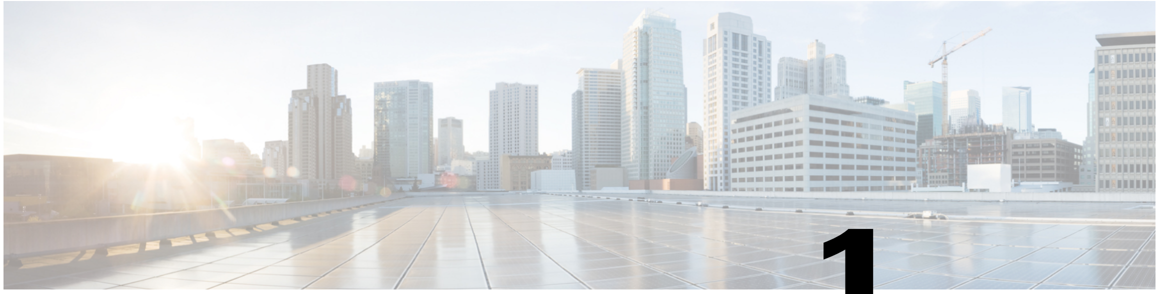
Configuration Guide for AppNav-XE on the Cisco CSR 1000V Series and Cisco ASR 1000 Series

First Published: 2017-01-10

Last Modified: --

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



CHAPTER 1

Preface

This preface describes the audience and conventions of the *Configuration Guide for AppNav-XE on Cisco CSR 1000V Series and Cisco ASR 1000 Series* . It also describes the available product documentation and provides information on how to obtain documentation and technical assistance.

Audience

This guide is intended primarily for network administrators, system administrators, and system integrators.

Conventions

This document uses the following conventions:

Convention	Item
boldface font	Commands and keywords.
italic font	Variables for which you supply values.
[enclosed in brackets]	Optional command keywords. You do not have to select any options.

Convention	Item
{options enclosed in braces | separated by vertical bar}	Required command keyword to be selected from a set of options. You must choose one option.
`screen` `font`	Displayed session and system information.

`boldface screen`
`font`
	Information you enter.

italic screen `font`	Variables you enter.
Option > Network Preferences	Choosing a menu item.

Note Means *reader take note* .

Caution Means *reader be careful* . In this situation, you might perform an action that could result in equipment damage or loss of data.

Related Documentation

The following related documentation is available on Cisco.com:

- Cisco Wide Area Application Services Configuration Guide
- Cisco Cloud Services Router 1000V Series documents available at http://www.cisco.com/en/US/products/ps12559/tsd_products_support_series_home.html
- Cisco ASR 1000 Series Aggregation Services Routers documents available at http://www.cisco.com/en/US/partner/products/ps9343/tsd_products_support_series_home.html

Obtaining Documentation and Submitting a Service Request

For information on obtaining documentation, submitting a service request, and gathering additional information, see the monthly *What's New in Cisco Product Documentation* , which also lists all new and revised Cisco technical documentation, at:

http://www.cisco.com/en/US/docs/general/whatsnew/whatsnew.html

Subscribe to the *What's New in Cisco Product Documentation* as a Really Simple Syndication (RSS) feed and set content to be delivered directly to your desktop using a reader application. The RSS feeds are a free service and Cisco currently supports RSS version 2.0.

CHAPTER 2

Overview of the AppNav-XE Solution

This guide provides an overview of the AppNav-XE component on the Cisco Cloud Services Router (CSR) 1000V Series in Cisco IOS-XE Release 3.9, and describes the quick start process to easily configure the features. It also provides details of the command line interface (CLI) commands along with examples and troubleshooting tips.

Overview of the AppNav-XE Solution on the Cisco Cloud Services Router 1000V Series

The AppNav-XE solution for the Cisco Cloud Services Router 1000V Series includes the following:

- AppNav Controller: Component that intelligently distributes traffic from a router to services.
- AppNav service node auto discovery feature: Feature that automatically discovers service nodes and adds them to an AppNav cluster. See the About the AppNav Service Node Auto Discovery Feature (For Cisco CSR 1000V Series Only), on page 6.

Note The WAAS service nodes are external to the Cisco CSR and their details are outside the scope of this document.

- WAAS Central Manager (WCM): Used to monitor and configure the AppNav-XE component.

Note You can also use WCM to configure and monitor the WAAS service nodes but those details are outside the scope of this document.

Overview of the AppNav-XE Solution on the Cisco ASR 1000 Series Aggregation Services Router

The AppNav-XE solution for the Cisco ASR 1000 Series Aggregation Services Router (called Cisco ASR 1000 Series) includes the following:

- AppNav Controller: Component that intelligently distributes traffic from a router to services.

Note The WAAS service nodes are external to the Cisco ASR 1000 Series and their details are outside the scope of this document.

- WAAS Central Manager (WCM): Used to monitor and configure the AppNav-XE component.

Note You can also use WCM to configure and monitor the WAAS service nodes but those details are outside the scope of this document.

AppNav-XE Component Overview

The AppNav-XE component is made up of a distribution unit called the AppNav Controller and service nodes. The AppNav Controller distributes flows and the service nodes process the flows. Additionally up to four AppNav Controllers can be grouped together to form an AppNav Controller group to support asymmetric flows and high availability. Note that all the routers in the AppNav Controller group need to be the same platform and also have the same memory capacity.

Advantage of Using the AppNav-XE Component

The advantages of using the AppNav-XE component are:

- It can intelligently redirect new flows based on the load on each service node. This includes loads of individual L7 application accelerators.
- For flows that do not require any optimization, service nodes can inform the AppNav Controller to directly pass-through the packets, thereby minimizing the latency and resource utilization.
- There is minimal impact to traffic when adding or removing service nodes.
- The AppNav-XE component supports VRF so that VRF information is preserved when traffic returns from a service node.

- For special applications such as MAPI (Exchange) and VDI (Citrix), the AppNav-XE component ensures that flows from the same client and destined to the same server and server port are redirected to the same service node.
- You can use an AppNav Controller group to optimize asymmetric flows. An asymmetric flow is when the traffic in one direction goes through one AppNav Controller and the return traffic goes through a different AppNav Controller, but both AppNav Controllers redirect the traffic to the same service node.
- Inter-router high availability, where if one router goes down, the traffic can be re-routed to a different router within the AppNav Controller group, keeping the traffic flows uninterrupted.
- Intra-router high availability of the AppNav Controller on Cisco ASR 1000 Series platforms that have dual RP or dual FP. This means that if the active RP fails, the standby RP takes over or if the active FP fails, the standby FP takes over and the flows continue uninterrupted.The intra-router high availability feature is available only on the Cisco ASR 1000 Series platforms.

Interoperability of the AppNav-XE Component

The AppNav-XE component can interoperate with the following features on the router:

- QoS
- NAT (Note that the video application accelerator is disabled and that asymmetric routing and inter-router high availability handled both by the AppNav-XE component and NAT is not supported.)
- AVC 2.0 (FNF, NBAR) (Note that AVC 2.0 does not support asymmetric routing and inter-router high availability.)
- IPSec
- GET-VPN (ASR 1000 Series only)
- EzVPN
- DMVPN
- ACL
- VRF
- MPLS (The supported topology is an MPLS network on the WAN side and an IP network on the LAN side.)
- WCCP-AppNav-XE coexistence (WCCP and AppNav-XE can be configured on the same interface only if they act on different flows. Use ACLs for this. WCCP and AppNav XE can be configured on different interfaces—AppNav-XE on WAN and WCCP on LAN.)
- BPBR/PFR (supported on Cisco IOS releases 3.10.1 or later.

The AppNav-XE component introduces the concept of a virtual interface, which allows users to configure features specific to compressed or uncompressed traffic. For instance, to monitor the traffic that is being redirected to the service node and the traffic that is returning from the service node, you can configure the FNF feature on the AppNav-UnCompress and AppNav-Compress virtual interfaces. Note that these AppNav-XE virtual interfaces appear to the user just as any other interface. However from the above list, the only features that work on the AppNav-XE virtual interfaces are FNF, ACL, and QoS (except for queueing).

About Configuring the AppNav-XE Component

Note the following points regarding configuring the AppNav-XE component:

- You must identify the WAN interfaces for the router that is running the AppNav Controller. The AppNav Controller intercepts packets on both ingress and egress of WAN interface. Only configure the AppNav Controller on WAN interfaces, including all WAN interfaces that will be load balancing.
- Do not use the VRF to access the service node from the AppNav Controller. Neither the service node nor the AppNav Controller IP address should have VRF on the AppNav Controller.
- You can use port channel between the AppNav Controller and the service nodes to increase AppNav Controller-service node bandwidth.
- The config replace command cannot be used with AppNav-XE configuration.
- If you use an AppNav Controller group with two or more AppNav Controllers, the AppNav-XE configuration on all the AppNav Controllers must be the same. This also means that the names of the AppNav policy maps and class maps on the AppNav Controllers need to match. Also the VRF names for the traffic seen by the AppNav-XE component need to be the same on all the AppNav Controllers.
- If AppNav-XE is managed by WCM, the authentication key in the service-context configuration cannot be modified using the command line interface (CLI).

For additional information and caveats about configuring the AppNav-XE component, see Chapter 1, “Detailed Configuration” .

About the AppNav Service Node Auto Discovery Feature (For Cisco CSR 1000V Series Only)

The AppNav service node auto discovery feature is targeted for small branch installations. With this feature, the system automatically discovers the service nodes within the same L2 connectivity of the AppNav-XE router and adds them to the service node cluster.

Restriction

The AppNav service node auto discovery feature can only be enabled on one interface on a service node.

To enable the AppNav service node auto discovery feature, do the following:

SUMMARY STEPS

1. Initiate a discovery request on the AppNav-XE component on router by doing the following:
2. Initiate a service respond on the service nodes by doing the following:

DETAILED STEPS

Step 1 Initiate a discovery request on the AppNav-XE component on router by doing the following:

a) Determine the service node group for which you want to enable the auto discovery.

b) Issue the following commands:

Example:

```
router(config)# service-insertion service-node-group sng
router(config-service-insertion-sng)# node-discovery enable
```

Step 2 Initiate a service respond on the service nodes by doing the following:

a) On the WAAS appliance, determine the interface for which you want to enable node discovery. This interface must be in the same subnet as the AppNav Controller.

b) Enable node discovery by issuing the following commands:

Example:

```
auto-sn(config)# service-insertion service-node
auto-sn(config-sn)# node-discovery enable GigabitEthernet 0/1
auto-sn(config-sn)# enable
```

Licensing Requirements for the Cisco CSR 1000V Series

The AppNav feature will be available in the premium package, but in the Cisco IOS-XE 3.9 release, it is not enforced.

Licensing Requirements for the Cisco ASR 1000 Series

The AppNav feature will be available in the Advanced IP Services (AIS) and Adventerprise (AES) packages, but in the Cisco IOS-XE release 3.9, it is not enforced.

CHAPTER 3

Quick Start for Data Center and Cloud Users

The following is an overview of the steps needed to configure an AppNav Controller and cluster. Perform these steps in order:

1. Preparing to Configure an AppNav Cluster, on page 9
2. Registering an AppNav Controller, on page 10
3. Creating an AppNav Cluster, on page 10
4. Configuring an AppNav Controller, on page 12

You can also monitor an AppNav cluster at any time. See the Monitoring an AppNav Cluster, on page 12. To learn about the reports available, see the AppNav-XE Reports, on page 13.

Preparing to Configure an AppNav Cluster

Before you begin, install and configure the individual AppNav Controllers and service nodes with basic network settings. Provision VRFs and network interfaces on each AppNav Controller. See the following information:

- To install and configure the Cisco CSR 1000V Series, see http://www.cisco.com/en/US/products/ps12559/tsd_products_support_series_home.html
- To install and configure the Cisco ASR 1000 Series, see http://www.cisco.com/en/US/partner/products/ps9343/prod_installation_guides_list.html .
- To install and configure the service nodes, see http://www.cisco.com/en/US/partner/products/ps6870/prod_installation_guides_list.html

Registering an AppNav Controller

Before you can use the WCM to manage the AppNav-XE features on the AppNav Controller and service nodes, you must register the AppNav Controller with the WCM.

Registering an AppNav Controller Using the WCM GUI

When you use the WCM GUI to register the AppNav Controller. the WCM creates an account record for the AppNav Controller and populates it with the information you enter.

Before You Begin

- Enable SSH on the AppNav Controller.

Procedure

SUMMARY STEPS

1. In the WCM GUI, navigate to the **Home > Admin > Registration > Cisco IOS Routers** page.
2. Enter the required information.
3. Click **Register**.

DETAILED STEPS

Step 1 In the WCM GUI, navigate to the **Home > Admin > Registration > Cisco IOS Routers** page.

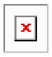
Step 2 Enter the required information.

Note For user name and password, enter the AppNav Controller credentials. The system uses these credentials one time only to retrieve configuration data, statistics, and status information.

Step 3 Click **Register**.

Viewing the AppNav Controller Registration

After you have successfully registered the AppNav Controller with the WCM, the WCM GUI displays the AppNav Controller under the devices list as “AppNav-XE Controller.” Navigate to **Devices > Device List**. See the following figure.



Creating an AppNav Cluster

Use the WCM AppNav Cluster creation wizard to create and configure an AppNav cluster. Configure settings, choose cluster devices, choose VRFs, configure the service insertion of the WAN-facing interface, and configure the cluster traffic interfaces. The wizard provides a single guided end-to-end configuration flow to the user.

Procedure

SUMMARY STEPS

1. Launch the WCM AppNav Cluster creation wizard by clicking **AppNav Cluster Wizard** in the WCM GUI.
2. (Optional) Configure AppNav class maps if you want to customize the default class map configuration. The system adds several default class maps that match traffic corresponding to most of the application accelerators and a class default class map that matches all traffic. See the "Configuring AppNav Class Maps" section on page 3 .
3. (Optional) Configure an AppNav policy if you want to customize the default policy. The system adds a default policy that distributes all traffic to the WNG-Default-nn WNG, which is the node group into which all service nodes are grouped by default. See the "Configuring AppNav Policy Maps" section on page 4 .
4. (Optional) Configure service node optimization class maps and policy rules. This step is necessary only if you want to customize the default optimization policy that is listed in Predefined Optimization Policy. See the Cisco Wide Area Application Services Configuration Guide for information.
5. Click **AppNav Clusters** to see all the configured clusters. To view the AppNav cluster settings at the AppNav Controller level, go to the device-level AppNav cluster page.

DETAILED STEPS

Step 1 Launch the WCM AppNav Cluster creation wizard by clicking **AppNav Cluster Wizard** in the WCM GUI.

Step 2 (Optional) Configure AppNav class maps if you want to customize the default class map configuration. The system adds several default class maps that match traffic corresponding to most of the application accelerators and a class default class map that matches all traffic. See the "Configuring AppNav Class Maps" section on page 3 .

Step 3 (Optional) Configure an AppNav policy if you want to customize the default policy. The system adds a default policy that distributes all traffic to the WNG-Default-nn WNG, which is the node group into which all service nodes are grouped by default. See the "Configuring AppNav Policy Maps" section on page 4 .

Step 4 (Optional) Configure service node optimization class maps and policy rules. This step is necessary only if you want to customize the default optimization policy that is listed in Predefined Optimization Policy. See the Cisco Wide Area Application Services Configuration Guide for information.

Step 5 Click **AppNav Clusters** to see all the configured clusters. To view the AppNav cluster settings at the AppNav Controller level, go to the device-level AppNav cluster page.

Note You cannot configure AppNav cluster settings at the device level as this page is read-only. The only operation you can do on this page is to remove the settings if a device has its own cluster settings. The only case when this page is editable is when it is displayed at WAAS Node device level and WAAS Node is participating in a cluster created using the CLI.

Configuring an AppNav Controller

To change the configuration of the AppNav cluster, in the WCM GUI navigate to the AppNav Cluster Home page. This page automatically presents live information collected from the AppNav Controller and service nodes.

You can perform the following tasks from the AppNav Cluster Home page.

- Configuring WAAS device interfaces.
- Creating a new AppNav context (WAAS cluster ID) with the AppNav context wizard.
- Configuring AppNav context settings like changing AppNav policy assignment and VRF associations.
- Adding new AppNav Controllers.
- Configuring AppNav Controllers, including changing service insertion on WAN interfaces and cluster traffic interfaces.
- Configuring AppNav cluster settings such as the authentication key.
- Adding service nodes.
- Configuring service node settings including changing service node groups.
- Configuring service node group settings including AppNav service node auto discovery (for Cisco CSR 1000V Series only).
- Adding and removing devices or AppNav contexts from the AppNav cluster.

Monitoring an AppNav Cluster

About Monitoring an AppNav Cluster

After you create an AppNav cluster, the system displays the topology representation along with live information from the cluster on the cluster home page. By default, the system collects data from every AppNav Controller every 300 seconds and presents the live information and statuses in the GUI. The system refreshes the cluster home page whenever the cluster changes. When you are viewing the topology page, data collection happens every 30 seconds.

Cluster State Representation

The system displays the link between the AppNav Controller and the service nodes in different colors to indicate their availability:

- Operational: When the AppNav cluster is fully operational, the system displays it in green.
- Degraded: When the AppNav cluster is degraded, the system displays it in yellow. For example, when some of the accelerators are not fully available (may be disabled), the AppNav cluster is degraded. Use the 360 degree view and the service node status information to identify the exact issue.

- Down: When the AppNav cluster is down, the system displays it in red. For example, when the only available service context in the cluster is disabled, the cluster status changes to down.

360 Degree View Representation for the AppNav Controller

Every AppNav Controller in the AppNav cluster has a 360 degree view that contains additional information about the AppNav Controller. To see the 360 degree view, hover over or click the AppNav Controller.

The AppNav Controller 360 degree view shows the following information:

- Cluster Control: The liveliness status of all the AppNav Controllers and service nodes. Includes the device name and its IP address.
- Interception: The interfaces where service insertion is configured and their statuses.
- Interfaces: All the available interfaces and their overall statuses.
- Overloaded Policies: The monitored accelerators that are overloaded, along with their corresponding policies. This is only displayed when there is an overload.
- Alarms: Alarm information from the AppNav Controller.

360 Degree View Representation for the Service Node

Every service node in the AppNav cluster has a 360 degree view that contains additional information about the service node. To see the 360 degree view, hover over or click the service node.

The service node 360 degree view shows the following information:

- Optimization: Displays the name, status, and description of all application accelerators in the service node.
- Alarms: Alarm information from the service node.
- Interfaces: All the available interfaces and their overall statuses.

AppNav-XE Reports

The WCM provides reports at the AppNav Controller and cluster level. These reports can be seen at the service context level or at a combined level, which includes all service contexts. The following are the different reports:

- Total AppNav Traffic: This chart represents the distributed, pass-through traffic as reported from all the class maps configured in the cluster.
- Top 10 AppNav Policies: This chart represents the intercepted, distributed, pass-through traffic for the top 10 class maps in the cluster.
- AppNav Policies: This chart represents the intercepted, distributed, pass-through traffic for class maps over time.
- Top 10 WAAS Node-Group Distribution: This chart represents the distribution of traffic across the top ten service node groups.

- Was Node-Group Distribution: This chart represents the distribution of traffic over time across the service node groups.
- Top 10 Pass-Through Reasons: This chart represents the top ten pass-through reasons that contributed to pass-through traffic.
- Pass-Through Reasons: This chart represents the pass-through reasons over time.

CHAPTER 4

Detailed Configuration

This chapter describes the detailed configuration for AppNav-XE and contains the following sections:

Configuring the AppNav Controller

To configure the AppNav Controller, follow these procedures:

Configuring AppNav Controller Groups

The AppNav Controller group configures the AppNav Controller. To configure the AppNav Controller group, enter the IP addresses used by the AppNav Controllers.

Restrictions

- The AppNav Controller group must always contain exactly one local IP address. This is the IP address of the local AppNav Controller (the local router). Note that this local IP address must belong to an interface from which all the other AppNav Controllers in the AppNav Controller group and all the service nodes are reachable.
- The AppNav Controller group cannot have more than four AppNav Controllers. This must include exactly one local IP address and optionally up to three non-local IP addresses.
- You can use the IP address from GigE, the VLAN interface, the loopback interface etc., but the interface must not have VRF configured.
- The system only supports configuration of one AppNav Controller group.

Use the following command:

```
(config)# [no] service-insertion appnav-controller-group
group-name
```

Submode command:

```
(config-service-insertion-acg)# [no] appnav-controller
IP_address
```

Optional command:

```
(config-service-insertion-acg)# [no] description
 group_description
```

Configuring a Reload Delay

(This feature is available in Cisco IOS XE release 3.10.2.)

Within an AppNav Controller Group (ACG), when a router has just rebooted, it is important to provide a brief delay before designating the router as active and handing over traffic. The delay enables the rebooted router to synchronize flows with the router currently handling traffic. The synchronization helps to avoid unintentionally resetting connections.

During the delay:

- Another AppNav controller in the ACG handles the traffic.
- On the affected device, AppNav enters a mode in which it drops all TCP packets, including border gateway protocol (BGP) packets.
- On the affected device, AppNav drops EIGRP and OSPF.

Command

Use the following command configure the delay:

```
(config-service-insertion-acg)# [no] service-insertion acg-reload-delay
 [
120-450
]
```

The default delay is 120 seconds. Typically, this is sufficient time for synchronizing flows.

The **no** form of the command cancels the delay upon next reload. Executing the **no** form of the command does not cancel the current delay if already applied.

Note Adding the command to the startup-config batch file ensures that the delay is configured during the reboot process.

Requirements for Use

The delay feature is intended for use in the following scenario:

- AppNav Controller Group configured with multiple controllers (see Configuring AppNav Controller Groups, on page 15)
- At least one AppNav service context enabled

- Router has just rebooted

Configuring Service Node Groups

You must configure a service node under a service node group. The AppNav-XE component intelligently distributes flows to the service node within the service node group.

Beginning with the Cisco IOS XE 3.13 release, a total of 64 service nodes may be included in a cluster. (Earlier releases permitted 32.)

Restriction

You cannot use VRF with either the AppNav Controller or the service node IP address. The IP addresses must be explicitly accessible without VRF. For example, you cannot use the management interface's IP address (with vrf Mgmt-intf) as the AppNav Controller IP address.

Use the following command:

```
(config)# [no] service-insertion service-node-group
 group_name
```

Submode commands:

```
(config-service-insertion-sng)# [no] description
 group_description
(config-service-insertion-sng)# [no] service-node
 IP_address
```

Configuring AppNav Class Maps

Use AppNav classes to determine which traffic should be handled by the AppNav-XE component. Use the appnav type class-map to classify the traffic based on the following set of parameters:

- Access list
- Service node peer device ID
- Special protocols supported by the service node

Table 1: ACL and ACE Platform Limits , on page 17 lists the ACL and ACE platform limits for each of the ASR and CSR platforms.

Table 1: ACL and ACE Platform Limits

Platforms	ACLs	ACEs (IPv4)	ACEs per ACL(IPv4)	ACEs (IPv6)	ACEs per ACL(IPv6)
ASR1K ESP-5	4K	25K	15K	8K	4K
ASR1K ESP-10	4K	50K	30K	15K	8K
ASR1K ESP-20/40	4K	100K	60K	30K	16K

Platforms	ACLs	ACEs (IPv4)	ACEs per ACL(IPv4)	ACEs (IPv6)	ACEs per ACL(IPv6)
ASR1K ESP-80/60	4K	400K	80K	200K	8K
CSR 1000V	350 IPv4 or 175 IPv6	1K	550	550	275

To create or modify a class map to be used for matching connections to a specified class, use the **class-map** command in global configuration mode. To remove an existing class map, use the **no** form of this command. The **class-map** command enters class-map configuration mode in which you can enter an optional description command and one or more of the match commands to configure the match criteria for this class.

The syntax for defining a class map is as shown below:

```
(config)# [no] class-map type appnav [match-all | match-any]
appnav_class_name
```

If you do not specify a match, the default is match-all.

Submode commands:

```
(config-cmap)# [no] description
 description_text
(config-cmap)# [no] match access-group
{ACL_number
 | name
ACL_name}
(config-cmap)# [no] match peer
device_ID
(config-cmap)# [no] match protocol
 app_def
```

Match Access-Group Command

The **match access-group** command specifies a numbered access-list or named access list whose contents are used as the match criteria against packets to determine if they belong to this class. The Access List (ACL) number can range from 1 to 2699.

Match Peer Command

The **match peer** command identifies a peer service node that may be performing optimization at the client side of a connection and must be specified in 01:23:45:67:89:ab format. The match peer clause is only useful if the AppNav-XE component is acting as core, that is, receiving a connection that has already been through a peer WAAS device.

Match Protocol Command

The **match protocol** command gets one of the following protocols:

- CITRIX
- MAPI
- MS-AD-REP
- MS-EXCH-NSPI

- MS-FRS
- MS-FRSAPI
- MS-RFR
- MS-SQL
- MSN-MESSENGER
- NETLOGON

The protocol is only used along with additional information provided by the service node to associate the packet with specific applications. The match protocol filter should not be confused with the **monitor-load** keyword in AppNav policy described below.

Configuring AppNav Policy Maps

After you configure the AppNav class maps, you can assign actions to them by using an AppNav policy map.

Limits for AppNav Policy Maps, Class maps, and Match Filters Per Class

Table 2: Limits for AppNav Policy Maps, Class maps, and Match Filters Per Class , on page 19 lists the limits for AppNav policy maps, class maps, and match filters per class.

Table 2: Limits for AppNav Policy Maps, Class maps, and Match Filters Per Class

Policy/Class/Filter Capacity	ASR 1000	CSR 1000V
Unique policy maps	4096 (16000 from Cisco IOS-XE Release 3.10 for RP2, ESP40, ESP100, ESP200 models only)	30
Unique class maps	4096	256
Number of classes per policy map	1000	32
Number of filters per class map	32	8

To create or modify a policy map that defines the service policy for the candidate optimization traffic, use the **policy-map** command in global configuration mode.

```
(config)# [no] policy-map type appnav
 appnav_policy_name
```

Submode commands:

```
(config-pmap)# [no] description
 description_text
(config-pmap)# [no] class
appnav_class_name
```

The **class** command above enters the policy-map-class configuration submode:

```
(config-pmap-c)# [no] distribute service-node-group
```

```
SNG_name
(config-pmap-c)# [no] monitor-load

application_accelerator_name
(config-pmap-c)# [no] pass-through
```

Distribute Command

The **distribute** command is the most common action in this class. The system sends the traffic that matches the class map to the service node group identified by the specified *SNG_name* parameter. If no service node group is available, or if no distribute is specified, the default action is to pass-through the traffic.

To configure primary and backup service node groups, use two **distribute** command statements:

```
(config-pmap-c)# distribute service-node-group

primary_SNG_name
(config-pmap-c)# distribute service-node-group

backup_SNG_name
```

If the service nodes in the primary service node group are not available, the system will use the backup service node group.

Monitor-Load Command

The **monitor-load** command determines which load values should be monitored. When you monitor an application accelerator, the AppNav Controller checks for overload on that application accelerator and does not send new flows to a service node that is overloaded. Flows are sent to a different service node in the service node group.

This command is optional; if you use it, the system monitors the application accelerator indicated by the *application_accelerator_name* parameter. If you do not use this command, the system monitors the TFO accelerator status. If you specify an application accelerator, it replaces the existing monitor-load if one exists.

The supported application accelerators are:

- MS-port-mapper (monitor Microsoft Endpoint Port Mapper load)
- cifs (monitor SMB or CIFS accelerator load)
- http (monitor HTTP accelerator load)
- ica (monitor ICA accelerator load)
- mapi (monitor MAPI accelerator load)
- nfs (monitor NFS accelerator load)
- ssl (monitor SSL accelerator load)
- video (monitor video accelerator load)

Pass-Through Command

Use the **pass-through** command to explicitly indicate that no redirection is to take place. You cannot use the **pass-through** command with the **distribute** or **monitor-load** commands. If you use the **pass-through** command, the system blocks any **distribute** or the **monitor-load** command actions and displays an error message. If you use either the **distribute** or the **monitor-load** command, then the system blocks any **pass-through** command actions.

Configuring Service Contexts

A service context is used to tie the AppNav Controller group, service node group, and AppNav policy map together.

Note If AppNav-XE is managed by WCM, the authentication key in the service-context configuration cannot be modified using the command line interface (CLI)

Use the following command to create a service context:

```
(config)# service-insertion service-context waas/
interface_ID
```

interface_ID is a number that is unique across all service contexts. It determines the naming of the automatically-created virtual interfaces called AppNav-Compress*interface_ID* and AppNav-UnCompress*interface_ID* .

Submode commands:

```
(config-service-insertion-context)# [no] appnav-controller-group
 ACG_name
(config-service-insertion-context)# [no] authentication sha1 key
 authentication_key
(config-service-insertion-context)# [no] service-node-group
 SNG_name
(config-service-insertion-context)# [no] service-policy
appnav_policy_name
(config-service-insertion-context)# [no] vrf { name
VRF_name
 | default | global}
(config-service-insertion-context)# [no] enable
```

AppNav Controller Group Command

ACG_name is the name of the AppNav Controller group to which this service context belongs. You can only configure one AppNav Controller group for each service context.

Authentication SHA1 Key Command

authentication-key is the shared authentication key used during AppNav Controller to service node registration. You must configure the key identically on service nodes in the same service context. Currently, the AppNav Controller group only supports one authentication key. All service contexts must use authentication or no service contexts can use authentication.

Service Node Group Command

SNG_name is the name of one or more service node groups that are part of the service context. The list is used to cross check the ones used in the AppNav policy. Note that the same service node group cannot be shared between two service contexts.

Service Policy Command

appnav_policy_name is the name of the AppNav policy for the service context.

VRF Name Command

VRF_name is the name of the VRF on the LAN interface for the traffic seen by the AppNav-XE component. You can enter more than one VRF name. You can define up to 64 VRF names, but there is no limit to the number of VRFs supported. VRF global is the same as the other VRF definitions except that it identifies traffic with no VRF. The VRF names are listed one after another such as the following:

```
        vrf name v1
        vrf name v2
        vrf name v3
        vrf global
```

If you do not configure a VRF in the service context, the system automatically applies the default configuration of vrf default. The purpose of vrf default is to match traffic that does not match a configured VRF name or vrf global.

The following logic is used to pick the right service context for a packet: The system compares the VRF on the LAN interface traversed by the packet against the VRF names (or vrf global) that is configured in the service contexts. If there is a match, the system picks the corresponding service context. If there is no match, the system picks a service context with vrf default, if available. If there is no such service context, then the system passes through the packet.

Enabling AppNav Interception

Currently, the only service supported by the AppNav-XE component is WAAS.

To enable the AppNav-XE component, identify your WAN interface and then use the **service-insertion** command.

```
(config)# interface
 if_name
(config-if)# [no] service-insertion waas
```

Note Both the incoming and outgoing TCP traffic of the interface are subject to AppNav processing according to their VRF and the service policy associated with the service context identified by the VRF.

Configuring the AppNav Service Node Auto Discovery Feature (For Cisco CSR 1000V Series Only)

This section contains the following subsections:

Enabling the AppNav Service Node Auto Discovery Feature (For Cisco CSR 1000V Series Only)

To configure the AppNav service node auto discovery feature, perform the following steps:

Procedure

SUMMARY STEPS

1. In Cisco IOS-XE, enter the following command. For the *SNG_name* parameter, enter the name of the service node group for which you want to enable the AppNav service node auto discovery feature. Ensure that the WAAS device is in the same subnet as the AppNav-XE component.
2. Enable the feature by entering the following:
3. On the WAAS device, enter the following command:
4. Select the interface to use and make sure it is in the same subnet as the AppNav-XE service requestor: If interface is not specified, the default is GigabitEthernet0/0.
5. Configure and enable the AppNav service node auto discovery feature by entering the following:

DETAILED STEPS

Step 1 In Cisco IOS-XE, enter the following command. For the *SNG_name* parameter, enter the name of the service node group for which you want to enable the AppNav service node auto discovery feature. Ensure that the WAAS device is in the same subnet as the AppNav-XE component.

Example:

```
router(config)# service-insertion service-node-group
 SNG_name
```

Step 2 Enable the feature by entering the following:

Example:

```
router(config-service-insertion-sng)# node-discovery enable
```

Step 3 On the WAAS device, enter the following command:

Example:

```
WAAS(config)# service-insertion service-node
```

Step 4 Select the interface to use and make sure it is in the same subnet as the AppNav-XE service requestor: If interface is not specified, the default is GigabitEthernet0/0.

Example:

```
WAAS(config)# node-discovery enable GigabitEthernet 0/1
```

Step 5 Configure and enable the AppNav service node auto discovery feature by entering the following:

Example:

```
WAAS(config)# enable
```

Disabling the AppNav Service Node Auto Discovery Feature (For Cisco CSR 1000V Series Only)

You can disable the AppNav service node auto discovery feature by doing either of the following:

- Go to Cisco IOS-XE and disable the entire service node auto discovery feature for the entire system by entering the following:

```
router(config)# service-insertion service-node-group sng
router(config-service-insertion-sng)# no node-discovery enable
```

- Disable the service response feature on the WAAS node, as follows:

```
router(config)# service-insertion service-node
router(config)# no enable
```

Removing the AppNav-XE Configuration

To remove the AppNav-XE configuration, follow these steps:

Procedure

SUMMARY STEPS

1. From configuration mode, remove the interception from the WAN interface. Use these CLI commands:
2. Disable the AppNav service context. Use these CLI commands:
3. Remove the AppNav service context, service node group, and AppNav Controller group. Use these CLI commands:
4. Remove the AppNav policy map, class map, and access list. Use these CLI commands:

DETAILED STEPS

Step 1 From configuration mode, remove the interception from the WAN interface. Use these CLI commands:

Example:

```
router(config)# interface GigabitEthernet0/0/1
router(config-if)# no service-insertion waas
router(config-if)# exit
```

Step 2 Disable the AppNav service context. Use these CLI commands:

Example:

```
router(config)# service-insertion service-context  waas/1
router(config-service-insertion-context)# no enable
router(config-service-insertion-context)# exit
```

Step 3 Remove the AppNav service context, service node group, and AppNav Controller group. Use these CLI commands:

Example:

```
router(config)# no service-insertion service-context  waas/1
router(config)# no service-insertion service-node-group ISR-WAAS-SNG
router(config)# no service-insertion appnav-controller-group ISR-WAAS-SCG
```

Step 4 Remove the AppNav policy map, class map, and access list. Use these CLI commands:

Example:

```
router(config)# no policy-map type appnav ISR-WAAS
router(config)# no class-map type appnav match-any ISR-WAAS
router(config)# no ip access-list extended ISR-WAAS
router(config)# end
```

Configuring Port Channel Support for AppNav-XE

You can configure port channel support for AppNav-XE by indicating to the dataplane to swap IP addresses in the packets so that they can be distributed between different port channels.

To do this, use the following command:

```
(config)# service-insertion swap src-ip
(config)# [no] service-insertion swap src-ip
```

This command also enables AppNav-XE to handle packets from the Service Node whose ip addresses are swapped.

CHAPTER 5

Monitoring the AppNav-XE Component

This chapter contains the following sections:

AppNav Controller Show Commands

Checking the Status of the AppNav Controller

For Cisco CSR 1000V Series

Use the following command to check on the general status of the AppNav Controller. The command also lists all the interfaces that have “service-insertion waas” configured.

```
router# show service-insertion status
Hostname: AppNav-1-06
Device ID:0050.568b.2591
Platform Type: cisco (CSR1000V) VXE
IOS Version: 15.3(20130617:045351)
AppNav Controller Version: 1.0.0
AppNav Enabled Interfaces:
GigabitEthernet 1
```

For Cisco ASR 1000 Series

Use the following command to check on the general status of the AppNav Controller. The command also lists all the interfaces that have “service-insertion waas” configured.

```
router#show service-insertion status
Hostname: ASR1RU-3-36
Device ID: 7051.04b9.b480
Platform Type: cisco (ASR1001) 1RU
IOS Version: 15.3(20130317:122629)
AppNav Controller Version: 1.0.0
AppNav Enabled Interfaces:
GigabitEthernet0/0/1
```

Checking the Membership of the AppNav Controller Group

Use the following command to check the membership of the AppNav Controller group. It also lists all the service nodes configured and registered with the AppNav Controller.

```
router# show service-insertion appnav-controller-group
All AppNav Controller Groups in service context
Appnav Controller Group                        : acg
Member Appnav Controller Count                 : 2
Members:
      IP Address
        21.0.0.36
  21.0.0.160
AppNav Controller                              : 21.0.0.36
Local AppNav Controller : Yes
Current status of AppNav Controller            : Alive
Time current status was reached                : Wed Sep  5 15:50:06 2012
Cluster protocol ICIMP version                 : 1.1
Cluster protocol Incarnation Number            : 1
Cluster protocol Last Sent Sequence Number     : 0
Cluster protocol Last Received Sequence Number : 0
Current AC View of AppNav Controller
      IP Address
        21.0.0.36
  21.0.0.160
Current SN View of AppNav Controller
      IP Address
      21.0.0.149
AppNav Controller  : 21.0.0.160
Local AppNav Controller  : No
Current status of AppNav Controller  : Alive
Time current status was reached  : Thu Dec 6 20:17:53 2012
Cluster protocol ICIMP version  : 1.1
Cluster protocol Incarnation Number  : 1
Cluster protocol Last Sent Sequence Number  : 1355098374
Cluster protocol Last Received Sequence Number  : 1355089899
Current AC View of AppNav Controller
 IP Address
 21.0.0.36
 21.0.0.160
Current SN View of AppNav Controller
 IP Address
 21.0.0.149
```

Displaying Detailed Information About Service Node Groups and Service Nodes

Use the **show service-insertion service-node-group [***SNG_name* **| all]** command to display detailed information about service node groups and individual service nodes. You can also use this command to check the status of individual application accelerators.

The output of this command shows the following:

- Cluster protocol information. The *last sent sequence number* and the *last received sequence number* values should be increasing continuously.
- Number of service nodes and associated service contexts.
- Status of each service node, which can be either Alive or Dead
- Load state, which displays the health of the application accelerators. The load state can be one of the following:

  - green—application accelerator is functional and accepting new flows
  - yellow—application accelerator is functional but not accepting new flows
  - red—application accelerator is not functional

- Overall availability of the service node group for each application accelerator

```
router# show service-insertion service-node-group
Service Node Group name : sng1
 Service Context : waas/1
 Member Service Node count : 1
Service Node (SN) : 21.0.0.149
Auto discovered : No
SN belongs to SNG : sng1
Current status of SN : Alive
Time current status was reached : Thu Dec 6 20:17:11 2012
Cluster protocol DMP version : 1.1
Cluster protocol incarnation number : 2
Cluster protocol last sent sequence number : 1355101043
Cluster protocol last received sequence number: 1348909100
Health Markers:
 AO Load State  Since
 tcp GREEN 0d 5h 39m 38s
 epm GREEN 0d 5h 39m 38s
 cifs GREEN 0d 5h 39m 38s
 mapi GREEN 0d 5h 39m 38s
 http GREEN 0d 5h 39m 38s
 video GREEN 0d 5h 39m 38s
 nfs GREEN 0d 5h 39m 38s
 ssl YELLOW 0d 5h 39m 38s
 ica RED 0d 0h 0m 0s
SNG Availability per Accelerator
 AO Available Since
 tcp Yes 0d 5h 39m 38s
 epm Yes 0d 5h 39m 38s
 cifs Yes 0d 5h 39m 38s
 mapi Yes 0d 5h 39m 38s
 http Yes 0d 5h 39m 38s
 video Yes 0d 5h 39m 38s
 nfs Yes 0d 5h 39m 38s
 ssl No 0d 0h 0m 0s
 ica No 0d 0h 0m 0s
```

Displaying Class Maps and Policy Maps

The following commands reflect the running configuration and are useful for checking classifications without having to scan through an entire running configuration.

To display all type AppNav class maps and their matching criteria, or a specific AppNav class map and its matching criteria, use the following command:

```
router# show class-map type appnav
 [AppNav_class_name
```

To display all type AppNav policy maps and their class and action mappings, or a specified policy map and its class or action mappings, use the following command:

```
router# show policy-map type appnav
 [AppNav_policy_name
```

The **show policy-map target service-context [***service_context_name***]** command displays policy map information for service contexts. Use this command to view the flow level stats of all the class maps and policy maps that are configured under a service context. If you do not specify a service context name, the command displays all the configured class maps and policy maps.

Here are two examples:

```
router# show policy-map target service-context waas/1
Service-policy appnav input: p1
      Class-map: c1 (match-all)
         Match: access-group 101
       distribute service-node-group sng1
         Distributed: 0 packets, 0 bytes
         Passed through: 0 packets, 0 bytes
        Aggregate: 0 packets, 0 bytes
       monitor-load http
      Class-map: class-default (match-any)
         Match: any
router# show policy-map target service-context
    Service-policy appnav input: p1
      Class-map: c1 (match-all)
         Match: access-group 101
       distribute service-node-group sng1
         Distributed: 0 packets, 0 bytes
         Passed through: 0 packets, 0 bytes
       Aggregate: 0 packets, 0 bytes
       monitor-load http
      Class-map: class-default (match-any)
         Match: any
    Service-policy appnav input: p3
    Class-map: c3 (match-all)
      Match: access-group 101
      distribute service-node-group sng3
         Distributed: 0 packets, 0 bytes
         Passed through: 0 packets, 0 bytes
       Aggregate: 0 packets, 0 bytes
    Class-map: class-default (match-any)
       Match: any
```

Displaying Service Context Information

To display information about service contexts, use the **show service-insertion service-context [***service_context_name***]** command. The output of this command displays the status of the specified service context, including the following:

- Current and last states of the Cluster Membership Manager (CMM) and FSM
- State of the cluster
- Views of the stable and current AppNav Controller and service nodes

Here is an example:

```
router# show service-insertion service-context waas/1
Service Context                             : waas/1
Cluster protocol ICIMP version              : 1.1
Cluster protocol DMP version                : 1.1
Time service context was enabled            : Thu Sep  8 08:38:41 2011
Current FSM state                           : Operational
Time FSM entered current state              : Thu Sep  8 08:48:26 2011
Last FSM state                              : Converging
Time FSM entered last state                 : Thu Sep  8 08:48:16 2011
Cluster operational state                   : Operational
Stable AppNav Controller View:
        2.58.2.40
Stable SN View:
        2.43.139.170     2.58.2.40
```

```
Current AppNav Controller View:
        2.58.2.40
Current SN View:
        2.43.139.170     2.58.2.40
```

Displaying Data Path Statistics

Displaying AppNav Controller Group Statistics

To see the number of “keepalives” sent to the other AppNav Controllers and received from the other AppNav Controllers and other statistics related to the AppNav Controller group, use the following command:

```
router# show service-insertion statistics appnav-controller-group
Appnav Controller Group    : acg
Number of AppNav Controllers    : 2
Members:
 IP Address
 21.0.0.36
 21.0.0.160
Aggregate Appnav Controller statistics
--------------------------------------
Time since statistics were last reset/cleared   : 0d 5h 47m 14s
Aggregate number of keepalives sent to ACs   : 168484
Aggregate number of keepalives received from ACs  : 166372
Aggregate number of invalid keepalives received :
 Total   : 0
 Incompatible ICIMP version   : 0
 Authentication Failed   : 0
 Stale keepalive   : 0
 Malformed keepalive   : 0
 Unknown keepalive   : 0
 Inactive keepalive   : 0
Aggregate number of times liveliness lost with ACs    : 1
Aggregate number of times liveliness gained with ACs : 2
```

Displaying Per Service Node and Service Node Group Statistics

To show the connections, packets, and bytes sent to each service node, use the following command:

```
router# show service-insertion statistics service-node
 [IP_address
```

To show the aggregated connections, packets, and bytes sent to each service node group, use this command:

```
router# show service-insertion statistics service-node-group [
NAME
]
```

Here is an example:

```
router# show service-insertion statistics service-node
Statistics for Service Node 21.0.0.149
-------------------------------------------------------
Time since statistics were last reset/cleared: 0d 18h 7m 54s
Number of probe requests sent to SN   : 326024
Number of probe responses received from SN   : 326014
Number of invalid probe responses received:
 Total  : 0
 Incompatible DMP version  : 0
 Authentication failed  : 0
 Stale response  : 0
 Malformed response  : 0
 Unknown response:   0
```

```
Number of times liveliness lost with SN : 0
Number of times liveliness regained with SN :1
Cluster IPC statistics
----------------------
Time since statistics were last reset/cleared: 0d 18h 8m 24s
Number of load updates received from CMM: 4
Number of erroneous load updates: 0
Time since last load update was received: 0d 14h 32m 43s
Load stats for Service Node 21.0.0.149
---------------------------------------------
Accelerator state transition statistics
---------------------------------------
Time since Accl load stats were last cleared: 0d 18h 8m 24s
Accl  Current    Previous   Red  Yellow   Green
tcp  GREEN    RED   0  0   1
epm  GREEN    RED   0  0   1
cifs  GREEN    RED   0  0   1
mapi  GREEN    RED   0  0   1
http  GREEN    RED   0  0   1
video  GREEN    RED   0  0   1
nfs  GREEN    RED   0  0   1
ssl  YELLOW    RED   0  1    0
ica  RED   RED   0  0   0
Traffic distribution statistics for service node 21.0.0.149
----------------------------------------------------------
Time since distribution stats were last cleared: 0d 18h 8m 24s
Packet and byte counts
----------------------
Redirected Bytes   : 2338
Redirected Packets   : 50
Received Bytes   : 3350
Received Packets   : 50
Occurences
-----------
Initial Redirects   : 2
Initial Redirects Accepted   : 2
Initial Redirect -> Passthrough   : 0
Redirect -> Passthrough   : 0
```

The important statistics are as follows:

- Probe Requests: The number of heartbeats sent to the service node.
- Probe Responses: The number of heartbeats received from the service node.
- Redirected Bytes: The number of bytes redirected to the service node.
- Redirected Packets: The number of data packets redirected to the service node.
- Received Bytes: The number of bytes received from the service node.
- Received Packets: The number of data packets received from the service node.
- Initial Redirects: The number of times that the SYN packet (the first packet for requesting connection in a TCP flow) was redirected to the service node.
- Initial Redirects Accepted: The number of times that the service node decided to optimize on SYN packet.
- Initial Redirects -> Passthrough: The number of times that the service node decided to pass-through on SYN packet.
- Redirect -> Passthrough: The number of times that the service node decided to pass-through a flow after it was initially accepted for optimize (e.g. due to lack of peer).

Displaying Service Context Statistics

To display statistics about the service context, use the **show service-insertion statistics service-context [***name***]** command. The output of this command displays the time spent in each FSM state by the CMM and the amount of time that each service context has been in each FSM state.

Here is an example:

```
Router# show service-insertion statistics service-context
Time spent in various FSM states
Converging     :       0d 0h 0m 31s
Initializing   :       0d 0h 0m 0s
Operational    :       1d 19h 27m 53s
Degraded       :       0d 0h 0m 0s
Internal Error :       0d 0h 0m 0s
Admin Disabled :       0d 0h 0m 0s
Number of entries into Converging State:      3
Number of entries into Initializing State:    1
Number of entries into Operational State:     3
Number of entries into Degraded State:        0
Number of entries into Internal Error State:  0
Number of entries into Admin Disabled State:  0
```

Displaying Flow Statistics

To query the flows in the flow table and to optionally filter the output by using specific criteria, use the following command:

```
router# show service-insertion statistics connection [[summary] | [vrf-name
 name
] [client-ip
IP_address
] [client-port
port_number
] [server-ip
IP_address
] [server-port
port_number
] [detail]]
```

As part of the flow query, the following information for every flow is available:

- Client IP address, client TCP port and server IP address, server TCP port number
- Service node IP address, passthrough
- VRF name

Here is an example:

```
router# show service-insertion statistics connection
Collecting Records. Please wait...
Client                Server                SN-IP  VRF-Name
51.0.222.4:64234      11.0.0.3:80           21.0.0.104 br_vrf
51.0.222.4:22415      11.0.0.3:80           21.0.0.104
51.0.222.4:15264      11.0.0.3:80           21.0.0.104
51.0.222.4:37759      11.0.0.3:80           21.0.0.104
51.0.222.4:55408      11.0.11.2:23          Passthrou
```

If you include the *detail* keyword, the report also displays the following on a per flow basis:

- Presence of session (3T) or App (2T) association
- Application ID

- Peer ID

The following is an example:

```
router# show service-insertion statistics connection detail
Collecting Records. Please wait...
Client: 192.168.80.4:60973
Server: 192.168.180.4:135
Service Node IP: 172.16.0.2
Flow association: 2T:No,3T:No
VRF-Name:
Application ID: 0
Peer-ID: 00:21:5e:76:65:08

Client: 192.168.80.4:60959
Server: 192.168.180.4:1092
Service Node IP: 172.16.0.2
Flow association: 2T:Yes,3T:Yes
VRF-Name:
Application ID: 78
Peer-ID: 00:21:5e:76:65:08
```

If you include the *summary* keyword, the report displays only the number of 2T and 3T entries, the number of optimized flows, and the number of passthrough flows and the number of flow synchronization failures due to VRF config mismatch on the AppNav Controllers.

The following is an example:

```
router# show service-insertion statistics connection summary
Number of 2T optimized flows    = 0
Number of 3T optimized flows    = 0
Number of optimized flows       = 3
Number of pass-through flows    = 1
Flow sync failures due to vrf-mismatch = 0
```

You can also use the **show platform software** command. It works exactly the same as the **show service-insertion statistics** command, but it can also be used to query the flows on the standby FP.

```
router# show platform software appnav-controller <f0 | f1 | fp active | fp standby>
connections ...
```

Displaying Application and Session Statistics

To query the application and session entries and to optionally filter the output by using specific criteria, use the following command:

```
router# show service-insertion statistics sessions [[vrf-name
name
] [client-ip
IP_address
][server-ip
IP_address
] [server-port
port_number
] [detail]]
```

Application entries do not have client or service node IP addresses.

Here is an example:

```
router# show service-insertion statistics sessions
Collecting Records. Please wait...
Client     Server      SN-IP    VRF-Name
N/A     192.168.180.4:1092      N/A
192.168.80.4:0     192.168.180.4:1092      172.16.0.2
```

If you include the detail keyword, the report also displays the application ID and the time since the last activity.

Here is an example:

```
Router# show service-insertion statistics sessions detail
Collecting Records. Please wait...
Client: 192.168.80.4:0
Server: 192.168.180.4:1098
Service Node IP: 172.16.0.2
VRF-Name:
Application ID: 78
Time since last activity : 0hr 36min 30sec

Client: N/A
Server: 192.168.180.4:1098
Service Node IP: N/A
VRF-Name:
Application ID: 78
Time since last activity : 0hr 36min 30sec
```

You can also use the show platform software command. It works exactly the same as the show service-insertion statistics command, but it can also be used to query the application and session entries on the standby FP.

```
router# show platform software appnav-controller <f0 | f1 | fp active | fp standby> sessions
 ...
```

Displaying Classification Statistics

Use the **show policy-map target service-context [***service_context_name***]** command to view the flow level statistics of all the class maps and policy maps that are configured under a service context. If you do not enter a service context name, the system displays all the configured class maps and policy map output.

The following are examples:

```
router# show policy-map target service-context waas/1
Service-policy appnav input: p1
Class-map: c1 (match-all)
      Match: access-group 101
      distribute service-node-group sng
        Distributed: 313450 packets, 135820480 bytes
        Passed through: 0 packets, 0 bytes
       Aggregate: 313450 packets, 135820480 bytes
 monitor-load http
    Class-map: c2 (match-all)
      Match: access-group 102
 Pass-through
  Distributed: 0 packets, 0 bytes
        Passed through: 40 packets, 30000 bytes
       Aggregate: 40 packets, 30000 bytes
    Class-map: class-default (match-any)
      Match: any
router# show policy-map target service-context
Service-policy appnav input: p1
      Class-map: c1 (match-all)
         Match: access-group 101
       distribute service-node-group sng1
         Distributed: 0 packets, 0 bytes
         Passed through: 0 packets, 0 bytes
       Aggregate: 0 packets, 0 bytes
       monitor-load http
      Class-map: class-default (match-any)
         Match: any
    Service-policy appnav input: p3
Class-map: c3 (match-all)
      Match: access-group 101
      distribute service-node-group sng3
         Distributed: 0 packets, 0 bytes
         Passed through: 0 packets, 0 bytes
       Aggregate: 0 packets, 0 bytes
    Class-map: class-default (match-any)
       Match: any
```

Displaying Pass Through Reason Statistics

To view the passthrough reason statistics aggregated for all the classes of a policy associated with the specified service context, use the following command:

```
router# show policy-map target service-context
context_name
 passthru-reason
```

To view the passthrough reason statistics for a particular class of a policy associated with the specified service context, use the following command:

```
router# show policy-map target service-context
context_name
 class
class_name
 passthru-reason
```

Here is an example:

```
router# show policy-map target service-context waas/1 class c4 passthru-reason
Service-policy appnav input: p4
Class-map: c4 (match-all)
 Match: access-group 101
 distribute service-node-group sng4
  Distributed: 11 packets, 222 bytes
  Passed through: 100 packets, 22000 bytes
 Aggregate: 111 packets, 22222 bytes
Collected by SC:
Passthrough Reasons Packets  Bytes
------------------------ -------  - ------
PT Flow Learn Failure  0 0
PT SNG Overload  0 0
PT Appnav Policy  0 0
PT Cluster Degrade  0 0
PT ZBFW  0 0
PT NAT ALG  0 0
PT Unknown  0 0
Indicated by SN:
Passthrough Reasons Packet   Bytes
------------------------ ------- - ------
PT No Peer  100 22000
PT Rjct Capabilities  0 0
PT Rjct Resources  0 0
PT Rjct No License  0 0
PT App Config  0 0
PT Global Config  0 0
PT Asymmetric  0 0
PT In Progress  0 0
PT Intermediate  0 0
PT Overload  0 0
PT Internal Error  0 0
PT App Override  0 0
PT Server Black List  0 0
PT AD Version Mismatch  0 0
PT AD AO Incompatible  0 0
PT AD AOIM Progress  0 0
PT DM Version Mismatch  0 0
PT Peer Override  0 0
PT Bad AD Options  0 0
PT Non-optimizing Peer  0 0
PT SN Interception ACL   0 0
PT IP Fragment Unsupported  0 0
```

```
 -----------  -------
PT Overall  100 22000
```

Displaying Alarms

Use the following command to display the alarms seen on the AppNav Controller. The **detail** option gives a brief explanation of each alarm and the **support** option gives a longer explanation along with a recommended action.

```
router# show service-insertion alarms [critical | major | minor] [detail [support]]
```

The following is an example:

```
router# show service-insertion alarms detail
Critical Alarms:
----------------
 Alarm Instance Alm ID Module  AC/SN IP Addr AO SNG
 1 degraded_cluster 29002 cmm N/A N/A N/A
 Cluster protocol detected inconsistency in AC view of peer ACs. Device will pass-through
all new connections.
Major Alarms:
-------------
 Alarm Instance Alm ID Module AC/SN IP Addr AO SNG
 1 ac_unreachable 29006 cmm 192.168.1.11 N/A N/A
 Cluster protocol on device cannot communicate with peer AC ("192.168.1.11").
 2 sn_unreachable 29007 cmm 192.168.2.31 N/A N/A
 Cluster protocol on device cannot communicate with peer SN ("192.168.2.31").
 3 sng_unavailable 30001 fdm N/A N/A sng1
 Service Node Group ("sng1") has become unavailable.
 4 sng_ao_unavailable 30000 fdm N/A sslsng
 Service Node Group ("sng") has become unavailable for accelerator - ("ssl").
Minor Alarms:
-------------
None
```

AppNav Service Node Auto Discovery Show Commands (For Cisco CSR 1000V Series Only)

Use the following commands to show information about the AppNav service node auto discovery feature:

Note The Auto Discovery feature is only available on the Cisco CSR 1000V Series.

show service-insertion service-node-group *SNG_name*

```
router# show service-insertion service-node-group sng
Service Node Group name : sng
Service Context : waas/1
Member Service Node count : 2
Service Node (SN) : 20.20.20.20
Auto discovered : Yes
SN belongs to SNG : sng
Current status of SN : Alive
Time current status was reached : Thu Dec 6 00:51:48 2012
Cluster protocol DMP version : 1.1
Cluster protocol incarnation number : 13
Cluster protocol last sent sequence number : 1355026900
Cluster protocol last received sequence number: 131214317
```

```
Health Markers:
 AO Load State Since
 tcp YELLOW 0d 14h 3m 53s
 epm RED 0d 0h 0m 0s
 cifs RED 0d 0h 0m 0s
 mapi RED 0d 0h 0m 0s
 http RED 0d 0h 0m 0s
 video RED 0d 0h 0m 0s
 nfs RED 0d 0h 0m 0s
 ssl RED 0d 0h 0m 0s
 ica RED 0d 0h 0m 0s
Service Node (SN) : 1.2.3.4
Auto discovered : No
SN belongs to SNG : sng
Current status of SN : Dead
Time current status was reached : Thu Dec 6 14:19:52 2012
Cluster protocol DMP version : 0.0
Cluster protocol incarnation number : 0
Cluster protocol last sent sequence number : 1355026901
Cluster protocol last received sequence number: 0
Health Markers:
 AO Load State Since
 tcp RED 0d 0h 0m 0s
 epm RED 0d 0h 0m 0s
 cifs RED 0d 0h 0m 0s
 mapi RED 0d 0h 0m 0s
 http RED 0d 0h 0m 0s
 video RED 0d 0h 0m 0s
 nfs RED 0d 0h 0m 0s
 ssl RED 0d 0h 0m 0s
 ica RED 0d 0h 0m 0s
SNG Availability per Accelerator
 AO Available Since
 tcp No 0d 0h 0m 0s
 epm No 0d 0h 0m 0s
 cifs No 0d 0h 0m 0s
 mapi No 0d 0h 0m 0s
 http No 0d 0h 0m 0s
 video No 0d 0h 0m 0s
 nfs No 0d 0h 0m 0s
 ssl No 0d 0h 0m 0s
 ica No 0d 0h 0m 0s
```

show service-insertion service-node-group *SNG_name* auto-discovered

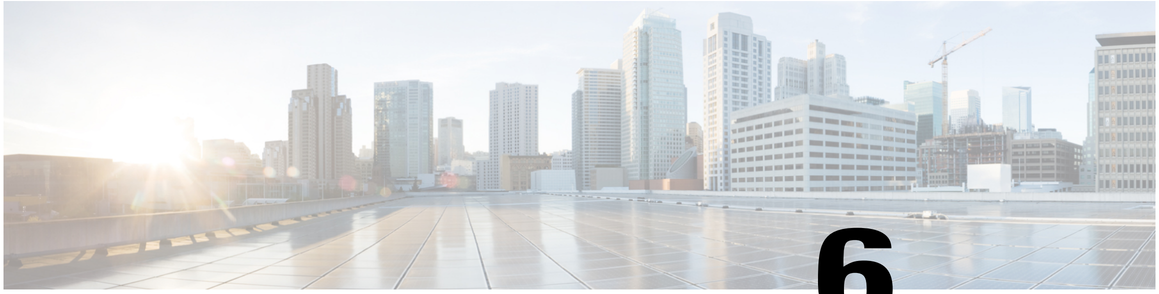
```
router# show service-insertion service-node-group sng auto-discovered
 MAC Address Resp Elapsed Minutes IP Address
 50:57:a8:e1:af:1 0 20.20.20.20
```

show mdns request

```
router# show mdns request
MDNS Outstanding Requests
======================================================
Request name : _appnav_waas_node._udp.local
Request type : PTR
Request class : IN
```

show mdns stat

```
router# show mdns stat
mDNS Statistics
mDNS packets sent : 852
mDNS packets received : 510
mDNS packets dropped : 0
```



CHAPTER 6

Troubleshooting

This chapter explains how to troubleshoot common problems and covers the following sections:

Using Debug Commands

AppNav-XE Debug Commands

Clearing AppNav-XE Statistics

To clear all the AppNav-XE statistics or just certain statistics, use the following command:

```
router# clear
service-insertion statistics ?
all Clear all service-insertion statistics
appnav-controller Clear appnav-controller statistics
appnav-controller-group Clear appnav-controller-group statistics
service-context Clear service-context statistics
service-node Clear service-node statistics
service-node-group  Clear service-node-group statistics
```

Debugging the Cisco IOS-XE Control Plane

Use the following debug commands to trace control plane activities:

```
router# debug appnav-controller ?
  cm-reg  Debugging AppNav Controller CM registration with the WCM server
  cmm     Debugging AppNav Controller Cluster Management (CMM)
  fdm     Debugging AppNav Controller Flow Distribution Module (FDM)
  ha      Enable AppNav Controller high availability (HA) redundancy checkpoint and ISSU
 infrastructure debugs
  vi      Debugging AppNav Controller Virtual Interface (VI), including the status at the
  time of creation and links to the compress and uncompress interface
router# debug appnav-controller cmm ?
  all      Enable all CMM debugs
  cli      Enable CMM CLI debugs
  events   Enable CMM state machine event debugs
  misc     Enable CMM misc debugs
  packets  Reception and transmission of packets (can be filtered based on IP address)
  shell    Enable CMM misc debugs
  timers   Enable CMM misc debugs
router# debug appnav-controller fdm ?
  all     Enable all FDM debugs
  events  Enable debugging for important events being handled by FDM
  infra   Enable debugging for FDM infrastructure events
```

The following debug commands are the most useful:

- **debug appnav-controller cmm events**
- **debug appnav-controller fdm events**
- **debug appnav-controller ha**

Debugging the Cisco IOS-XE Infrastructure

Showing Packet Drop Statistics

Use the following command to display unplanned packet drops:

```
router# show platform hardware qfp active statistics drop
-------------------------------------------------------------------------
Global Drop Stats                         Packets                  Octets
-------------------------------------------------------------------------
AppNavBadRoute                                 38                    2888
Ipv4AclLookupMiss                              42                    3034
Ipv4NoRoute                                  4408                 1293334
UnconfiguredIpv4Fia                            19                    1710
```

The following are the reasons for which packets may drop:

- AppNavInvSNpkt—Malformed or unsupported packet from the service node.
- AppNavInternalErr—Logic error within the AppNav-XE component. Uncommon.
- AppNavBadRoute—A non-AppNav-XE packet appeared at the AppNav-XE virtual or tunnel interface. Very common when routing protocols are enabled.
- AppNavNoTunnel—There is no tunnel facility available for the service node-bound packet.
- AppNavNoSvcCtx—There is no service context matching the flows from the service node.
- AppNavInvFOState—The flow state is no longer valid. This is usually due to changes in the configuration.
- AppNavUnexpctdpkt—The AppNav-XE component did not expect to process more packets because it has been shut down.

Showing Data Path CPU Utilization

To display the data path CPU utilization, use the following command:

```
router# show platform hardware qfp active datapath utilization
 CPP 0: Subdev 0 5 secs 1 min 5 min 60 min
Input: Priority (pps)  0 0 0 0
   (bps) 0 72 88 48
Non-Priority (pps)  226455 225968 198785 72441
  (bps) 1879325304 1875408168 1648044616 599951168
 Total (pps) 226455 225968 198785 72441
  (bps) 1879325304 1875408240 1648044704 599951216
Output: Priority (pps) 229023 228474 245267 90057
 (bps) 1619093520 1641710256 2389617496 949076160
Non-Priority (pps) 209522 208053 293300 104501
 (bps) 180090080 178161632 3124680344 1191566064
 Total (pps) 438545 436527 538567 194558
 (bps) 1799183600  1819871888 5514297840 2140642224
Processing: Load (pct) 26 26 19 8
```

Showing Data Path Memory Utilization

Use the following command to show statistics about the data path memory use.

Note The value for In Use DRAM memory must be less than 90 percent of the value for Total DRAM memory; otherwise, the AppNav-XE component stops optimizing new flows.

```
router# show platform hardware qfp active infrastructure exmem statistics
QFP exmem statistics
Type: Name: DRAM, QFP: 0
 Total: 268435456
 InUse: 99933184
 Free: 168502272
 Lowest free water mark: 168502272
Type: Name: IRAM, QFP: 0
 Total: 134217728
 InUse: 8087552
 Free: 126130176
 Lowest free water mark: 126130176
Type: Name: SRAM, QFP: 0
 Total: 32768
 InUse: 15088
 Free: 17680
 Lowest free water mark: 17680
```

Debugging the Data Plane

The output of the following debug command is displayed as a log file named /tmp/fp/trace/cpp_cp_*Fx* -0.log under the FP shell, where *Fx* is either F0 or F1 depending on the active FP module. You need a shell license to access the FP shell.

If you do not have shell access, you can use the **test platform software trace slot fp act cpp-control-process rotate** command to force the log to flush to bootflash:tracelogs.

```
router# debug platform hardware qfp active feature appnav-controller datapath ?
classify    Debug QFP flow classification such as traces, policy, peer ID, and
classification action (which service node group)
drop        Enable drop debugging and shows traces of packet drop due to errors
fdl         Debug QFP flow distribution such as selecting a service node within a service
   node group
ha          Debug QFP high availability (HA) and AppNav Controller issues. Shows traces
 related to syncing flows between AppNav Controllers and between active and    standby FPs
interop     Debug QFP feature interoperations such as FNF, NBAR, and NAT
pkt-path    Debug QFP packet processing and packet interception
proxy       Debug QFP proxy issues related to interface with the control place, such as
 statistics reporting and configuration
```

Each of the above categories (other than **drop**, which has no level) has the following four levels:

- Error—Displays error level debugs and detects potential issues.
- Warn—Displays warnings and errors.

- Info—Displays information, warnings, and errors.
- All—The lowest level of debugging. Displays all debugs.

To limit the number of debug messages, we recommend that you only enable the error debug level first and then slowly reduce the debug level.

You can also use the following command to check on packets dropped by the router. The command lists all the packets that were dropped with a reason. If you see AppNav drop reasons, you can enable the debug drop command to see the actual packet drops inside the trace logs.

```
router# show platform hardware qfp active statistics drop
Global Drop Stats Packets Octets
-------------------------------------------------------
The Global drop stats were all zero
```

AppNav Service Node Auto Discovery Debug Commands (For Cisco CSR 1000V Series Only)

Use the following debug commands to trace the AppNav service node auto discovery feature on the Cisco CSR 1000V Series:

- **debug appnav auto-discovery**
- **debug mdns all**
- **debug mdns packet**

Note The Auto Discovery debug commands are applicable to the Cisco CSR 1000V Series only.

Common Problems

Traffic Not Redirected

If traffic is not redirected properly, ensure that “service-insertion waas” is present on interfaces on which the traffic is supposed to be intercepted. Issue the **show service-insertion status** command to verify this.

Traffic Passed Through Instead of Redirected

The show service-insertion statistics connection command indicates whether traffic is passed through or redirected. If traffic is passed through instead of being redirected, use the show policy-map target service-context *context_name* passthru-reason command to find out the reason. For details, see the “Displaying Pass Through Reason Statistics” section on page 11 .

You can also monitor the service node counters. See the “Displaying Per Service Node and Service Node Group Statistics” section on page 6 .

The term “Initial Redirect” indicates that flows are being redirected to the service nodes. If the flows are not being redirected to the service nodes, maybe the policy did not cover the traffic type.

The “Initial Redirect -> Passthrough” counter indicates that the service node has decided to pass-through the flow. This is likely due to policies on the service node.

The “Redirect -> Passthrough” counter indicates that the service node later decided to pass-through the flow. This is likely due to lack of a peer WAAS device. Two WAAS devices are needed along the path to optimize a flow.

Degraded Cluster

If connections are passed through and you are using an AppNav Controller group that has two or more AppNav Controllers, it is possible that the cluster state is degraded instead of operational. This means that the AppNav Controller view is not the same on each of the AppNav Controllers.

To check the cluster state and the stable AppNav Controller view on each of the AppNav Controllers, use the following command:

```
router# show service-insertion service-context
Service Context  : waas/1
Cluster protocol ICIMP version  : 1.1
Cluster protocol DMP version  : 1.1
Time service context was enabled  : Fri Dec 7 19:28:11 2012
Current FSM state  : Degraded
Time FSM entered current stat  : Fri Dec 7 21:58:29 2012
Last FSM state  : Converging
Time FSM entered last state   : Fri Dec 7 21:58:19 2012
Cluster operational state   : Degraded
Stable AppNav controller View:
 21.0.0.145
 21.0.0.160
Stable SN View:
 21.0.0.149
```

The reason for the difference in AppNav Controller views on the AppNav Controllers may be due to a mismatch in the AppNav Controller group configuration on the AppNav Controllers or due to a connectivity problem between the AppNav Controllers.

It is also useful to check the alarms on each of the AppNav Controllers by using the following command that also suggests corrective actions:

```
router# show service-insertion alarms detail support
Critical Alarms:
----------------
 Alarm Instance Alm ID Module  AC/SN IP Addr AO SNG
 1 degraded_cluster 29002 cmm N/A N/A N/A
 Cluster protocol detected inconsistency in AC view of peer ACs. Device will pass-through
all new connections.
 Cluster view is degraded.
```

Explanation:

Cluster membership manager has detected a discrepancy in the AC view of peer ACs. Optimization will be turned off on this device for cluster consistency.

Action:

```
  Check the network for partial connectivity issues.
Major Alarms:
-------------
 Alarm Instance Alm ID Module AC/SN IP Addr AO SNG
 1 ac_unreachable 29006 cmm 192.168.1.11 N/A N/A
 Cluster protocol on device cannot communicate with peer AC ("192.168.1.11").
```

AppNav controller is unreachable.

Explanation:

Cluster protocol detected failure of the peer AC. This could happen due to several reasons - configuration mismatch or network issues preventing communication between the ACs or the AC actually being down.

```
 Action:
  Other alarms will indicate if this is a configuration issue. If so, correcting the
configuration mismatch will cause this alarm to go away. Otherwise, check the network to
see if the devices are able to communicate with each other.
Minor Alarms:
-------------
None
```

Service Node Excluded

If no traffic is redirected to a particular service node and you are using an AppNav Controller group with two or more AppNav Controllers, it is possible that the service node is excluded. This happens when the service node view is not the same on each of the AppNav Controllers.

To check the stable service node view on each of the AppNav Controllers, use the show service-insertion service-context command.

The reason for the difference in service node views could be due to a mismatch in the service node group configuration on the AppNav Controllers or due to a connectivity problem between one or more of the AppNav Controllers and the excluded service node.

To check if any service nodes are excluded or unreachable, look for the SN_excluded and SN_unreachable alarms by using the show service-insertion alarms detail support command on each of the AppNav Controllers.

Flows Not Synced Between AppNav Controllers

This could be due to a mismatch in the VRF names for the traffic seen by the AppNav Controllers in the ACG.

Check the output of the show service-insertion statistics connection summary command for the counter for Flow Sync Failures due to vrf mismatch.

```
router# show service-insertion statistics connection summary
Number of 2T entries=0
Number of 3T entries=0
Number of optimized flows=0
Number of pass-through flows=0
Flow sync failures due to vrf-mismatch=3
```

Connection Hangs

A connection might be considered “hung” for various reasons. In many cases, it helps to use telnet to simulate a connection to the server. For example, enter **telnet** *HTTP_server* **80**.

If the connection hangs during the TCP 3-way handshake, verify that both the connection and the route to the service node are properly set up.

If the connection hangs after the connection was established, verify the connection along the path. Make sure that the MTU along the path is correct.

Use the **show service-insertion statistics connection** command on the AppNav Controller and the **show statistics connection** command on the service node to cross check the connections between the AppNav Controller and the service node.

Use the **show platform hardware qfp active statistics drop** command to check for packet drops.

Connection Resets

You can usually see the reason for the connection reset by issuing the **show statistics connection closed** command and the **show statistics connection closed conn-id** *connection_ID* command on the service node. Capturing packets is also useful in analyzing the reason for the connection reset.

Use the **show platform hardware qfp active statistics drop** command to check for dropped packet.

Application Accelerator Status Shows as Red with No Load

Some older service nodes may not support all application accelerators.

Individual application accelerators, such as the video application accelerator, require a separate license.

The AppNav-XE Component Fails to Initialize

If the system displays an ERROR_NOTIFY syslog message when you enable the **service-insertion waas** command on the interface, it could be that the AppNav-XE component failed to initialize due to low memory. Check the amount of memory by using the following command:

```
router# show platform hardware qfp active infrastructure exmem statistics
QFP exmem statistics
Type: Name: DRAM, QFP: 0
  Total: 268435456
  InUse: 102283264
  Free: 166152192
  Lowest free water mark: 166152192
Type: Name: IRAM, QFP: 0
  Total: 134217728
  InUse: 8186880
  Free: 126030848
  Lowest free water mark: 126030848
Type: Name: SRAM, QFP: 0
  Total: 32768
  InUse: 15088
  Free: 17680
  Lowest free water mark: 17680
```

If the available memory is less than 10 percent of the total memory, the AppNav-XE component may not be able to initialize, which results in no flows being redirected.

If the output of the **show policy-map target service-context waas/1** command is blank, instead of listing the AppNav policy being used, it may indicate that the system was unable to initialize.

Flow Limit Reached

Both the AppNav Controller and the service nodes have a limit on the number of flows that they can support. On the AppNav Controller, the limit is 2 million flows. Beyond that, all flows are passed through. If you exceed the limit, the system displays the following error message:

```
03/10 00:53:51.720 [errmsg]: (warn): %CFT_CLIENT-4-MAX_FCS_TOUCH_WARN: CFT number of
flow-context threshold is reached, can't allocate more memory for flow-context.
```

The flow limit may be reached in advance due to available memory. In this case, the system displays the following syslog message:

```
*Aug 24 00:29:17.205: %CFT_CLIENT-4-CFT_MEMORY_BOUNDARY_TOUCH_WARN: F0: cpp_cp:  CFT reached
 maximum configured memory utilization. Can't allocate more memory for flow-context.
```

In both cases, when the existing flows are completed and the number of flows dips below the threshold, flows are optimized again.

Other AppNav-XE Known Issues

If the AppNav Controller does not respond to a WAAS TCP trace, the system forwards the TCP trace to the service node and the service node generates a response along with a list of service nodes along the path.