, then GTPP group name will be used instead (For default GTPP groups, context-name will be used).

Important If this **node-id-suffix** is not configured, the GGSN uses the GTPP context name as the Node-id-suffix (truncated to 16 characters) and the SGSN uses the GTPP group named as the node-id-suffix.

packet-count

Default: Disabled

Specifying this option includes the optional field "datapacketFBCUplink" and "datapacketFBCDownlink" in the CDR.

Important This keyword is applicable to custom24 GTPP dictionary.

pco-nai

Specifying this option includes optional field "PCO- Network Access Identifier" in the P-GW CDR.

Important This keyword is applicable to custom44 GTPP dictionary.

pdn-connection-id

Default: Enabled

This keyword controls the inclusion of the optional field "PDN Connection ID" in the x-CDRs.

pdp-address

Default: Enabled

This keyword controls the inclusion of the optional field "PDP Address" in the x-CDRs.

pdp-type

Default: Enabled

This keyword controls the inclusion of the optional field "PDP Type" in the x-CDRs.

pgw-ipv6-addr

Default: Disabled

Specifying this option allows to configure the P-GW IPv6 address.

Important This attribute can be controllably configured in custom24 and custom35 SGW-CDR dictionaries.

pgw-plmn-id

Default: Enabled

This keyword controls the inclusion of the optional field "PGW PLMN-ID" in the x-CDRs.

plmn-id [unknown-use]

Default: Enabled

For SGSN, reports the SGSN PLMN Identifier value (the RAI) in the S-CDR provided if the dictionary supports it.

For GGSN, reports the SGSN PLMN Identifier value (the RAI) in the G-CDR if it was originally provided by the SGSN in the GTP create PDP context request. It is omitted if the SGSN does not supply one.

Normally when SGSN PLMN-id information is not available, the attribute sgsnPLMNIdentifier is not included in the CDR. This keyword enables the inclusion of the sgsnPLMNIdentifier with a specific value when the SGSN PLMN-id is not available.

unknown-use *hex_num*: is aa hexadecimal number from 0x0 through 0xFFFFFF that identifies a foreign SGSN that has not provided a PLMN-id. For GGSN only.

qos max-length

Default: Disabled

Specifying this option will change the parameters related to QoS sent in S-CDR and SaMOG CDR. The **max-length** option is used to modify the length of QoS sent in CDR. The **qos_value** must be an integer from 4 through 24.

This feature is introduced to support Rel.7+ QoS formats.

rat

Default: Enabled

For SGSN: includes the RAT (identifies the radio access technology type) value in the S-CDR.

For GGSN: includes the RAT (identifies the radio access technology type) value in the G-CDR.

recordextension

Default: Disabled

This keyword controls the inclusion of the optional field "RecordExtension" in the x-CDRs.

record-extensions rat

Default: Disabled

Enables network operators and/or manufacturers to add their own recommended extensions to the CDRs according to the standard record definitions from 3GPP TS 32.298 Release 7 or higher.

record-type { sgsnpdprecord | sgwrecord }

Important This keyword is available only when the SaMOG Mixed Mode license (supporting both 3G and 4G) is configured.

Default: sgwrecord

Specifies the SaMOG CDR type to use.

For an SaMOG 3G license, this keyword will not be available. However, sgsnpdprecord type will be used as the default record type.

served-mnai

Default: Disabled

This keyword controls the inclusion of the optional field "Served MNAI" in the x-CDRs.

served-pdp-pdn-address-extension

Default: Disabled

In support of IPv4v6 dual-stack PDP address types, this keyword causes the service to include IPv4v6 address information in the CDR. The IPv4 address goes in the Served PDP PDN Address Extension field and the IPv6 address goes in the Served PDP Address or Served PDP PDN Address field.

Important This attribute will not be displayed if the GTPP dictionary is set to custom34.

Note For SGSN, on enabling **served-pdp-pdn-address-extension** all custom S-CDR dictionaries will support the CDR field "Served PDP/ PDN Address extension" except for the following dictionaries:

- custom17
- custom18
- custom23
- custom42
- custom41

served-pdp-pdn-address-prefix-length

Default: Enabled

In support of IPv6 prefix delegation, this keyword causes the service to include this field "Served PDP PDN Address" in the x-CDRs.

If this field is configured, the servedPDPPDNAddress field will support reporting the IPv6 prefix length as outlined in 3GPP 32.298. The prefix length will only be reported if:

- it is configured
- it is not the default length of 64
- it is an IPv6 or IPv4v6 call

af-record-info

Defaulr: Disabled

Enable attribute to include the**AF Charging Identifier** keyword and associated flow identifiers generated by the AF and received by the P-GW over Gx interfaces. This keyword is applicable to custom24 GTPP dictionary.

sgsn-change

Default: Enabled

This keyword is specific to SGSN and is license restricted.

This keyword controls the inclusion of the S-CDR attribute "SGSN Change" in the S-CDRs. It is enabled by default and the attribute "SGSN Change" is included in the S-CDRs by default.

Note For SGSN specific custom33 dictionary, it is recommended to disable this keyword before an upgrade to prevent billing issues.

sgw-ipv6-addr

Default: Disabled

Specifying this option allows to configure the S-GW IPv6 address.

Important This attribute can be controllably configured in custom24 and custom35 SGW-CDR dictionaries.

sms { destination-number | recording-entity | service-centre }

This keyword is specific to the SGSN.

Entering this keyword causes the inclusion of an SMS-related field in the SMS-MO-CDR or SMS-MT-CDR.

destination-number: Includes the "destinationNumber" field in the SMS-MO-CDR or SMS-MT-CDR.

recording-entity: Includes the "recordingEntity" field in the SMS-MO-CDR or SMS-MT-CDR.

service-centre: Includes the "serviceCentre" field in the SMS-MO-CDR or SMS-MT-CDR.

sna-ipv6-addr

Default: Disabled

Specifying this option allows to configure the Serving Node IPv6 Address (SNAv6).

Important This attribute can be controllably configured in custom24 and custom35 SGW-CDR dictionaries.

sponsor-id

Default: Disabled

Includes the Sponsor ID and Application-Service-Provider-Identity fields in PGW-CDR.

Note that the "Sponsor ID" and "Application-Service-Provider-Identity" attributes will be included in PGW-CDR if the PCEF supports Sponsored Data Connectivity feature or the required reporting level is sponsored connectivity level as described in 3GPP TS 29.212.

This feature is implemented to be in compliance with Release 11 3GPP specification for CDRs. So, this behavior is applicable to all GTPP dictionaries that are Release 11 compliant, i.e. custom35.

start-time

Default: Enabled

This keyword controls the inclusion of the optional field "Start-Time" in the x-CDRs.

stop-time

Default: Enabled

This keyword controls the inclusion of the optional field "Stop-Time" in the x-CDRs.

twanuli

Default: Disabled

This keyword controls the inclusion of the optional field "TWAN User Location Information" in the CDRs.

ue-tun-ip-port

Default: Disabled

In 21.9.5 and later releases, this keyword is introduced for P-GW to include new parameter in CDR generated for S2b (VoWifi) call/subscriber.

Important This keyword is applicable to custom24 GTPP dictionary.

uwanuli

Default: Disabled

This keyword controls the inclusion of the optional field "UWAN User Location Information" in the CDRs.

uli

Default: Enabled

This keyword controls the inclusion of the optional field "User Location Information" in the x-CDRs.

user-csg-information

Default: Disabled

This keyword controls the inclusion of the optional field "User CSG Information" in the x-CDRs.

Important Currently, UCI values are only supported for SGW-CDRs.

This attribute will not be displayed if the GTPP dictionary is set to custom11, custom34, or custom35.

+

Indicates that this command can be entered multiple times to configure multiple attributes.

Usage Guidelines

This command dictates some of the optional information fields that should be reported in CDRs generated by the GGSN. In addition, it controls how the information for some of the mandatory fields are reported.

Fields described as optional by the standards but not listed above will always be present in the CDRs, except for Record Extensions (which will never be present).

Example

The following command disables the inclusion of the field "SGSN Change" in the S-CDR:

```
no gtpp attribute sgsn-change
```

Example

The following command dictates that the time provided in the Duration field of the CDR is reported in milliseconds:

```
gtpp attribute duration-ms
```

gtpp trigger

Disables GTPP trigger conditions that cause either partial CDR record closure or opening of a new CDR record container. GTPP Triggers are specified in 3GPP TS 32.251 v6.6.0. All GTPP trigger changes take effect immediately, except **volume-limit**.

Product

ECS

GGSN

P-GW

SAEGW

SGSN

S-GW

Privilege

Security Administrator, Administrator

Command Modes

Exec > Global Configuration > Context Configuration > GTPP Server Group Configuration

configure > **context** *context_name* > **gtpp group** *group_name*

Entering the above command sequence results in the following prompt:

```
[context_name]host_name(config-gtpp-group)#
```

Syntax Description

```
gtpp trigger { apn-ambr-change | [ default-bearer-only |
all-non-gbr-bearers | all-bearers ] | cell-update |  ciot-userplane-change|
 dcca | direct-tunnel | egcdr max-losdv | ggsn-preservation-mode-change
| inter-plmn-sgsn-change | ms-timezone-change | plmn-id-change | qos-change
 | rat-change [ generate { cdr | container } ] | routing-area-update |
service-idle-out | serving-node-change-limit | sgsn-change-limit |
tariff-time-change | time-limit | uli-change | volume-limit }
default gtpp trigger
no gtpp trigger { apn-ambr-change | [ default-bearer-only |
all-non-gbr-bearers | all-bearers ] | cell-update |  ciot-userplane-change|
 dcca | direct-tunnel | egcdr max-losdv | ggsn-preservation-mode-change
| inter-plmn-sgsn-change | ms-timezone-change | plmn-id-change | qos-change
 | rat-change [ generate { cdr | container } ] | routing-area-update |
service-idle-out | serving-node-change-limit | sgsn-change-limit |
tariff-time-change | time-limit | uli-change | volume-limit }
```

default

Sets the specified trigger condition back to the default setting. All trigger conditions are enabled by default.

no

Disables the specified trigger condition.

apn-ambr-change [default-bearer-only | all-non-gbr-bearers | all-bearers]

Default: Disabled

Enables APN AMBR trigger only for default-bearer or for all bearers for that PDN or selectively for apn-non-gbr bearers.

Important This keyword option will be available only if a valid license is installed. For more information, contact your Cisco account representative.

The APN Aggregate Maximum Bit Rate (AMBR) is a subscription parameter stored per APN. It limits the aggregate bit rate that can be expected to be provided across all non-GBR bearers and across all PDN connections of the same APN. Each of these non-GBR bearers potentially utilize the entire APN AMBR, e.g. when the other non-GBR bearers do not carry any traffic.

In 15.0 and later releases, this CLI command should be configured along with the following additional options to enable APN-AMBR trigger for SGW-CDRs in all GTPP dictionaries.

- **default-bearer-only**: Adds container only to default bearer.
- **all-non-gbr-bearers**: Adds container to all non-gbr-bearers.
- **all-bearers**: Adds containers for all bearers.

Important This CLI command and the associated options are not available for products other than S-GW and P-GW.

The first container of each CDR includes apn-ambr fields along with QoS. In the following containers this field is present if previous change condition is "QoS change" or "APN AMBR Change".

cell-update

Enables the cell update trigger for S-CDRs, if the dictionary specified in the **gtpp dictionary** configuration includes support for cell update. This trigger is available only for 2G. Currently, custom18 dictionary supports the cell update trigger.

ciot-userplane-change

Enables User Plane change trigger for CDR.

dcca

This keyword enables or disables the addition of LOSDV in PGW-CDR for the following DCCA generated triggers.

- Time Threshold Reached

- Volume Threshold Reached
- Service Specific Unit Threshold Reached
- Time Exhausted
- Volume Exhausted
- Validity Timeout
- Reauthorization Request
- Continue Ongoing Session
- Retry And Terminate Ongoing Session
- Terminate Ongoing Session
- Service Specific Unit Exhausted
- Envelope Closure

direct-tunnel

Enables the direct tunnel trigger for CDRs.

egcdr max-losdv

Enables the trigger for an eG-CDR/P-CDR if the List of Service Data Volume (LoSDV) containers crosses the configured limit for LOSDV containers. Default: Disabled

ggsn-preservation-mode-change

This keyword is for GGSN only.

This trigger enables the preservation-mode-change trigger for G-CDR.

inter-plmn-sgsn-change

This keyword is for GGSN only.

Disabling this trigger ignores an Inter-PLMN SGSN change and doesn't release a G-CDR. Default: Enabled

ms-timezone-change

This keyword is specific to GGSN.

No partial record closure for a time zone change occurs when this trigger is disabled. MS time zone change should be applicable only for 3GPP R6 based GTPP dictionaries. Default: Enabled

plmn-id-change

This trigger keyword is specific to the 2G SGSN and is proprietary (non-standard).

Enables the PLMNID change trigger for S-CDRs if the dictionary specified in the **gtpp dictionary** configuration supports the PLMNID change. If enabled, the SGSN generates a partial S-CDR when the MS changes the PLMN while under the same SGSN (intra-system intra-SGSN PLMN-ID handover). Currently, custom18 dictionary supports this trigger. Default: Disabled

qos-change

Enables the QoS-change trigger for CDRs. Disabling this trigger ignores a QoS-change and does not open a new CDR for it. Default: Enabled

When QoS changes are observed, the system generates only containers. However when the max-container condition is reached, an interim CDR is generated.

rat-change [generate { cdr | container }]

Enables or disables the partial record closure for a RAT change. If disabled, no partial record closure for a RAT change occurs. RAT change should be applicable only for 3GPP R6 based GTPP dictionaries. Default: Enabled

In SGSN, RAT change trigger (2G<->3G) means inter-service handoff (SGSN service <-> GPRS service). If this trigger is enabled, after the RAT change interim CDR is generated. After this RAT change CDR, CDR thresholds such as volume/time etc. and GTPP Group are applicable from new service. If RAT change trigger is disabled, the CDR thresholds and GTPP group etc. will not change and will continue to use from old service.

After the RAT change, the System Type field in CDR changes to indicate the new system type. If this trigger is disabled, then the next CDR generated will indicate System Type, but the data count in the CDR does not necessarily belong to the system type indicated in CDR; instead, it may belong to both 2G and 3G as CDR is not closing when handover takes place.

Important The System Type field in CDR-related change is not applicable to customized CDR formats, which does not use the System Type field.

generate { cdr | container }: Sets generation of CDR or just a Container on a RAT change.

cdr: Generates a CDR on a RAT-change.

container: Generates a container only on a RAT-change.

routing-area-update

Enables the routing-area-update trigger for CDRs.

service-idle-out

This keyword enables or disables the addition of LOSDV in PGW-CDR when a service idles out.

Note that the CDR module receives service idle out trigger from DCCA module when the quota hold timer expires, or from ACS manager when rulebase has a service idle out configuration.

serving-node-change-limit [also-intra-sgsn-multiple-address-group-change]

This keyword is enabled for P-GW, S-GW, and GGSN. However, the **also-intra-sgsn-multiple-address-group-change** is enabled only for GGSN. Default: Enabled

Disabling this trigger ignores an SGSN change and does not add the SGSN IP address into the SGSN address list of the CDR. This helps to reduce the release of CDRs due to SGSN changes crossing the configured limit.

also-intra-sgsn-multiple-address-group-change: This keyword includes Intra-SGSN group changes as an SGSN change.

sgsn-change-limit [also-intra-sgsn-multiple-address-group-change]

This keyword is obsolete and is available to maintain the backward compatibility for existing customers. The new keyword for **sgsn-change-limit** is **serving-node-change-limit**. Default: Enabled

Disabling this trigger ignores an SGSN change and does not add the SGSN IP address into the SGSN address list of the CDR. This helps to reduce the release of CDRs due to SGSN changes crossing the configured limit.

also-intra-sgsn-multiple-address-group-change: This keyword includes Intra-SGSN group changes as an SGSN change.

tariff-time-change

When this trigger is disabled, container closure does not happen for a tariff-time change. Default: Enabled

This trigger is applicable for G-MB-CDRs for MBMS session too.

time-limit

When this trigger is disabled, no partial record closure occurs when the configured time limit is reached. Default: Enabled

This trigger is applicable for G-MB-CDRs for MBMS session too.

uli-change

Enables the user location update trigger for eG-CDRs/PGW-CDRs/SGW-CDRs, if the dictionary specified in the GTPP dictionary configuration includes support for user location update trigger. Default: Enabled

volume-limit

When this trigger is disabled no partial record closure occurs when volume limit is reached. Default: Enabled

This trigger is applicable for G-MB-CDRs for MBMS session too.

Usage Guidelines

Use this command to disable or enable GTTP triggers that can cause partial CDR record closure or cause a new CDR to be created.

Example

The following command disables partial record closure when a configured time limit is reached:

```
gtpp trigger time-limit
```

The following command re-enables partial record closure when a configured time limit is reached:

```
no gtpp trigger time-limit
```

gtpu-error-ind

Configures the actions to be taken upon receiving a GTP-U error indication from an RNC, eNodeB, SGSN, or P-GW.

Product

S-GW

SAEGW

Privilege

Administrator

Command Modes

Exec > Global Configuration > Context Configuration > S-GW Service Configuration

configure > **context** *context_name* > **sgw-service** *service_name*

Entering the above command sequence results in the following prompt:

```
[context_name]host_name(config-sgw-service)#
```

Syntax Description

```
gtpu-error-ind { { s12 | s1u | s11u } { local-purge | page-ue [
custom1-behavior ] } | { s4u | s5u } { local-purge | signal-peer } }
default gtpu-error-ind { s12 | s1u | s11u | s4u | s5u }
```

default

Resets the command to the default action for the specified interface. For S12 and S1-U, **page-ue** is the default action. For S4-U and S5-U, **local-purge** is the default action.

{ s12 | s1u | s11u }{ local-purge | page-ue [custom1-behavior] }

Specifies the action to take when a GTP-U error indication is received from a Radio Network Controller (RNC) over an S12 interface or from an eNodeB over the S1-U interface.

local-purge: The S-GW clears the affected bearer (or PDN if error-indication is received on default bearer) locally without informing peer.

page-ue [custom1-behavior]: The S-GW moves the complete UE state to S1-Idle and starts paging for this UE. If the custom1-behavior option is specified, the S-GW will guard the paging attempt with a timer of 60 seconds. Within this time the bearer must have the eNodeB TEID refreshed by an MME. Otherwise, the S-GW will clear the affected bearer with signaling. This is the default action for GTP-U error indication messages received on the S12 and S1-U interfaces.

{ s4u | s5u }{ local-purge | signal-peer }

Specifies the action to take when a GTP-U error indication is received from an SGSN over an S4-U interface or from a P-GW over the S5-U interface.

local-purge: The S-GW clears the affected bearer (or PDN if error-indication is received on a default bearer) locally without informing the peer. This is the default action for GTP-U error indication messages received on the S4-U and S5-U interfaces.

signal-peer: The S-GW initiates control signalling towards the peer MME and P-GW. When signalling:

- For a bearer deletion, the S-GW sends a Delete-Bearer-Command message to the P-GW and a Delete-Bearer-Request (with EBI) message to the MME.
- For PDN deletion, the S-GW sends a Delete-Session-Request message to the P-GW and a Delete-Bearer-Request (with LBI) message to the MME.
- The S-GW will not wait for Delete replies from the peer. The request will be sent only once and local resources will be reset.

Usage Guidelines

Use this command to specify the action to taken upon receiving a GTP-U error indication from an RNC over an S12 interface, an eNodeB across an S1-U interface, an SGSN over an S4-U interface, or from a P-GW across an S5-U interface.

Example

The following command sets the action to take upon receipt of a GTP-U error indication from the eNodeB to clear affected bearer:

```
gtpu-error-ind s1u local-purge
```

ie-override

This command is used to override the RAT type AVP value with the configured value for messages sent from MME to HSS.

Important This command ensures backward compatibility with previous releases as the HSS does not support the new NB-IoT RAT type.

Product MME

Privilege Administrator

Command Modes Exec > Global Configuration > Call Control Profile Configuration

configure > call-control-profile *profile_name*

Entering the above command sequence results in the following prompt:

`[local]host_name(config-call-control-profile-profile_name)#`

Syntax Description

```
[ remove ] ie-override s6a rat-type wb-eutran
```

remove

The keyword remove deletes the existing configuration.

ie-override

This keyword allows the operator to configure IE override in messages sent from MME to HSS.

s6a

This keyword is used to specify the interface as s6a. The s6a interface used by the MME to communicate with the Home Subscriber Server (HSS).

rat-type

Use this keyword to configure the supported RAT type AVP IE.

wb-eutran

Use this keyword to specify the WB-EUTRAN AVP Value.

Usage Guidelines

Use this command to override the RAT type AVP value with the configured value for messages sent from MME to HSS over the s6a interface. If the configured RAT type is NB-IoT, it is changed to wb-eutran for messages sent from the MME to HSS. This command is not enabled by default.

Example

The following command is used to enable override of the RAT type AVP value with the configured value of WB-EUTRAN:

```
ie-override s6a rat-type wb-eutran
```

iftask mcdmatxbatch

Configures multi-channel direct memory access (MCDMA) transmit batching. The MCDMA is the path from the IFTASK to the SESSMGR. This command applies only to StarOS on virtualized platforms.

Product

All

Privilege

Operator

Command Modes

Exec > Global Configuration

configure

Entering the above command sequence results in the following prompt:

```
[local]host_name(config)#
```

Syntax Description

```
[ no ] iftask mcdmatxbatch { burstsize number_of_packets | latency milliseconds }
```

no

Deletes the setting for iftask mcdmatxbatch.

burstsize *number_of_packets*

Maximum packets per burst from 1 through 1024.

latency *milliseconds*

Not currently supported.

Usage Guidelines

The following example sets the maximum number of packets per burst for MCDMA to 512:

```
iftask mcdmatxbatch burstsize 512
```

iftask txbatch

Configures transmit batching. This command applies only to StarOS on virtualized platforms.

Product

All

Privilege

Operator

Command Modes

Exec > Global Configuration

configure

Entering the above command sequence results in the following prompt:

`[local]host_name(config)#`

Syntax Description

[no] iftask txbatch { burstsize *number_of_packets* **| flush_latency | latency** *milliseconds* **}**

no

Deletes the setting for iftask txbatch.

burstsize *number_of_packets*

Specifies the maximum number of packets from 1 through 1024 to accumulate in a vector before sending to the ethernet interface.

latency *milliseconds*

Not currently supported.

Usage Guidelines

Use this command to configure the transmit batching parameters for system-wide IFTASK operation.

The following example sets the maximum number of packets per burst for MCDMA to *512*:

```
iftask txbatch burstsize 512
```

The following example sets the maximum wait time to *1000* milliseconds to flush the bytes on the control port:

```
iftask txbatch flush_latency 1000
```

ip name-servers

Modifies the list of domain name servers the current context may use for logical host name resolution.

Product

All

Privilege

Security Administrator, Administrator

Command Modes

Exec > Global Configuration > Context Configuration

configure > **context** *context_name*

Entering the above command sequence results in the following prompt:

`[context_name]host_name(config-ctx)#`

Syntax Description

ip name-servers *ip_address secondary_ip_address* [*third_ip_address*]
no ip name-servers *ip_address*

no

Indicates the name server specified is to be removed from the list of name servers for the current context.

ip_address

Specifies the IP address of a domain name server using IPv4 dotted-decimal or IPv6 colon-separated-hexadecimal notation.

secondary_ip_address

Specifies the IP address of a secondary domain name server using either IPv4 dotted-decimal or IPv6 colon-separated-hexadecimal notation.

third_ip_address

Specifies the IP address of a third domain name server using either IPv4 dotted-decimal or IPv6 colon-separated-hexadecimal notation. (VPC only)

Usage Guidelines

Manage the list of name servers the current context may use in resolving logical host names.

The DNS can be specified at the Context level in Context configuration as well as at the APN level in APN Configuration Mode with **dns** and **ipv6 dns** commands, or it can be received from AAA server.

When DNS is requested in PCO configuration, the following preference will be followed for DNS value:

1. DNS Values received from LNS have the first preference.
2. DNS values received from RADIUS Server has the second preference.
3. DNS values locally configured with APN with **dns** and **ipv6 dns** commands has the third preference.
4. DNS values configured at context level has the last preference.

Important The same preference would be applicable for the NBNS servers to be negotiated via ICPC with the LNS.

Example

```
ip name-servers 10.2.3.4
```

ip qos-dscp

Defines the IP parameters for this APN profile.

Product

MME

SGSN

S-GW

SAEGW

Privilege

Administrator

Command Modes

Exec > Global Configuration > APN Profile Configuration

configure > **apn-profile** *profile_name*

Entering the above command sequence results in the following prompt:

`[local]host_name(apn-profile-profile_name)#`

Syntax Description

```
ip { qos-dscp {  { { downlink | uplink } { backgroundforwarding |
conversational forwarding | interactive traffic-handling-priority priority
forwarding | streaming forwarding } + } s11u-mme  value }  | source-violation
 { deactivate [ all-pdp | exclude-from accounting | linked-pdp |
tolerance-limit } | discard [ exclude-from-accounting ] | ignore }
default ip { qos-dscp [ downlink | uplink | s11u-mme ] | source-violation
 }
no ip qos-dscp { downlink | uplink } { background | conversational |
interactive | streaming } +
```

Important All parameters not specifically configured will be included in the configuration with default values.

default

Resets the configuration to the default values.

no

Disables the specified IP QoS-DSCP mapping.

qos-dscp

Configures the Differentiated Services Code Point (DCSP) marking to be used for sending packets of a particular 3GPP QoS class.

downlink | uplink

Configures the packets for either downlink (network to subscriber) or uplink (subscriber to network) direction. **downlink** and **uplink** configuration must include one or more of the following:

- **background** - Configures the DSCP marking to be used for packets of sessions subscribed to 3GPP background class. Must be followed by a DSCP marking
- **conversational** - Configures the DSCP marking to be used for packets of sessions subscribed to 3GPP conversational class. Must be followed by a DSCP marking
- **interactive** - Configures the DSCP marking to be used for packets of sessions subscribed to different traffic priorities in the 3GPP interactive class. Must be followed by a traffic handling priority (THP): 1, 2, or 3.
- **streaming** - Configures the DSCP marking to be used for packets of sessions subscribed to 3GPP streaming class. Must be followed by a DSCP marking

DSCP marking options

Downlink and uplink must include a DSCP forwarding marking; supported options include:

- af11 - Designates use of Assured Forwarding 11 PHB
- af12 - Designates use of Assured Forwarding 12 PHB
- af13 - Designates use of Assured Forwarding 13 PHB
- af21 - Designates use of Assured Forwarding 21 PHB
- af22 - Designates use of Assured Forwarding 22 PHB
- af23 - Designates use of Assured Forwarding 23 PHB
- af31 - Designates use of Assured Forwarding 31 PHB
- af32 - Designates use of Assured Forwarding 32 PHB
- af33 - Designates use of Assured Forwarding 33 PHB
- af41 - Designates use of Assured Forwarding 41 PHB
- af42 - Designates use of Assured Forwarding 42 PHB
- af43 - Designates use of Assured Forwarding 43 PHB
- be - Designates use of Best Effort forwarding PHB
- ef - Designates use of Expedited Forwarding PHB

Forwarding defaults for both uplink and downlink are:

- conversational - ef;
- streaming - af11;
- interactive 1 - ef;
- interactive 2 - af21;
- interactive 3 - af21;
- background - be

s11u-mme *value*

This keyword is used to configure the S11-U interface parameters. The DSCP values can be specified using this keyword. The DSCP value for S11-U interface can be separately specified for each APN. This keyword is enabled by default. The default value is “be”. Listed below are DSCP values which can be configured for the S11U interface:

- af11 - Designates use of Assured Forwarding 11 PHB

- af12 - Designates use of Assured Forwarding 12 PHB
- af13 - Designates use of Assured Forwarding 13 PHB
- af21 - Designates use of Assured Forwarding 21 PHB
- af22 - Designates use of Assured Forwarding 22 PHB
- af23 - Designates use of Assured Forwarding 23 PHB
- af31 - Designates use of Assured Forwarding 31 PHB
- af32 - Designates use of Assured Forwarding 32 PHB
- af33 - Designates use of Assured Forwarding 33 PHB
- af41 - Designates use of Assured Forwarding 41 PHB
- af42 - Designates use of Assured Forwarding 42 PHB
- af43 - Designates use of Assured Forwarding 43 PHB
- be - Designates use of Best Effort forwarding PHB
- cs0 - Designates use of Class Selector 0 PHB
- cs1 - Designates use of Class Selector 1 PHB
- cs2 - Designates use of Class Selector 2 PHB
- cs3 - Designates use of Class Selector 3 PHB
- cs4 - Designates use of Class Selector 4 PHB
- cs5 - Designates use of Class Selector 5 PHB
- cs6 - Designates use of Class Selector 6 PHB
- cs7 - Designates use of Class Selector 7 PHB
- ef - Designates use of Expedited Forwarding PHB

source-violation

Configures settings related to IP source-violation detection with one of the following criteria:

- **deactivate** - deactivate the PDP context with one of the following conditions:
  - **all-pdp** - deactivates all PDP context of the MS/UE. Default is to deactivate errant PDP contexts.
  - **exclude-from-accounting** - excludes packets having an invalid source IP address from the statistics used in the accounting records.
  - **linked-pdp** - deactivate all associated pdp contexts (primary and secondary). Default is to deactivate errant pdp context.
  - **tolerance-limit** - Configures maximum number of allowed IP source violations before the session is deactivated.

- **discard** - discard errant packets, can include the following option:

- **exclude-from-accounting** - excludes packets having an invalid source IP address from the statistics used in the accounting records.

- **ignore** - ignore checking of packets for MS/UE IP source violation.

Usage Guidelines

This command configures a range of IP functions to be associated with the APN profile; such as:

- SGSN/S-GW action in response to detected IP source violations,
- DSCP marking for downlink and uplink configuration per traffic class,
- QoS class diffserv code.
- Configures the S11U interface parameters.

Example

The following command configures the APN profile to instruct the SGSN or S-GW not to check incoming packets for IP source violation information:

```
ip source-violation ignore
```

The following command configures the S11-U interface parameters and specifies the DSCP marking value as “ef”:

```
ip qos-dscp s11u-mme ef
```

nb-iot

This command enables Extended Discontinuous Reception (eDRX) and configures the respective parameters for NB-IoT subscribers on the MME.

Product

MME

Privilege

Administrator

Command Modes

Exec > Global Configuration > Call Control Profile Configuration

configure > **call-control-profile** *profile_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-call-control-profile-profile_name)#
```

Syntax Description

```
nb-iot  {  edrx { ptw ptw_value edrx-cycle cycle_length_value | ue-requested }
 [ dl-buf-duration [ packet-count packet_count_value ] ] | mo-exception-data
 reporting-threshold-value threshold_value  }
remove nb-iot {  edrx  | mo-exception-data }
```

remove

This keyword disables the eDRX configuration on the MME for NB-IoT subscribers.

edrx

This keyword configures extended discontinuous reception parameters.

ptw *ptw_value*

This keyword configures the Paging Time Window (PTW) value. *ptw_value* must be an integer value in seconds. The allowed values are 2.56, 5.12, 7.68, 10.24, 12.80, 15.36, 17.92, 20.48, 23.04, 25.60, 28.16, 30.72, 33.28, 35.84, 38.40 and 40.96 seconds.

ue-requested

This keyword specifies the UE requested values of the Paging Time Window (PTW) and the eDRX cycle length received from the UE in the Attach Request or TAU Request message be accepted.

edrx-cycle *cycle_length_value*

This keyword configures the eDRX cycle length. *cycle_length_value* is an integer value in seconds. The allowed values are 5.12, 7.68, 10.24, 12.80, 15.36, 17.92, 20.48, 40.96, 81.92, 163.84, 327.68, 655.36, 1310.72, 2621.44, 5242.88 and 10485.76 seconds.

dl-buf-duration

This optional keyword sends downlink buffer duration in DDN ACK when unable to page UE.

packet-count *packet_count_value*

This optional keyword sends "DL Buffering Suggested Packet Count" in DDN ACK when unable to page UE. The *packet_count_value* is an integer value from 0 to 65535. If the *packet_count_value* is not configured locally, the subscription provided value for the *packet_count_value* is used. The subscription value can be 0 in which case the packet count IE will not be sent for that subscriber even if it is configured locally.

mo-exception-data

Configures NBIOT RRC Cause MO Exception Data counter.

reporting-threshold-value *value*

Specifies reporting threshold value. *value* Must be an integer from 1 to 50.

Usage Guidelines

Use this command to enable eDRX on the MME for NB-IoT subscribers. The operator can use this command for:

- Accept eDRX parameters: Paging Time Window (PTW) and eDRX cycle length value, from the UE
- Configure PTW and eDRX cycle length value
- Configure downlink buffer duration in DDN ACK when unable to page UE
- Configure "DL Buffering Suggested Packet Count" in DDN ACK when unable to page UE

When the eDRX feature is enabled on the MME, it pages the NB-IoT subscribers only at valid paging occasions. The MME sends the NB-IoT eDRX paging parameters to the eNodeB during paging. The operator can either configure the option to accept the UE requested values or configure the values using this command. This command is not enabled by default.

A similar CLI command is implemented for WB-EUTRAN subscribers. Both WB-UTRAN eDRX and NB-IoT eDRX parameters can be configured on the system for WB-UTRAN and NB-IoT subscribers.

See the *eDRX Support on the MME* feature chapter in the *MME Administration Guide* for more information.

Example

The following command configures the PTW and eDRX cycle length. The command also sends the downlink buffer duration in the DDN ACK along with a suggested packet count:

```
nb-iot edrx ptw 256 edrx-cycle 512 dl-buf-duration packet-count 10
```

path-failure

Configures the action to take upon the occurrence of a path failure between the S-GW and the MME, P-GW, RNC, SGSN, or eNodeB.

Product

S-GW

SAEGW

Privilege

Administrator

Command Modes

Exec > Global Configuration > Context Configuration > S-GW Service Configuration

configure > **context** *context_name* > **sgw-service** *service_name*

Entering the above command sequence results in the following prompt:

```
[context_name]host_name(config-sgw-service)#
```

Syntax Description

```
path-failure { s11 | s11u | s12 | s1u | s4 | s4u | s5 | s5u } ( local-purge
 | signal-peer }
default path-failure { s11 | s11u | s12 | s1u | s4 | s4u | s5 | s5u } (
local-purge | signal-peer }
```

default

Returns the command to the default setting of "local purge" for the selected interface.

{ s11 | s12 | s1u | s4 | s4u | s5 | s5u }

Specifies the interface to which the action will be applied.

s11: Applies the path failure action to the S11 interface between the S-GW and the MME.

s11u: Applies the path failure action to the S11-U interface between the S-GW and the MME.

s12: Applies the path failure action to the S12 interface between the S-GW and the RNC.

s1u: Applies the path failure action to the S1-U interface between the S-GW and the eNodeB.

s4: Applies the path failure action to the S4 control plane interface between the S-GW and the SGSN.

s4u: Applies the path failure action to the S4-U user plane interface between the S-GW and the SGSN.

s5: Applies the path failure action to the S5 interface between the S-GW and the P-GW.

s5u: Applies the path failure action to the S5-U user plane interface between the S-GW and the P-GW.

{ local-purge | signal-peer }

Specifies the action to apply to the selected interface.

local-purge: The S-GW clears the affected bearer (or PDN if path failure is received on a default bearer) locally without informing the peer. This is the default action for all interface.

signal-peer: The S-GW initiates control signalling towards the peer MME and P-GW. When signalling:

- For a bearer deletion, the S-GW sends a Delete-Bearer-Command message to the P-GW and a Delete-Bearer-Request (with EBI) message to the MME.
- For PDN deletion, the S-GW sends a Delete-Session-Request message to the P-GW and a Delete-Bearer-Request (with LBI) message to the MME.
- The S-GW will not wait for Delete replies from the peer. The request will be sent only once and local resources will be reset.

Usage Guidelines

Use this command to specify the type of action to take when a path failure occurs on one of the supported interfaces.

Example

The following command sets the path failure action for the S5 interface to "signal peer":

```
path-failure s5 signal-peer
```

pco-options

The following commands are explained below:

pco-options custom1

In releases prior to 21.1.V0 (N5.1):

This command controls the sending of customized PCO (Protocol Configuration Options) options in the network to MS GTP messages and configures APN to include link MTU in PCO IE.

In release 21.1.V0 (N5.1) and later:

Configures APN to include protocol configuration options in PCO/APCO/EPCO IE as applicable.

Product

GGSN

P-GW

Privilege

Security Administrator, Administrator

Command Modes

Exec > Global Configuration > Context Configuration > APN Configuration

configure > **context** *context_name* > **apn** *apn_name*

Entering the above command sequence results in the following prompt:

```
[context_name]host_name(config-apn)#
```

Syntax Description

pco-options { custom1 [ue-requested] | link-mtu *bytes* **[non-ip** *bytes* **] }**epdg fqdn *domain_name*
{ default | no } pco-options [custom1 | link-mtu [non-ip]]

custom1

Enable sending of customized PCO options in the network to MS messages; send customized PCO options to all UEs regardless of support.

ue-requested

Enable sending of customized PCO options in the network to MS messages for "UE-Requested" mode; send PCO to only UEs that request customized PCO options.

link-mtu *bytes*

In releases prior to 21.1.V0 (N5.1):

Configures APN to include link MTU in PCO IE, if it is requested by UE.

In release 21.1.V0 (N5.1) and later:

Configures APN to include Link MTU in PCO/APCO/EPCO IE of IP and Non-IP PDN connection response, if it is requested by UE.

When UE sends IPv4 Link MTU Size PCO request during Initial attach/ Standalone PDN connection, then the S-GW/SGSN/HSGW sends the same transparently in Create Session Request, Create/Update PDP Context Request, or PBU to P-GW, GGSN, or PMIP-PGW. Create Session Response, Create/ Update PDP Context Response/ PBA will be sent with latest configured MTU size PCO value in APN. If UE is in outbound roaming, then default value (1500) will be provided in the MTU size PCO.

bytes must be an integer from 1280 to 2000.

Default: 1500

non-ip *bytes*

Link MTU for Non-IP PDN. *bytes* must be an integer from 128 to 2000. Default is 1358.

epdg

Enables operator specific epdg selection in the PCO. By default it is disabled.

fqdn

Specifies fully qualified domain name. Based on this, IP addresses would be queried from the DNS.

default

Disable sending of customized PCO options in the network to MS messagesand/ or sets the link MTU PCO to 1500 bytes.

no

Do not send customized PCO options to any UEs and/ or sets the link MTU PCO to 1500 bytes.

Usage Guidelines

Use this command to enable or disable sending of customized PCO options in the network to MS GTP messages and configure link MTU size PCO value.

Important Configure custom PCO values in **pco-custom1** command in *ACS Charging Action Configuration Mode*.

Example

The following command enables sending customized PCO options to all UEs regardless of support:

```
pco-options custom1
```

The following command disables sending of customized PCO options in the network to MS messagesand sets the link MTU PCO to 1500 bytes:

```
default pco-options
```

The following command configures epdg.com

```
pco-options epdg fqdn epdg.com
```

pco-options custom2

This command controls the sending of customized PCO (Protocol Configuration Options) options in the network to MS GTP messages and configures APN to include in PCO IE.

Product

GGSN

P-GW

Privilege

Security Administrator, Administrator

Command Modes

Exec > Global Configuration > Context Configuration > APN Configuration

configure > **context** *context_name* > **apn** *apn_name*

Entering the above command sequence results in the following prompt:

```
[context_name]host_name(config-apn)#
```

Syntax Description

[no] pco-options { custom2 [ue-requested] value *custom_value* **| value** *custom_value* **}**

no

Removes PCO configuration at APN configuration mode

custom2

Enable sending of customized PCO options in the network to MS messages; send customized PCO options to all UEs regardless of support.

ue-requested

Enable sending of customized PCO options in the network to MS messages for "UE-Requested" mode; send PCO to only UEs that request customized PCO options.

Usage Guidelines

Use this command to enable or disable sending of customized PCO options in the network to MS GTP messages and configure link MTU size PCO value.

Example

The following command enables sending customized PCO options to all UEs regardless of support:

```
pco-options custom2
```

pco-options custom3

This command controls the sending of customized PCO (Protocol Configuration Options) options in the network to MS GTP messages and configures APN to include in PCO IE.

Product

GGSN

P-GW

Privilege

Security Administrator, Administrator

Command Modes

Exec > Global Configuration > Context Configuration > APN Configuration

configure > **context** *context_name* > **apn** *apn_name*

Entering the above command sequence results in the following prompt:

[*context_name*]*host_name*(config-apn)#

Syntax Description

[no] pco-options { custom3 [ue-requested] value *custom_value* **| value** *custom_value* **}**

no

Removes PCO configuration at APN configuration mode

custom3

Enable sending of customized PCO options in the network to MS messages; send customized PCO options to all UEs regardless of support.

ue-requested

Enable sending of customized PCO options in the network to MS messages for "UE-Requested" mode; send PCO to only UEs that request customized PCO options.

Usage Guidelines

Use this command to enable or disable sending of customized PCO options in the network to MS GTP messages and configure link MTU size PCO value.

Example

The following command enables sending customized PCO options to all UEs regardless of support:

```
pco-options custom3
```

pco-options custom4

This command controls the sending of customized PCO (Protocol Configuration Options) options in the network to MS GTP messages and configures APN to include in PCO IE.

Product

GGSN

P-GW

Privilege

Security Administrator, Administrator

Command Modes

Exec > Global Configuration > Context Configuration > APN Configuration

configure > **context** *context_name* > **apn** *apn_name*

Entering the above command sequence results in the following prompt:

```
[context_name]host_name(config-apn)#
```

Syntax Description

[no] pco-options { custom4 [ue-requested] value *custom_value* **| value** *custom_value* **}**

no

Removes PCO configuration at APN configuration mode

custom4

Enable sending of customized PCO options in the network to MS messages; send customized PCO options to all UEs regardless of support.

ue-requested

Enable sending of customized PCO options in the network to MS messages for "UE-Requested" mode; send PCO to only UEs that request customized PCO options.

Usage Guidelines

Use this command to enable or disable sending of customized PCO options in the network to MS GTP messages and configure link MTU size PCO value.

Example

The following command enables sending customized PCO options to all UEs regardless of support:

```
pco-options custom4
```

pco-options custom5

This command controls the sending of customized PCO (Protocol Configuration Options) options in the network to MS GTP messages and configures APN to include in PCO IE.

Product

GGSN

P-GW

Privilege

Security Administrator, Administrator

Command Modes

Exec > Global Configuration > Context Configuration > APN Configuration

configure > **context** *context_name* > **apn** *apn_name*

Entering the above command sequence results in the following prompt:

`[context_name]host_name(config-apn)#`

Syntax Description

`[ no ] pco-options { custom5 [ ue-requested ] value` *`custom_value`* **`| value`** *`custom_value`* **`}`**

no

Removes PCO configuration at APN configuration mode

custom5

Enable sending of customized PCO options in the network to MS messages; send customized PCO options to all UEs regardless of support.

ue-requested

Enable sending of customized PCO options in the network to MS messages for "UE-Requested" mode; send PCO to only UEs that request customized PCO options.

Usage Guidelines

Use this command to enable or disable sending of customized PCO options in the network to MS GTP messages and configure link MTU size PCO value.

Example

The following command enables sending customized PCO options to all UEs regardless of support:

```
pco-options custom5
```

pdn-type

This command is used to configure the PDN type indicator in the APN profile.

Product

MME

Privilege

Security Administrator, Administrator

Command Modes

Exec > Global Configuration > APN Profile Configuration

configure > **apn-profile** *profile_name*

Entering the above command sequence results in the following prompt:

`[local]host_name(apn-profile-profile_name)#`

Syntax Description

pdn-type { ip | non-ip { sgi | t6a [scef-id *scef_id* **[scef-realm** *realm_name* **]] } }**
remove pdn-type

remove

The keyword remove deletes the existing configuration.

ip

Use this keyword to configure the Cellular IoT PDN type as IP PDN.

non-ip

Use this keyword to configure the Cellular IoT PDN type as Non-IP PDN.

sgi

Use this keyword to configure the Cellular IoT Non-IP PDN delivery path type as SGi.

t6a

Use this keyword to configure the Cellular IoT Non-IP PDN delivery path type as T6a.

scef-id *scef_id*

The user can optionally specify the SCEF ID using this keyword. The SCEF identifier is a string of length 1 up to 63 characters.

scef-realm *realm_name*

Use this keyword to optionally specify the SCEF diameter realm name. The *realm_name* is string of length 1 up to 127 characters.

Usage Guidelines

Use this command to specify the Cellular IoT PDN type. With this command the user has an option to override the HSS provided APN subscription PDN type. This command is applicable during Attach and additional PDN connectivity only and not during Handover scenarios. This command is not enabled by default.

Use the following command to configure the PDN type as Non-IP and the delivery path type as SGi:

pdn-type non-ip sgi

Use the following command to specify the PDN type as Non-IP and the delivery path as T6a along with the SCEF identifier and realm name:

pdn-type non-ip t6a scef-id *sc1* **scef-realm** *xyz.com*

pdp-type

Configures the type of PDP contexts that are supported by this APN.

Product

GGSN

P-GW

SAEGW

Privilege

Security Administrator, Administrator

Command Modes

Exec > Global Configuration > Context Configuration > APN Configuration

configure > **context** *context_name* > **apn** *apn_name*

Entering the above command sequence results in the following prompt:

```
[context_name]host_name(config-apn)#
```

Syntax Description

```
pdp-type { ipv4 [ ipv6 ] | ipv6 [ ipv4 ] | ppp | non-ip }
default pdp-type
```

default

Configures the default PDP type, IPv4, for the APN.

ipv4 [ipv6]

Enables support for IPv4 PDP contexts. Also enables support for IPv6 if the IPv6 optional keyword is entered in this command. Default: Enabled

Important Entering both IPv4 and IPv6 in either order enables support for both.

ipv6 [ipv4]

Enables support for IPv6 PDP contexts. Also enables support for IPv4 if the IPv6 optional keyword is entered in this command. Default: Disabled

Important Entering both IPv4 and IPv6 in either order enables support for both.

ppp

Enables support for PPP PDP contexts. Default: Disabled

non-ip

Enables support for Non-IP PDP Type for the APN.

Usage Guidelines

IP PDP context types are those in which the MS is communicating with a PDN such as the Internet or an intranet using IP. PPP PDP contexts are those in which PPP or PPP Network Control Protocol (NCP) frames from the MS are either terminated at, or forwarded by the GGSN.

If a session specifies a PDP type that is not supported by the APN, the system rejects the session with a cause code of 220 (DCH, Unknown PDP address or PDP type).

Caution For the IPv6 calls to work, the destination context must have at least one IPv6 interface configured.

Example

The following command configures the APN to support PPP context types:

```
pdp-type ppp
```

psm

This command is used to configure UE Power Saving Mode parameters.

Product

MME

Privilege

Security Administrator, Administrator

Command Modes

Exec > Global Configuration > Call Control Profile Configuration

configure > **call-control-profile** *profile_name*

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-call-control-profile-profile_name)#
```

Syntax Description

[remove] psm {ue-requested [dl-buf-duration [packet-count *packet_value* **]]| t3324-timeout** *t3324_value* **t3412-extended-timeout** *t3412_ext_value* **[dl-buf-duration [packet-count** *packet_value* **]]}**

remove

The **remove** keyword deletes the existing power saving mode configuration.

ue-requested

Use this keyword when UE requested values for Active and Extended Periodic timers are to be accepted.

t3324-timeout *t3324_value*

Use this keyword to configure the T3324 active timer value.

t3324_value

The T3324 active timer is an integer value in the range 0 up to 11160 seconds.

t3412-extended-timeout *t3412_ext_value*

Use this keyword to configure the t3412 Extended timer value.

t3412_ext_value

The T3412 extended timer is an integer value in the range 0 up to 35712000 seconds.

dl-buf-duration

Use this keyword to Send Downlink Buffer Duration in DDN ACK when unable to page UE.

packet-count *packet_value*

Use this keyword to send 'DL Buffering Suggested Packet Count' in DDN ACK when unable to page UE.

packet_value

The *packet_value* is an integer value from 0 up to 65535.

Usage Guidelines

Use this CLI command to configure the T3324 active and T3412 extended timers. The CLI also provides an option to either accept UE requested values or HSS subscribed values or MME configured values for these timers. This command is used to configure either to send or not send the Downlink Buffer Duration in DDN Ack, the DDN Ack Optional IE "Downlink Suggested Packet Count". The CLI option **dl-buf-duration [packet-count** *packet_value* **]** is used to optionally configure either to send or not send the downlink buffer duration in DDN Ack, the DDN Ack Optional IE "Downlink Suggested Packet Count" can also be configured. If this option is not configured and not sent in subscription, MME does not send IE in DDN reject. If the **packet-count** value is not configured locally, the subscription value for **packet-count** is used. The subscription value can be "0", in this case the packet count IE will not be sent for that subscriber even if it is configured locally. If the T3324 active and T3412 extended timers are locally configured these values are always used. If the **psm** command is configured to use the UE requested values for Active and Extended Periodic timers the UE requested values are accepted, but in case if the UE does not request T3412 extended timer, then the value available in subscription data are used for Extended Periodic timer. If the values are not available in the subscription data then the values configured under the MME service are used .

As per latest version of 3GPP TS 24.008, the maximum value of T3412 extended timer can be "320*31" hours that is "35712000" seconds. Due to MME constraints on timer implementation the T3412 extended timer is restricted to 1050 hours that is "3780000" seconds. However, the nearest usable value of this timer as 3GPP TS 24.008 GPRS Timer 3 is 960 hours (320 * 3) that is 3456000 seconds.

Example

Use the following command to enable power saving mode and to accept UE requested values for T3324 and T3412 timers.

```
psm ue-requested
```

Use the following command enable UE power saving mode and provide operator desired values for T3324 and T3412 timers:

```
psm t3324-timeout 100 t3412-extended-timout 5000
```

Use the following command to enable PSM and accept UE requested values for T3324 and T3412 timers. This command also specifies the 'DL Buffering Suggested Packet Count' in DDN ACK when unable to page UE.

```
psm ue-requested dl-buf-duration packet-count 100
```

In the following example, PSM is enabled and values of T3324 and T3412 timers are specified along with configuring a packet count in DDN ACK:

```
psm t3324-timeout 1000 t3412-extended-timeout 5000 dl-buf-duration
packet-count 100
```

require session ipsecmgr-per-vcpu

Configures the number of IP Security Manager (ipsecmgr) processes per vCPU.

Product

ePDG (VPC-DI platform only)

Privilege

Security Administrator, Administrator

Command Modes

Exec > Global Configuration

configure

Entering the above command sequence results in the following prompt:

`[local]host_name(config)#`

Syntax Description

[default] require session ipsecmgr-per-vcpu *count* **}**

default

Returns the number of ipsecmgrs per vCPU to the default of 1.

Note The default value can be adjusted as needed per your call model deployment requirements. Please contact your Sales or Support representative for more information.

count

Sets the number from 1 through 2 of the ipsecmgr processes to be created for each vCPU. Default: 1.

Usage Guidelines

Enables multiple IP Security Manager (ipsecmgr) processes per vCPU.

Example

The following command configures the system to create 2 ipsecmgrs per vCPU:

```
require session ipsecmgr-per-vcpu 2
```

require session sessmgr-per-vcpu

Configures the number of Session Manager (sessmgr) processes per vCPU.

Product

All (VPC-DI platform only)

Privilege

Security Administrator, Administrator

Command Modes

Exec > Global Configuration

configure

Entering the above command sequence results in the following prompt:

```
[local]host_name(config)#
```

Syntax Description

[default] require session sessmgr-per-vcpu *count* **}**

default

Returns the number of sessmgrs per vCPU to the default of 1.

Caution

The default value can be adjusted as needed per your call model deployment requirements. However, the recommendation is to use the default value. To change or adjust the default value, contact your Sales or Support representative.

count

Sets the number the sessmgr processes to be created for each vCPU. The valid values are 1 and 2 . The default value is 1.

Only for MME/SGSN, the count can go up to 2 for the number of sessmgrs per vCPU.

All other values are reserved.

Usage Guidelines

For applications that are light on CPU usage but heavy on RAM usage, such as Internet of Things (IoT) Gateway, it is more efficient to have multiple session manager (sessmgr) processes per vCPU.

Table 39: vCPU Support per Platform

Platform	vCPU Support
Gateway	• For 1 sessmgr process per vCPU, 16 sessmgr processes per Service Function (SF) VM are supported. • For 2 sessmgr processes per vCPU, 32 sessmgr processes per SF VM are supported.
MME/SGSN	For 2 sessmgr processes per vCPU, 56 sessmgr processes per SF VM are supported.

Example

The following command configures the system to create 2 sessmgrs per vCPU:

```
require session sessmgr-per-vcpu 2
```

scef-service

This command associates SCEF-service to the Call Control Profile.

Product

MME

Privilege

Administrator

Command Modes

Exec > Global Configuration > Call Control Profile Configuration

configure > call-control-profile *profile_name*

Entering the above command sequence results in the following prompt:

`[local]host_name(config-call-control-profile-profile_name)#`

Syntax

[remove] associate scef-service *service_name*

remove

This command prefix removes the SCEF association from Call Control Profile.

associate

This command associates the SCEF service with Call Control Profile.

scef-service*service_name*

This command associates SCEF with the call-control-profile, which is identified by a service name. The service name is a string which ranges from 1 to 63.

Usage Guidelines

Use this command to associate an SCEF service to the Call Control Profile for Non-IP Data Delivery (NIDD).

scef-service

This command associates SCEF-service to the MME Service.

Product

MME

Privilege

Administrator

Command Modes

Exec > Global Configuration > Context Configuration > MME Service

configure > context *context_name* **> mme-service** *service_name*

Entering the above command sequence results in the following prompt:

`[context_name]host_name(config-mme-service)#`

Syntax

[remove] associate scef-service *service_name*

remove

This command prefix removes the SCEF association from MME Service.

associate

This command associates the SCEF service with MME Service.

scef-service*service_name*

This command associates SCEF with the MME Service, which is identified by a service name. The service name is a string which ranges from 1 to 63.

Usage Guidelines

Use this command to associate an SCEF service to the MME Service for Non-IP Data Delivery (NIDD).

serving-plmn-rate-control

This command is used to configure the serving PLMN rate control for control plane CIoT optimization. The serving PLMN rate control limits the rate at which UE or PGW/SCEF can send data over the control plane when CP optimization is enabled.

Product

MME

Privilege

Security Administrator, Administrator

Command Modes

Exec > Global Configuration > Call Control Profile Configuration

configure > call-control-profile *profile_name*

Entering the above command sequence results in the following prompt:

`[local]host_name(config-call-control-profile-profile_name)#`

Syntax Description

serving-plmn-rate-control ul-rate *ul_rate_value* **dl-rate** *dl_rate_value*
remove serving-plmn-rate-control

remove

The keyword remove deletes the existing configuration.

ul-rate *ul_rate_value*

The maximum number of data NAS PDUs the UE can send in uplink path per deci-hour (6 minutes). The uplink rate is an integer from 10 up to 65535. A value of 65535 in this case implies no limit on the number of PDUs the UE can send in the uplink path per deci-hour.

dl-rate *dl_rate_value*

The maximum number of data NAS PDUs the PGW/SCEF can send in the downlink path to the UE per deci-hour (6 minutes). The downlink rate is an integer from 10 up to 65535. A value of 65535 in this case implies no limit on the number of PDUs the PGW/SCEF can send in the downlink path per deci-hour.

Usage Guidelines

This command configures serving PLMN rate for data over NAS. It limits the rate for data exchange between UE and the PGW/SCEF while using control plane CIoT optimization. This command is not enabled by default.

Example

Use the following command to configure the serving PLMN rate for data over NAS, with uplink rate as 35 and downlink rate as 45:

```
serving-plmn-rate-control ul-rate 35 dl-rate 45
```

show card

The output of the **show card table** and **show card information** commands were modified to reflect a new Slot Type.

Service Functions (SFs), Network Functions (NFs), and Application Functions (AFs) are now collectively represented as Function Cards and are grouped under a common Slot Type of "FC" in the output of these commands.

Previously, these cards were represented by the following slot types:

- Service Function: SFC
- Network Function: NFC
- Application Function: AFC

Note Thie Slot Type designator for the Control Function (CF) virtual card type ,which are always in slots 1 and 2, remains unchanged as "CFC".

The following are examples of the new **show card table** and **show card information** command output:

show card table Example

```
show card table

1: CFC      Control Function Virtual Card      Standby        -
2: CFC      Control Function Virtual Card      Active         No

3: FC  2-Port Service Function Virtual Card    Standby        -
8: FC  1-Port Network Function Virtual Card    Active         No
9: FC  1-Port Application Func. Virtual Card   Active         No
10:FC  1-Port Service Function Virtual Card    Standby        -
```

show card information Example

```
show card information 3
Card 3:
  Slot Type               : FC
  Card Type               : 2-Port Service Function Virtual Card
  Operational State       : Standby
  Desired Mode            : Standby
```

show cloud configuration

Displays the contents of the configuration file.

Product

All

Privilege

Security Administrator, Administrator, Inspector, Operator

Mode

Exec

The following prompt is displayed in the Exec mode:

```
[local]host_name#
```

Syntax

```
show cloud configuration
```

Usage

This command dumps the contents of the configuration file to the screen. It displays the configuration file on the config disk or the local flash. Usually the user does not have direct access to these files. The local param file on the flash is defined during the VPC installation and the config disk is usually created by the orchestrator and then attached to the card.

Example

This command displays the hardware configuration associated with card number 1:

```
show cloud configuration
```

show cloud hardware

Displays the hardware configuration for each card or a specific card.

Product

All

Privilege

Security Administrator, Administrator, Inspector, Operator

Mode

Exec

The following prompt is displayed in the Exec mode:

```
[local]host_name#
```

Syntax

show cloud hardware *card_number*

card_number

Specifies the number of the card for which to display information.

Usage

Displays the configuration of the underlying VM hardware for a specific card or all cards in the VPC. It provides information regarding the configured vCPU, memory size, huge page size, crypto hardware and the NIC.

Example

This command displays the hardware configuration associated with card number 1:

show cloud hardware 1

show cloud hardware optimum

Displays the optimum hardware configuration of the hardware parameters listed to achieve highest throughput.

Product All

Privilege Security Administrator, Administrator, Inspector, Operator

Mode

Exec

The following prompt is displayed in the Exec mode:

```
[local]host_name#
```

Syntax

show cloud hardware optimum

Usage

Displays the optimum configuration of the underlying VM hardware according to the available parameters. It provides information regarding the configured vCPU, memory size, huge page size, crypto hardware and the NIC.

Example

This command displays the optimum hardware configuration for the associated VM hardware:

show cloud hardware optimum

show cloud hardware test

Compares the current hardware configuration for each card or a specific card against the optimum settings.

Product

All

Privilege

Security Administrator, Administrator, Inspector, Operator

Mode

Exec

The following prompt is displayed in the Exec mode:

```
[local]host_name#
```

Syntax

show cloud hardware test *card_number*

card_number

Specifies the number of the card for which to display information.

Usage

Compares the configuration of the underlying VM hardware of a specific card or all cards in the VPC to the optimum configuration. It provides information regarding the configured vCPU, memory size, huge page size, crypto hardware and the NIC and indicates the optimum values for each parameter.

Example

This command displays the hardware configuration associated with card number 1:

show cloud hardware test 1

show cloud monitor

Displays VPC-DI network latency and packet loss statistics for all cards or a specific card in the VPC.

Product

All

Privilege

Security Administrator, Administrator, Inspector, Operator

Command Modes

Exec

The following prompt is displayed in the Exec mode:

```
[local]host_name#
```

Syntax Description

show cloud monitor di-network {**detail** | **summary**} *card_number*

detail

Displays detailed information about the VPC-DI network.

summary

Displays summary information about the VPC-DI network.

card_number

Specifies the number of the card for which to display information.

Usage Guidelines

Displays the configuration of the underlying VM hardware for a specific card or all cards in the VPC. It provides information regarding the configured vCPU, memory size, huge page size, crypto hardware and the NIC.

Example

This command displays summary monitored statistics for VPC-DI network communications from and to the third card in the VPC. The display shows the test packet loss rate for the past five minutes and past 60 minutes. If the rate is larger than 1%, the health status is marked as "Bad".

show cloud monitor di-network summary 3

```
Card 3 Test Results:

ToCard   Health   5MinLoss   60MinLoss
   1      Good       0.0%       0.0%
   2      Good       0.0%       0.0%
   4       Bad       6.32%      5.36%
   5      Good       0.0%       0.0%
   6      Good       0.0%       0.0%
```

show scef-service statistics

Displays SCEF Service configuration and status information.

Product

MME

Privilege

Security Administrator, Administrator, Operator, Inspector

Command Modes

Exec

The following prompt is displayed in the Exec mode:

```
[local]host_name#
```

Syntax Description

show scef-service statistics { all | name *service_name* **| summary }**

all

Displays all available configuration and status information for all SCEF Services.

name *service_name*

Displays all status information for a specified SCEF service name.

summary

Displays the summary of the available SCEF service statistics.

Usage Guidelines

Use this command to display SCEF service information and its statistics.

Example

The following command displays all SCEF service statistics:

```
show scef-service statistics all
```

The following command displays information for an SCEF service configuration with the service name *Test*:

```
show scef-service statistics name Test
```

Important

Output descriptions for commands are available in the *Statistics and Counters Reference*.

show system ssh key status

Displays the fingerprint of the current internal SSH key in use, the source of where the key was found, and the SSH status of all online VMs.

Product

VPC-DI

Privilege

Security Administrator, Administrator, Operator

Command Modes

Exec

The following prompt is displayed in the Exec mode:

```
[local]host_name#
```

Syntax Description

show system ssh key status [| { grep *grep_options* **| more }]**

| { grep *grep_options* | more }

Pipes (sends) the output of the command to the command specified. You must specify a command to which the output will be sent.

For details on the usage of the **grep** and **more** commands, refer to the *Regulating a Command's Output* section of the *Command Line Interface Overview* chapter.

Usage Guidelines

This command displays information about the SSH keys used for internal communication between all component VMs in a VPC-DI system, such as for remote command execution and file transfers.

system packet-dump

Initiates a packet dump on an SF or CF card in a VPC-DI system.

Product

All

Privilege

Security Administrator, Administrator

Command Modes

Exec

The following prompt is displayed in the Exec mode:

`[local]host_name#`

Syntax Description

system packet-dump { di-net card *slot_num* **| port** *service_port* **} [bond { a | b } | direction { both-rxtx | rx | rxtx | tx } | duration** *seconds* **| packet-type { ipv4 | ipv6 } | pcapfile-size** *size* **| pcapfile-split-val** *value* **| protocol { icmpv4 | icmpv6 | tcp | udp } | to file** *filename* **]**

di-net card *slot_num*

Specifies the card from 1 through *n*.

port *card_port/port_num*

Specifies the ethernet interface based on the card number from 1 through *n* and port number from 1 through 50, for example 3/1.

bond { a | b }

Specifies a slave for bonded interfaces.

direction { both-rxtx | rx | rxtx | tx }

Specifies a filter for the direction of the packets to capture, either receive (**rx**), transmit (**tx**), or both (**rxtx**). Use the **both-rxtx** option to capture both receive and transmit, but output each to separate files.

duration *seconds*

Specifies the number of seconds from 1 through 600 for the packet dump. Default: 5 seconds

packet-type { ipv4 | ipv6 }

Specifies a filter for the type of the packets to capture, either **ipv4** or **ipv6**.

pcapfile-size *size*

Specifies the maximum size for each packet capture (pcap) file from 10 to 800 megabytes. Default: 10 megabytes.

pcapfile-split-val *value*

Specifies the number of pcap files to generate for a given capture from 0 to 10. Default: 0 (do not split files).

protocol { icmpv4 | icmpv6 | tcp | udp }

Specifies a filter for the protocol of the packets to capture, either **icmpv4**, **icmpv6**, **tcp**, or **udp**.

to file { /flash | /hd-raid | /cdrom1 | /sftp } [/directory]/ *filename*

Specifies the output location and filename.

Usage Guidelines

Use this command to perform packet captures to troubleshoot issues within a VPC-DI deployment.

Example

The following command initiates a packet dump on card in slot 7, port 1, and output the dump to a file stored locally at /flash/example7-1.pcap

```
system packet-dump port 7/1 to file /flash/example7-1.pcap
```

system ping

Initiates a ping test on the internal network between two VMs within the VPC-DI system.

Product

VPC-DI

Privilege

Security Administrator, Administrator

Command Modes

Exec

The following prompt is displayed in the Exec mode:

```
[local]host_name#
```

Syntax Description

```
system ping from card slot_num to card slot_num [ count number_of_packets | size
 bytes ]
```

from card *slot_num*

Specifies the card slot number from 1 through *n* from which the ping test originates.

to card *slot_num*

Specifies the destination card slot number from 1 through *n*.

count *number_of_packets*

Sets the number of ping packets from 1 through 10000 to be sent. Default: 5 packets

size *bytes*

Sets the size of the ICMP Datagram in bytes from 40 to 18432. Default: 56

Usage Guidelines

Use this command to perform ping tests to troubleshoot connectivity issues within a VPC-DI deployment.

Example

The following command initiates a ping test of 1000 packets from the card in slot 1 to the card in slot 9:

```
system ping from card 1 to card 9 count 1000
```

system ssh

Manages the persistent ssh user keys used for the internal ssh sessions between cards (VMs) in a VPC-DI system.

Product

VPC-DI

Privilege

Security Administrator

Command Modes

Exec

The following prompt is displayed in the Exec mode:

```
[local]host_name#
```

Syntax Description

```
system ssh key { copy boot1 to card slot_num | create boot1 }
 no system ssh key boot1 { all | card slot_num }
```

no system ssh key boot1 { all | card *slot_num* }

Deletes the persistent ssh keys on a specific card or all cards in the VPC-DI system. Deletion of keys may be used to purge a VM of the persistent keys or prepare the system for using a different distribution method (ESC, OpenStack, attached ISO).

- **all** : Deletes the ssh keys on all cards in the VPC-DI system.
- **card** *slot_num* : Deletes the ssh keys on the card specified by *slot_num* .

Note

This command does not affect the VM until it is rebooted. It will continue to use the active key found during its boot.

copy boot1 to card *slot_num*

Transfers the persistent ssh keys (both public and private) in /boot1 on the active CF to another VM. That VM must be in a state to accept it by a user with console access placing it in receiver mode during its failed boot.

create boot1

Creates new persistent ssh keys (both public and private) and stores it in /boot1 on the active CF.

Note This command does not affect the VM until it is rebooted. It will continue to use the active key found during its boot.

Usage Guidelines

Use this command to manage the internal ssh keypairs in a VPC-DI deployment. While StarOS provides sshd services for user CLI and SFTP sessions on the management VMs (CF), another set of sshd services run for the exlusive use of internal communication between all component VMs, such as for remote command execution and file transfers. This internal sshd is only used on the internal DI-network interface.

This command enables you to store and manage ssh keys on the VM's virtual hard disk drive (HDD). This provides an alternate option for storing ssh keypairs besides the other methods such as Cisco Elastic Services Controller (ESC), OpenStack, or a directly attached ISO. The /boot1 partition is only accessible by a security administrator.

Use the **show system ssh key status** command to display the fingerprint of the current public key in use, the origin of where the key was found, and the status of all online VMs.

Example

The following command copies the ssh keypairs from the active CF to the card in slot 12

```
system ssh key copy boot1 to card 12
```

tunnel udpip

Configures UDP-IPv4 or UDP-IPv6 tunneling parameters between the P-GW and an external application server for the APN.

Product

P-GW

S-GW

Privilege

Security Administrator, Administrator

Command Modes

Exec > Global Configuration > Context Configuration > APN Configuration

configure > **context** *context_name* > **apn** *apn_name*

Entering the above command sequence results in the following prompt:

```
[context_name]host_name(config-apn)#
```

Syntax Description

tunnel udpip peer-address *peer_address* **peer-port** *peer_udp_port* **[local-port** *local_udp_port* **]**
no tunnel udpip

no

Disables UDP-IPv4 or UDP-IPv6 tunneling for the APN.

peer-address *peer_address*

Specifies the Peer address for the tunnel.

peer_address must be expressed in dotted-decimal notation.

peer-port *peer_udp_port*

Specifies the port number of the peer for the tunnel.

peer_udp_port must be expressed in dotted-decimal notation.

local-port *local_udp_port*

Specifies the local UDP port number.

Default: 49152

Usage Guidelines

For local and peer UDP port number, it is recommended to use unregistered port number with IANA.

This CLI command takes effect during new subscriber call creation on S5/S8 interface to the APN.

Example

The following command configures the system to encapsulate subscriber traffic using UDP-IPv4 and tunnel it from a locally assigned IP address with port number *49152* to an external application server with an IP address of *192.168.1.100* on peer UDP port *11220*:

```
tunnel udpip peer-address 192.168.1.100 peer-port 11220 local-port 49152
```