ECS Administration Guide, StarOS Release 21.17

First Published: 2019-12-19

Last Modified: 2021-02-05

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

Note The HA, HSGW, PDSN, and SecGW products have reached end of life and are not supported in this release. Any references to these products (specific or implied) their components or functions including CLI commands and parameters in this document are coincidental and are not supported. Full details on the end of life for these products are available at https://www.cisco.com/c/en/us/products/collateral/wireless/asr-5000-series/eos-eol-notice-c51-740422.html.

This preface describes the Enhanced Charging Services Administration Guide, how it is organized and its document conventions.

Enhanced Charging Services (ECS) is a StarOS in-line service application that runs on Cisco® ASR 5500 and virtualized platforms.

- Conventions Used, on page xix
- Supported Documents and Resources, on page xx
- Contacting Customer Support, on page xxi

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**

Supported Documents and Resources

Related Common Documentation

The following common documents are available:

- *AAA Interface Administration and Reference*
- *Command Line Interface Reference*
- *GTPP Interface Administration and Reference*
- *Installation Guide* (platform dependant)
- *Release Change Reference*
- *SNMP MIB Reference*
- *Statistics and Counters Reference*
- *System Administration Guide* (platform dependant)
- *Thresholding Configuration Guide*

Related Product Documentation

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

The following product documents are also available and work in conjunction with the ECS:

- *ADC Administration Guide*

- *CF Administration Guide*
- *GGSN Administration Guide*
- *IPSec Reference*
- *MME Administration Guide*
- *MURAL Installation and Administration Guide*
- *MURAL User Guide*
- *NAT Administration Guide*
- *PSF Administration Guide*
- *P-GW Administration Guide*
- *SAEGW Administration Guide*
- *SGSN Administration Guide*
- *S-GW Administration Guide*

Obtaining Documentation

The most current Cisco documentation is available on the following website:

http://www.cisco.com/cisco/web/psa/default.html

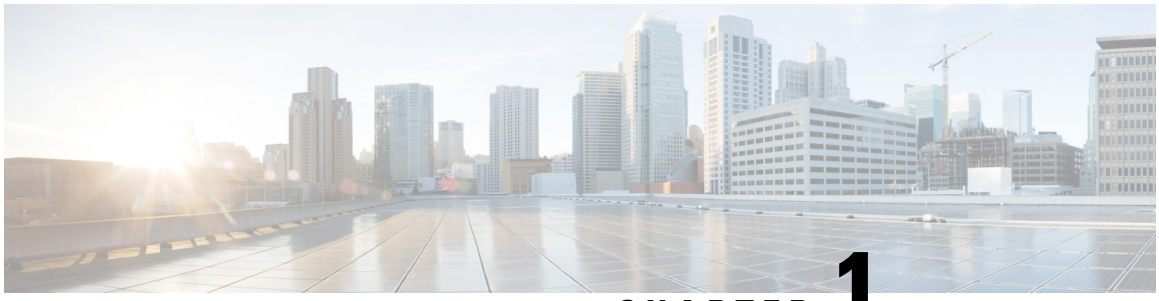
Use the following path selections to access the ECS documentation:

Products > Wireless > Mobile Internet> Inline Services > Cisco Enhanced Charging Services

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

Enhanced Charging Service Overview

This chapter provides an overview of the Enhanced Charging Service (ECS) in-line service, also known as Active Charging Service (ACS).

This chapter covers the following topics:

Introduction

The Enhanced Charging Service (ECS) is an in-line service feature that enables operators to reduce billing-related costs and gives the ability to offer tiered, detailed, and itemized billing to their subscribers. Using shallow and deep packet inspection (DPI), ECS allows operators to charge subscribers based on actual usage, number of bytes, premium services, location, and so on. ECS also generates charging records for postpaid and prepaid billing systems.

The ECS is an enhanced or extended premium service. The *System Administration Guide* provides basic system configuration information, and the product administration guides provide information to configure the core network service functionality. It is recommended that you select the configuration example that best meets your service model, and configure the required elements for that model before using the procedures in this document.

Qualified Platforms

ECS is a StarOS in-line service application that runs on Cisco ASR 5500 and virtualized platforms. For additional platform information, refer to the appropriate *System Administration Guide* and/or contact your Cisco account representative.

License Requirements

The ECS in-line service is a licensed Cisco feature. Separate session and feature licenses 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 the *Software Management Operations* chapter in the *System Administration Guide*.

Basic Features and Functionality

This section describes basic features of the ECS in-line service.

Shallow Packet Inspection

Shallow packet inspection is the examination of the layer 3 (IP header) and layer 4 (for example, UDP or TCP header) information in the user plane packet flow. Shallow packet analyzers typically determine the destination IP address or port number of a terminating proxy.

Deep Packet Inspection

Deep-packet inspection is the examination of layer 7, which contains Uniform Resource Identifier (URI) information. In some cases, layer 3 and 4 analyzers that identify a trigger condition are insufficient for billing purposes, so layer 7 examination is used. Whereas, deep-packet analyzers typically identify the destination of a terminating proxy.

For example, if the Web site "www.companyname.com" corresponds to the IP address 1.1.1.1, and the stock quote page (www.companyname.com/quotes) and the company page (www.companyname.com/business) are chargeable services, while all other pages on this site are free. Because all parts of this Web site correspond to the destination address of 1.1.1.1 and port number 80 (http), determination of chargeable user traffic is possible only through the actual URL (layer 7).

DPI performs packet inspection beyond layer 4 inspection and is typically deployed for:

- Detection of URI information at level 7 (for example, HTTP, WTP, RTSP URLs)
- Identification of true destination in the case of terminating proxies, where shallow packet inspection would only reveal the destination IP address/port number of a terminating proxy such as the OpCo's WAP gateway
- De-encapsulation of nested traffic encapsulation, for example MMS-over-WTP/WSP-over-UDP/IP
- Verification that traffic actually conforms to the protocol the layer 4 port number suggests

Charging Subsystem

ECS has protocol analyzers that examine uplink and downlink traffic. Incoming traffic goes into a protocol analyzer for packet inspection. Routing rules definitions (ruledefs) are applied to determine which packets to inspect. This traffic is then sent to the charging engine where charging rules definitions are applied to perform actions such as block, redirect, or transmit. These analyzers also generate usage records for the billing system.

Traffic Analyzers

Traffic analyzers in ECS are based on configured ruledefs. Ruledefs used for traffic analysis analyze packet flows and create usage records. The usage records are created per content type and forwarded to a prepaid server or to a billing system.

The Traffic Analyzer function can perform shallow (layer 3 and layer 4) and deep (above layer 4) packet inspection of IP packet flows. It is able to correlate all layer 3 packets (and bytes) with higher layer trigger criteria (for example, URL detected in an HTTP header). It also performs stateful packet inspection for complex protocols like FTP, RTSP, and SIP that dynamically open ports for the data path and this way, user plane payload is differentiated into "categories". Traffic analyzers can also detect video streaming over RTSP, and image downloads and MMS over HTTP and differential treatment can be given to the Vcast traffic.

Traffic analyzers work at the application level as well, and perform event-based charging without the interference of the service platforms.

The ECS content analyzers can inspect and maintain state across various protocols at all layers of the OSI stack. ECS supports the following protocols:

- Domain Name System (DNS)
- File Transfer Protocol (FTP)
- Hyper Text Transfer Protocol (HTTP)
- Internet Control Message Protocol (ICMP)
- Internet Control Message Protocol version 6 (ICMPv6)
- Internet Message Access Protocol (IMAP)
- Internet Protocol version 4 (IPv4)
- Internet Protocol version 6 (IPv6)
- Multimedia Messaging Service (MMS)
- Mobile IPv6 (MIPv6)
- Post Office Protocol version 3 (POP3)
- Remote Authentication Dial In User Service (RADIUS)
- RTP Control Protocol/Real-time Transport Control Protocol (RTCP)
- Real-time Transport Protocol (RTP)
- Real Time Streaming Protocol (RTSP)
- Session Description Protocol (SDP)
- Secure-HTTP (S-HTTP)
- Session Initiation Protocol (SIP)
- Simple Mail Transfer Protocol (SMTP)
- Transmission Control Protocol (TCP)
- User Datagram Protocol (UDP)

- WebSocket Protocol
- Wireless Session Protocol (WSP)
- Wireless Transaction Protocol (WTP)

Notes:

- Apart from the above protocols, ECS also supports analysis of downloaded file characteristics (for example, file size, chunks transferred, and so on) from file transfer protocols such as HTTP and FTP.
- Mobile IPv6 (MIPv6) protocol analyzer provides network-based IP mobility management support to a mobile node, without requiring the participation of the mobile node in any IP mobility related signaling. The mobile node may be an IPv4-only node or IPv6-only node.

How ECS Works

This section describes the base components of the ECS solution, and the roles they play.

Content Service Steering

Content Service Steering (CSS) enables directing selective subscriber traffic into the ECS subsystem (in-line services internal to the system) based on the content of the data presented by mobile subscribers.

CSS uses Access Control Lists (ACLs) to redirect selective 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 (for PDSN) or an APN profile (for GGSN) in the destination context.

Important For more information on CSS, refer to the *Content Service Steering* chapter of the *System Administration Guide*. For more information on ACLs, refer to the *IP Access Control Lists* chapter of the *System Administration Guide*.

Protocol Analyzer

The Protocol Analyzer is the software stack responsible for analyzing the individual protocol fields and states during packet inspection.

The Protocol Analyzer performs two types of packet inspection:

- **Shallow Packet Inspection**—Inspection of the layer 3 (IP header) and layer 4 (for example, UDP or TCP header) information.
- **Deep Packet Inspection**—Inspection of layer 7 and 7+ information. DPI functionality includes:
  - Detection of Uniform Resource Identifier (URI) information at level 7 (for example, HTTP, WTP, and RTSP URLs)
  - Identification of true destination in the case of terminating proxies, where shallow packet inspection would only reveal the destination IP address/port number of a terminating proxy
  - De-encapsulation of upper layer protocol headers, such as MMS-over-WTP, WSP-over-UDP, and IP-over GPRS

- Verification that traffic actually conforms to the protocol the layer 4 port number suggests

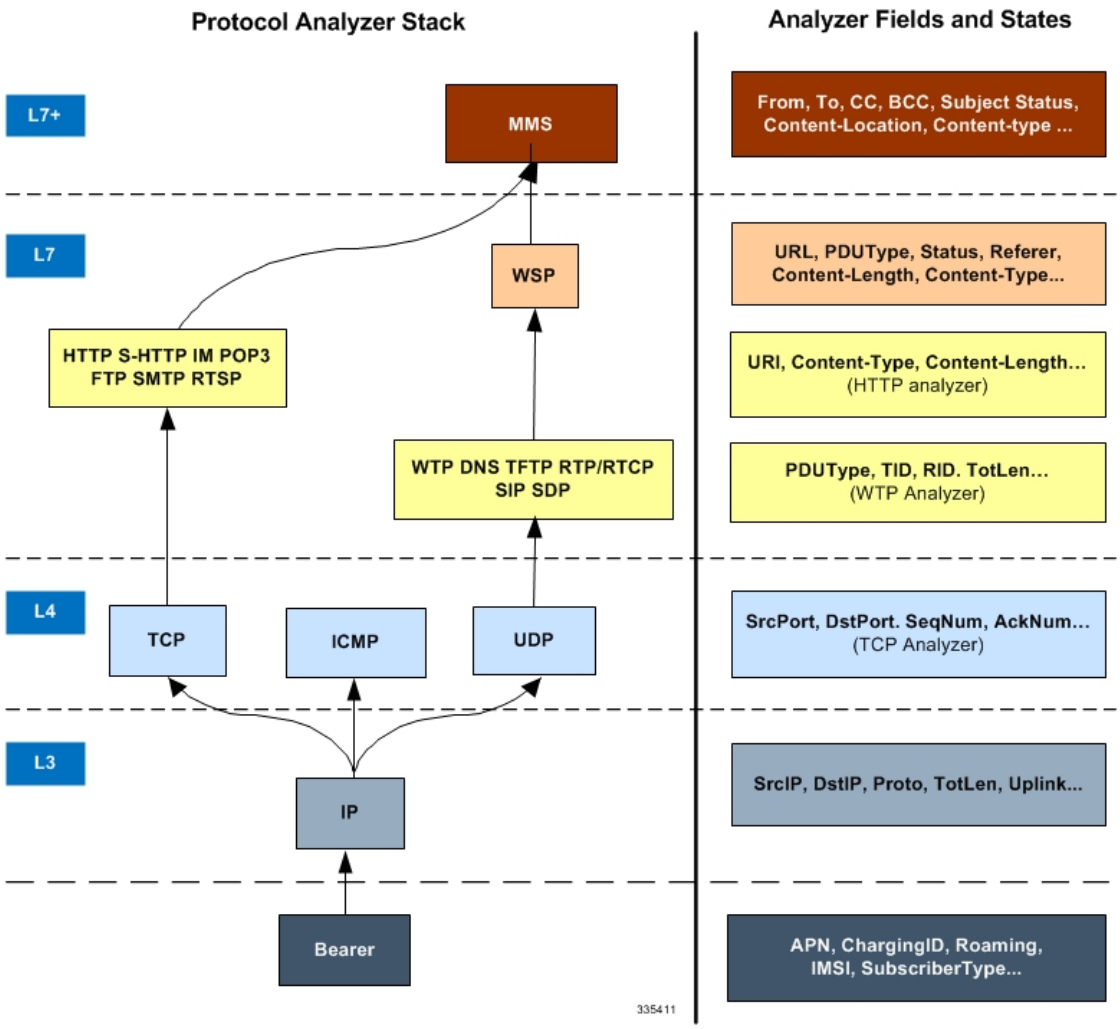
The Protocol Analyzer performs a stateful packet inspection of complex protocols, such as FTP, RTSP, and SIP, which dynamically open ports for the data path, so the payload can be classified according to content.

The Protocol Analyzer is also capable of determining which layer 3 packets belong (either directly or indirectly) to a trigger condition (for example, URL). In cases where the trigger condition cannot be uniquely defined at layers 3 and 4, then the trigger condition must be defined at layer 7 (that is, a specific URL must be matched).

Protocol Analyzer Software Stack

Every packet that enters the ECS subsystem must first go through the Protocol Analyzer software stack, which comprises of individual protocol analyzers for each of the supported protocols.

Figure 1: ECS Protocol Analyzer Stack



Note that protocol names are used to represent the individual protocol analyzers.

Each analyzer consists of fields and states that are compared to the protocol-fields and protocol-states in the incoming packets to determine packet content.

Important In 14.0 and later releases, the ECS HTTP analyzer supports both CRLF and LF as valid terminators for HTTP header fields.

Rule Definitions

Rule definitions (ruledefs) are user-defined expressions based on protocol fields and protocol states, which define what actions to take on packets when specified field values match.

Rule expressions may contain a number of operator types (string, =, >, and so on) based on the data type of the operand. For example, "string" type expressions like URLs and host names can be used with comparison operators like "contains", "!contains", "=", "!=", "starts-with", "ends-with", "!starts-with" and "!ends-with". In 19.2 and later releases, "!present" and "present" operators are added to enhance rule detection on the basis of absence/presence of Accept, Referer, X-header, User-agent, Cookies and Version fields in HTTP header request.

In 14.0 and later releases, ECS also supports regular expression based rule matching. For more information, refer to the *Regular Expression Support for Rule Matching* section.

Integer type expressions like "packet size" and "sequence number" can be used with comparison operators like "=", "!=", ">=", "<=". Each ruledef configuration consists of multiple expressions applicable to any of the fields or states supported by the respective analyzers.

Ruledefs are of the following types:

- **Routing Ruledefs** — Routing ruledefs are used to route packets to content analyzers. Routing ruledefs determine which content analyzer to route the packet to when the protocol fields and/or protocol-states in ruledef expression are true. Up to 256 ruledefs can be configured for routing.
- **Charging Ruledefs** — Charging ruledefs are used to specify what action to take based on the analysis done by the content analyzers. Actions can include redirection, charge value, and billing record emission.

  In releases prior to 21.1: Up to 2048 ruledefs can be configured in the system.

  In 21.1 and later releases: Up to 2500 ruledefs can be configured in the system.
- **Post-processing Ruledefs** — Used for post-processing purposes. Enables processing of packets even if the rule matching for them has been disabled.

Important When a ruledef is created, if the rule-application is not specified for the ruledef, by default the system considers the ruledef as a charging ruledef.

Ruledefs support a priority configuration to specify the order in which the ruledefs are examined and applied to packets. The names of the ruledefs must be unique across the service or globally. A ruledef can be used across multiple rulebases.

Important Ruledef priorities control the flow of the packets through the analyzers and control the order in which the charging actions are applied. The ruledef with the lowest priority number invokes first. For routing ruledefs, it is important that lower level analyzers (such as the TCP analyzer) be invoked prior to the related analyzers in the next level (such as HTTP analyzer and S-HTTP analyzers), as the next level of analyzers may require access to resources or information from the lower level. Priorities are also important for charging ruledefs as the action defined in the first matched charging rule apply to the packet and ECS subsystem disregards the rest of the charging ruledefs.

Each ruledef can be used across multiple rulebases, and up to 2048 ruledefs can be defined in a charging service in releases prior to 21.1. In 21.1 and later releases, up to 2500 ruledefs can be configured in a charging service.

In 15.0 and later releases, a maximum of 32 rule expressions (rule-lines) can be added in one ruledef.

Ruledefs have an expression part, which matches specific packets based upon analyzer field variables. This is a boolean (analyzer_field operator value) expression that tests for analyzer field values.

The following is an example of a ruledef to match packets:

```
http url contains cnn.com
```

–or–

```
http any-match = TRUE
```

In the following example the ruledef named "rule-for-http" routes packets to the HTTP analyzer:

```
route priority 50 ruledef rule-for-http analyzer http
```

Where, **rule-for-http** has been defined with the expressions: **tcp either-port = 80**

The following example applies actions where:

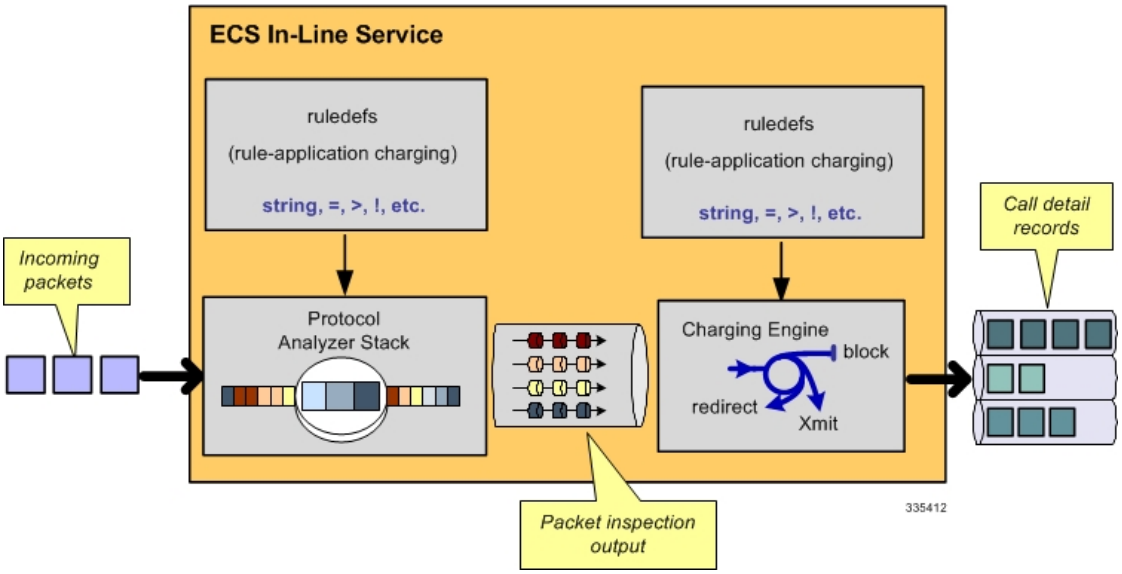
- Subscribers whose packets contain the expression "bbc-news" are not charged for the service.
- All other subscribers are charged according to the duration of use of the service.

```
ruledef port-80
     tcp either-port = 80
     rule-application routing
     exit
ruledef bbc-news
     http url starts-with http://news.bbc.co.uk
     rule-application charging
     exit
ruledef catch-all
     ip any-match = TRUE
     rule-application charging
     exit
charging-action free-site
     content-id 100
     [ ... ]
     exit
charging-action charge-by-duration
     content-id 101
     [ ... ]
     exit
rulebase standard
     [ ... ]
     route priority 1 ruledef port-80 analyzer http
```

```
action priority 101 ruledef bbc-news charging-action free-site
action priority 1000 ruledef catch-all charging-action charge-by-duration
[ ... ]
exit
```

The following figure illustrates how ruledefs interact with the Protocol Analyzer Stack and Action Engine to produce charging records.

Figure 2: ECS In-line Service Processing

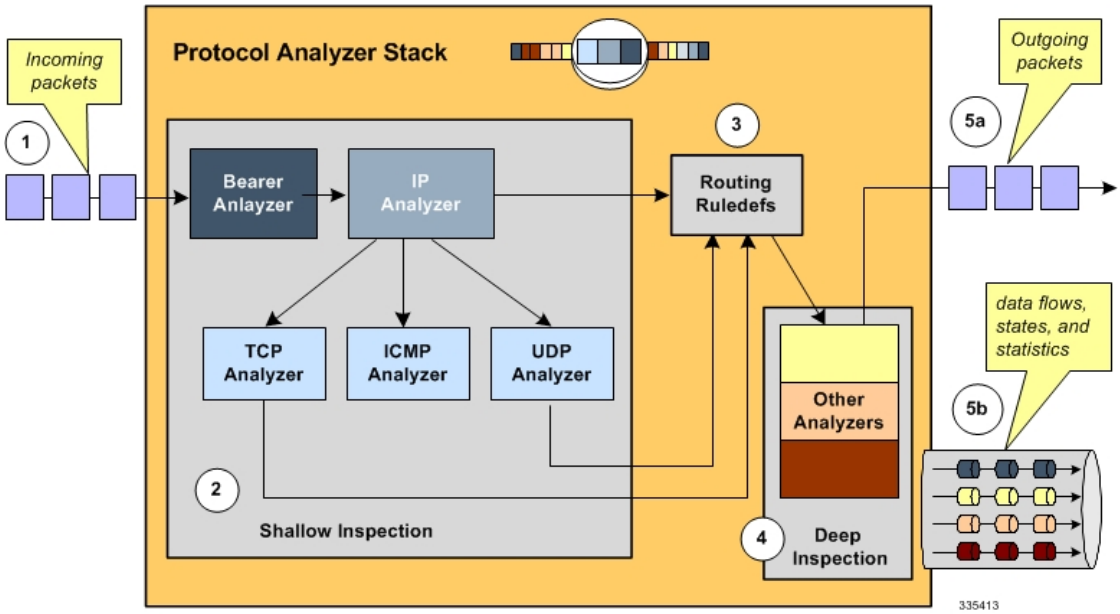


Packets entering the ECS subsystem must first pass through the Protocol Analyzer Stack where routing ruledefs apply to determine which packets to inspect. Then output from this inspection is passed to the charging engine, where charging ruledefs apply to perform actions on the output.

Routing Ruledefs and Packet Inspection

The following figure and the steps describe the details of routing ruledef application during packet inspection.

Figure 3: Routing Ruledefs and Packet Inspection



Step 1 The packet is redirected to ECS based on the ACLs in the subscriber's template /APN and packets enter ECS through the Protocol Analyzer Stack.

Step 2 Packets entering Protocol Analyzer Stack first go through a shallow inspection by passing through the following analyzers in the listed order:

a) Bearer Analyzer
b) IP Analyzer
c) ICMP, TCP, or UDP Analyzer as appropriate

Important In the current release traffic routes to the ICMP, TCP, and UDP analyzers by default. Therefore, defining routing ruledefs for these analyzers is not required.

Step 3 The fields and states found in the shallow inspection are compared to the fields and states defined in the routing ruledefs in the subscriber's rulebase.

The ruledefs' priority determines the order in which the ruledefs are compared against packets.

Step 4 When the protocol fields and states found during the shallow inspection match those defined in a routing ruledef, the packet is routed to the appropriate layer 7 or 7+ analyzer for deep-packet inspection.

Step 5 After the packet has been inspected and analyzed by the Protocol Analyzer Stack:

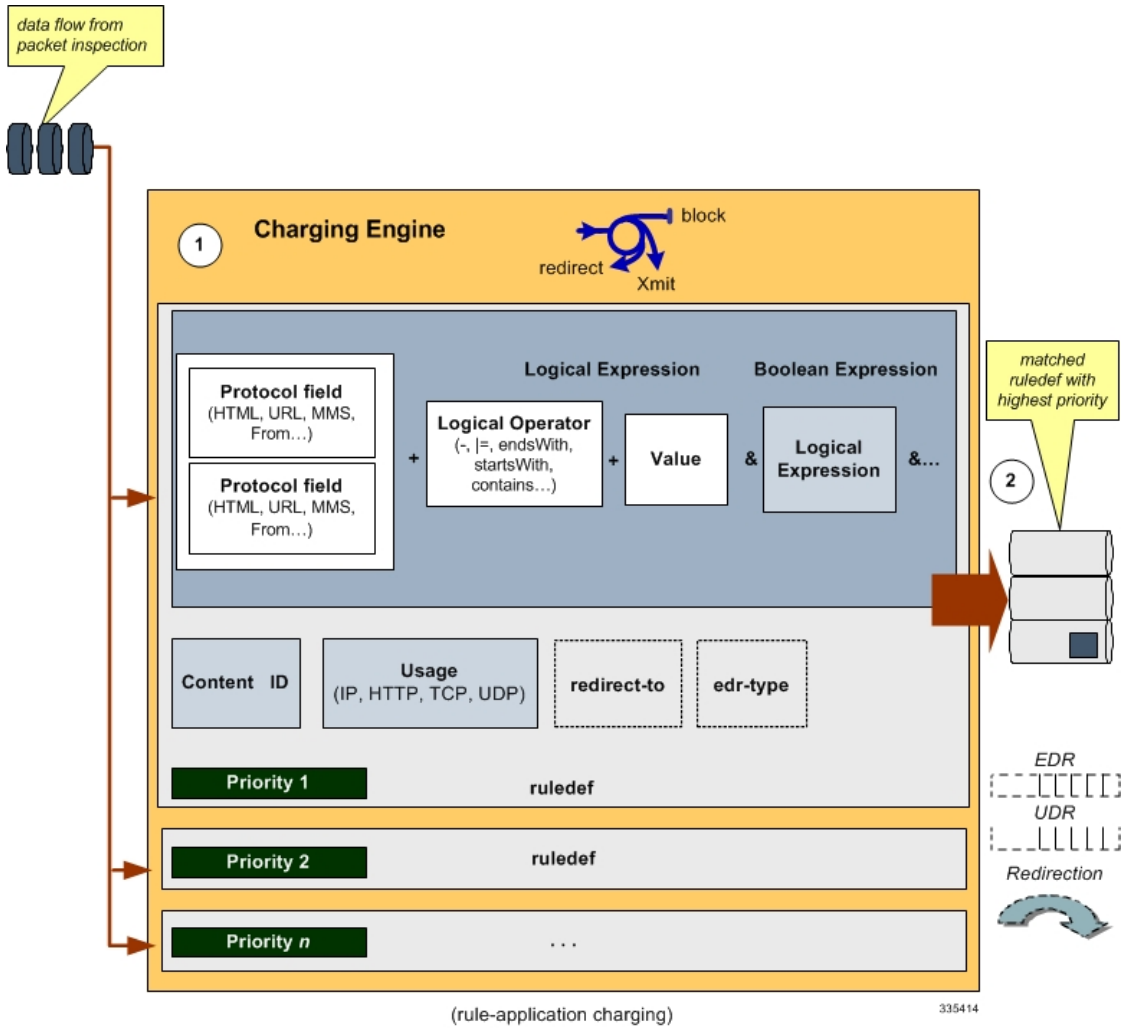
a) The packet resumes normal flow and through the rest of the ECS subsystem.
b) The output of that analysis flows into the charging engine, where an action can be applied. Applied actions include redirection, charge value, and billing record emission.

Charging Ruledefs and the Charging Engine

This section describes details of how charging ruledefs are applied to the output from the Protocol Analyzer Stack.

The following figure and the steps that follow describe the process of charging ruledefs and charging engines.

Figure 4: Charging Ruledefs and Charging Engine



Step 1 In the Classification Engine, the output from the deep-packet inspection is compared to the charging ruledefs. The priority configured in each charging ruledef specifies the order in which the ruledefs are compared against the packet inspection output.

Step 2 When a field or state from the output of the deep-packet inspection matches a field or state defined in a charging ruledef, the ruledef action is applied to the packet. Actions can include redirection, charge value, or billing record emission. It is also possible that a match does not occur and no action will be applied to the packet at all.

Regular Expression Support for Rule Matching

This section describes ECS support for regular expression (regex) rule matching.

In this release, ECS supports regex rule matching only for the following string-based rules:

- http host

- http referer
- http uri
- http url
- rtsp uri
- wsp url
- www url

When rule lines are added or modified, the entire trie is recreated and it mallocs memory for every URL present in the configuration. This leads to huge memory allocation that gets freed once the trie is created.

The following table lists the special characters that you can use in regex rule expressions.

Table 1: Special Characters Supported in Regex Rule Expressions

Regex Character	Description
*	Zero or more characters
+	Zero or more repeated instances of the token preceding the +
?	Match zero or one character **Important** The CLI does not support configuring "?" directly, you must instead use "077". For example, if you want to match the string "xyz<any one character>pqr", you must configure it as: **http host regex** *"xyz077pqr"* In another example, if you want to exactly match the string "url?resource=abc", you must configure it as: **http uri regex** *"url077resource=abc"* Where, the first "\" (backslash) is for the escaping of "?", and then "\077" for specifying "?" to the CLI.
\character	Escaped character
\?	Match the question mark (\<ctrl-v>?) character
\+	Match the plus character
*	Match the asterisk character
\a	Match the alert (ASCII 7) character
\b	Match the backspace (ASCII 8) character
\f	Match the form-feed (ASCII 12) character
\n	Match the new line (ASCII 10) character

Regex Character	Description
\r	Match the carriage return (ASCII 13) character
\t	Match the tab (ASCII 9) character
\v	Match the vertical tab (ASCII 11) character
\0	Match the null (ASCII 0) character
\\	Match the backslash character
Bracketed range [0-9]	Match any single character from the range
A leading ^ in a range	Do not match any in the range. All other characters represent themselves.
.\x##	Any ASCII character as specified in two-digit hex notation. For example, \x5A yields a "Z".
|	Specify OR regular expression operator **Important** When using the regex operator "|" in regex expressions, always wrap the string in double quotes. For example, if you want to match the string pqr OR xyz, you must configure it as: **http host regex** *"pqr|xyz'"*.

The following are some examples of the use of regex characters in rule expressions:

- The following command specifies a regex rule expression using the regex character * (asterisk) to match any of the following or similar values in the HTTP Host request-header field: host1, host101, host23w01.

  http host regex *"host*1"*

- The following command specifies a regex rule expression using the regex character + (plus) to match any of the following or similar values in the HTTP Host request-header field: host1, host101, host23w01.

  http host regex *"host+"*

- The following command specifies a regex rule expression using the regex character \077 (?) to match any of the following or similar values in the HTTP Host request-header field: host101.

  http host regex *"hos\077t101"*

- The following command specifies a regex rule expression using the regex character (escaped character) to match the following value in the HTTP Host request-header field: host?example.

  http host regex *"host\\\077example"*

  The first two \form an escape sequence and \077 is converted to ?. The \? is converted to ? as a character and not a place-holder.

- The following command specifies a regex rule expression using the regex character (escaped backslash character) to match the following value in the HTTP Host request-header field: host*01.

  http host regex *"host*01"*

  The first \ is used as an escape sequence for the second .

- The following command specifies a regex rule expression using the regex character \+ (escaped + character) to match the following value in the HTTP Host request-header field: host+01.

  http host regex *"host\\+01"*

- The following command specifies a regex rule expression using the regex character \\ (escaped backslash character) to match the following value in the HTTP Host request-header field: host\01.

  http host regex *"host\\01"*

- The following command specifies regex rule expression using the regex [0-9] to match any of the following or similar values in the HTTP Host request-header field: hostaBc, hostXyZ, hosthost. Values starting with the word "host" and not containing numbers.

  http host regex *"host[^0-9]"*

- The following command specifies regex rule expression using the regex [a-z] to match any of the following or similar values in the HTTP Host request-header field: hostabc, hostxyz, hosthost. Values starting with the word "host" and containing only lowercase letters.

  http host regex *"host[a-z]"*

- The following command specifies a regex rule expression using the regex | (or) to match either of the following values in the HTTP Host request-header field: host1, host23w01.

  http host regex *"host1|host23w01"*

- The following command defines a regex rule expression to match any of the following or similar values in the RTSP URI string: rtsp://pvs29p.cvf.fr:554/t1/live/Oui17, rtsp://pvs00p.cvf.fr:554/t1/live/Nrj12, rtsp://pvs90p.cvf.fr:554/t1/live/France24_fr.

  rtsp uri regex *"rtsp://pvs([0-9][0-9])p.cvf.fr:554/t1/live/(Gulli|Tf1|Tmc|Nrj12|France24_fr|Oui17)*"*

- The following command defines a regex rule expression to match either of the following values in the WWW URL string: http://tp2.site.com/httppvc_clnsssite.com.wap.symphonieserver.musicwaver.com/, http://134.210.11.13/httppvc_clnsssite.com.wap.symphonieserver.musicwaver.com/.

  www url regex
  "http://(tp2.site.com|134.210.11.3)/httppvc_clnsssite.com.wap.symphonieserver.musicwaver.com/"

- The following command defines a regex rule expression to match any of the following or similar values in the WSP URL string: wsp://home.opera.yahoo.com, wsp://dwld.yahoo.com, wsp://dwld2.yahoo.com.

  wsp url regex *"wsp://(dwld|opera|home.opera|dwld[1-3]).yahoo.com"*

- The following command defines a regex rule expression to match any of the following or similar values in the HTTP URL string: http://yahoo.com, http://www.yahoo.co.in, http://yahoo.com/news.

  http url regex *"(http://|http://www).yahoo.(co.in|com)*"*

- The following command defines a regex rule expression to match any of the following or similar values in the HTTP URI string: http://server19.com/search?form=zip, http://server20.com/search?form=pdf.

  http uri regex *"(http://|http://www).server[0-2][0-9].com/search?form=(pdf|zip)"*

How it Works

This section describes how regex rule matching works.

The following steps describe how regex rule matching works:

1. Regex ruledefs/group-of-ruledefs are configured in the CLI.

   Regex ruledefs are ruledefs that contain regex rule expressions. A ruledef can contain both regex and regular rule expressions.

   Regex group-of-ruledefs are group-of-ruledefs that contain regex ruledefs. A group-of-ruledefs can contain both regex and regular ruledefs.

2. After the regex ruledefs are configured, on the expiry of an internal 30 second timer, building of the regex engines is triggered.

   Note that one regex engine is built per each regex rule expression type.

   Just as with first-time or incremental configurations, SessCtrl/SessMgr recovery/reconciliation also triggers the building of regex engines.

3. The regex engine matches the regex string (specified in the regex expression) against live traffic, and returns all matching ruledefs.

4. The rule matches are then verified with those configured in the rulebase to determine the best matching rule.

Limitations and Dependencies

This section lists known limitations and restrictions to regex rule matching.

- Changes to ruledefs cause the optimization engines to get updated, hence any changes to ruledefs must be done with care. Preferably during low load times.
- Addition, modification, and deletion of regex ruledefs will result in rebuilding of regex engines, which is time consuming and resource intensive. While the engines are being rebuilt, rule-matching based on the old engines and old configurations may yield inconsistent results.

  Addition, modification, and deletion of action priority lines inside the rulebase has no impact on the regex engines. The regex engines remain intact and the removed action priorities from the rulebase are ignored during rule matching. Similarly, addition, modification (adding or removing ruledefs from it), or deletion of a group-of-ruledefs has no impact on regex engines.

- When adding regex ruledefs, use the following guidelines:
  - As per the current implementation, a maximum of 12 ruledefs is supported which contains rule lines as "xyz*" or "*xyz" or "*xyz*" as they are known to consume large memory. Instead, configure Aho-Corasick rules using "starts-with xyz" or "contains xyz" or "ends-with xyz" constructs, which comparatively consume less memory. The "starts-with", "ends-with" and "contains" operators are specially tailored for these types of operations, and work much faster (with lot less memory) than the corresponding "regex xyz*" or "regex *xyz*" operators. Hence, it is recommended that the "starts-with", "ends-with" and "contains" approach be preferred. Every regex rule line which contains "*" increases the memory/performance impact and its use must be avoided as much as possible.
  - Do not configure rules frequently. Push as much configuration as possible simultaneously so that all the regex rules are available for engine building at the same time. Frequent configuration changes may result in infinite loops with wasted memory and CPU cycles.

- Do not configure large number of regex rules as memory utilization will be high depending on the type of regex rules.
- Frequently monitor status of the engine using the **show active-charging regex status { all | instance** <*instance*> **}** CLI command in the Exec Mode. Where <*instance*> is the SessMgr instance number.

- When deleting ruledefs use the following guidelines:
  - Avoid deleting ruledefs at heavy loads, instead remove them from the required rulebases using the **no action priority** <*action_priority*> CLI command in the ACS Rulebase Configuration Mode. Doing so has no impact on regex building, although it uses additional memory there is no impact on traffic processing.
  - Deletion of ruledefs must be done during low load times. As described earlier, it is highly recommended that ruledefs be added, modified, or deleted in bulk, as it results in optimization engine updates.

Group of Ruledefs

Group-of-Ruledefs enable grouping ruledefs into categories. When a group-of-ruledefs is configured in a rulebase and any of the ruledefs within the group matches, the specified charging-action is applied and action instances are not processed further.

A group-of-ruledefs may contain optimizable ruledefs. Whether a group is optimized or not is decided on whether all the ruledefs in the group-of-ruledefs can be optimized, and if the group is included in a rulebase that has optimization turned on.

When a new ruledef is added, it is checked if it is included in any group-of-ruledefs, and whether it requires optimization.

The group-of-ruledefs configuration enables setting the application for the group (group-of-ruledefs-application parameter). When set to gx-alias, the group-of-ruledefs is expanded only to extract the rule names out of it (with their original priority and charging actions) ignoring the field priority set within the group. This is just an optimization over the PCRF to PCEF interface where a need to install/remove a large set of predefined rules at the same time exists. Though this is possible over the Gx interface (with a limit of 256), it requires a large amount of PCRF resources to encode each name. This also increases the message size.

This aliasing function enables to group a set of ruledef names and provides a simple one-name alias that when passed over Gx, as a Charging-Rule-Base-Name AVP, is expanded to the list of names with each rule being handled individually. From the PCEF point of view, it is transparent, as if the PCRF had activated (or deactivated) those rules by naming each one.

In 14.1 and earlier releases, a maximum of 128 ruledefs can be added to a group-of-ruledefs, and a maximum of 64 group-of-ruledefs can be configured.

In 15.0 and later releases, a maximum of 128 ruledefs can be added to a group-of-ruledefs, and a maximum of 128 group-of-ruledefs can be configured.

In 20.1 and later releases, a maximum of 512 ruledefs can be added to a group-of-ruledefs, and a maximum of 384 group-of-ruledefs can be configured.

Important The total number of ruledefs supported for all GoRs must be used with caution due to the high memory impact. Any modifications to the ruledef or GoR configurations beyond the WARN state of the SCT Task memory may have adverse impact on the system.

Rulebase

A rulebase allows grouping one or more rule definitions together to define the billing policies for individual subscribers or groups of subscribers.

A rulebase is a collection of ruledefs and their associated billing policy. The rulebase determines the action to be taken when a rule is matched. A maximum of 512 rulebases can be specified in the ECS service.

It is possible to define a ruledef with different actions. For example, a Web site might be free for postpaid users and charge based on volume for prepaid users. Rulebases can also be used to apply the same ruledefs for several subscribers, which eliminate the need to have unique ruledefs for each subscriber.

Rulebase List

A rulebase list allows grouping one or more rulebases together, enabling the Online Charging System (OCS) to choose the rulebase for a subscriber from the rulebase list.

A rulebase list enables a list of rulebases to be sent to the OCS over the Gy interface using a buffer. The OCS can then select a specific rulebase from the rulebase list, and apply the ruledefs and billing policies associated with that rulebase to subscribers.

Rulebase lists are created and configured in the ACS Configuration Mode. The maximum length of an individual rulebase-list name is 64 bytes. The buffer that stores space-separated rulebase names within a rulebase-list is of 256 bytes.

In 12.3 and earlier releases, a maximum of 20 rulebase lists can be configured per active charging service.

In 14.0 and later releases, a maximum of 128 rulebase lists can be configured per active charging service.

When a subscriber call is connected, the Session Manager provides the list of rulebase names to the OCS, which chooses the rulebase to be used for the subscriber session from the list.

In case the OCS is not reachable, the rulebase configured as the default will be used.

Bulk Statistics Support

The system's support for bulk statistics allows operators to choose which statistics to view and to configure the format in which the statistics is presented. This simplifies the post-processing of statistical data since it can be formatted to be parsed by external, back-end processors.

The system can be configured to collect bulk statistics (performance data) and send them to a collection server (called a receiver). Bulk statistics are statistics that are collected in a group. The individual statistics are grouped by schema. The following schemas are supported by ECS:

- **ECS**: Provides Enhanced Charging Service statistics
- **ECS Rulebase**: Provides Enhanced Charging Service Rulebase statistics

The system supports the configuration of up to 4 sets (primary/secondary) of receivers. Each set can be configured with to collect specific sets of statistics from the various schemas. Statistics can be pulled manually from the chassis or sent at configured intervals. The bulk statistics are stored on the receiver(s) in files.

The format of the bulk statistic data files can be configured by the user. Users can specify the format of the file name, file headers, and/or footers to include information such as the date, chassis host name, chassis uptime, the IP address of the system generating the statistics (available for only for headers and footers), and/or the time that the file was generated.

When the Web Element Manager is used as the receiver, it is capable of further processing the statistics data through XML parsing, archiving, and graphing.

The Bulk Statistics Server component of the Web Element Manager parses collected statistics and stores the information in the PostgreSQL database. If XML file generation and transfer is required, this element generates the XML output and can send it to a Northbound NMS or an alternate bulk statistics server for further processing.

Additionally, if archiving of the collected statistics is desired, the Bulk Statistics server writes the files to an alternative directory on the server. A specific directory can be configured by the administrative user or the default directory can be used. Regardless, the directory can be on a local file system or on an NFS-mounted file system on the Web Element Manager server.

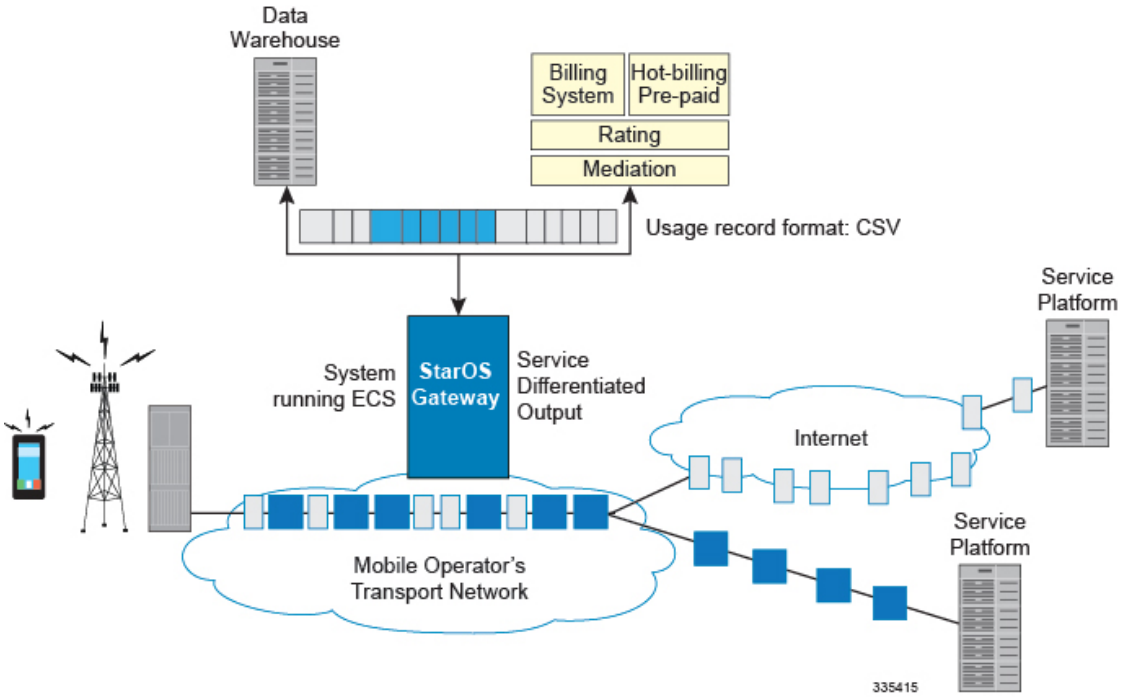
For more information on bulk statistic configuration, refer to the *Configuring and Maintaining Bulk Statistics* chapter in the *System Administration Guide*.

For more information on bulk statistic variables, see the *ECS Schema Statistics* and *ECS Rulebase Schema Statistics* chapter of the *Statistics and Counters Reference*.

ECS Deployment and Architecture

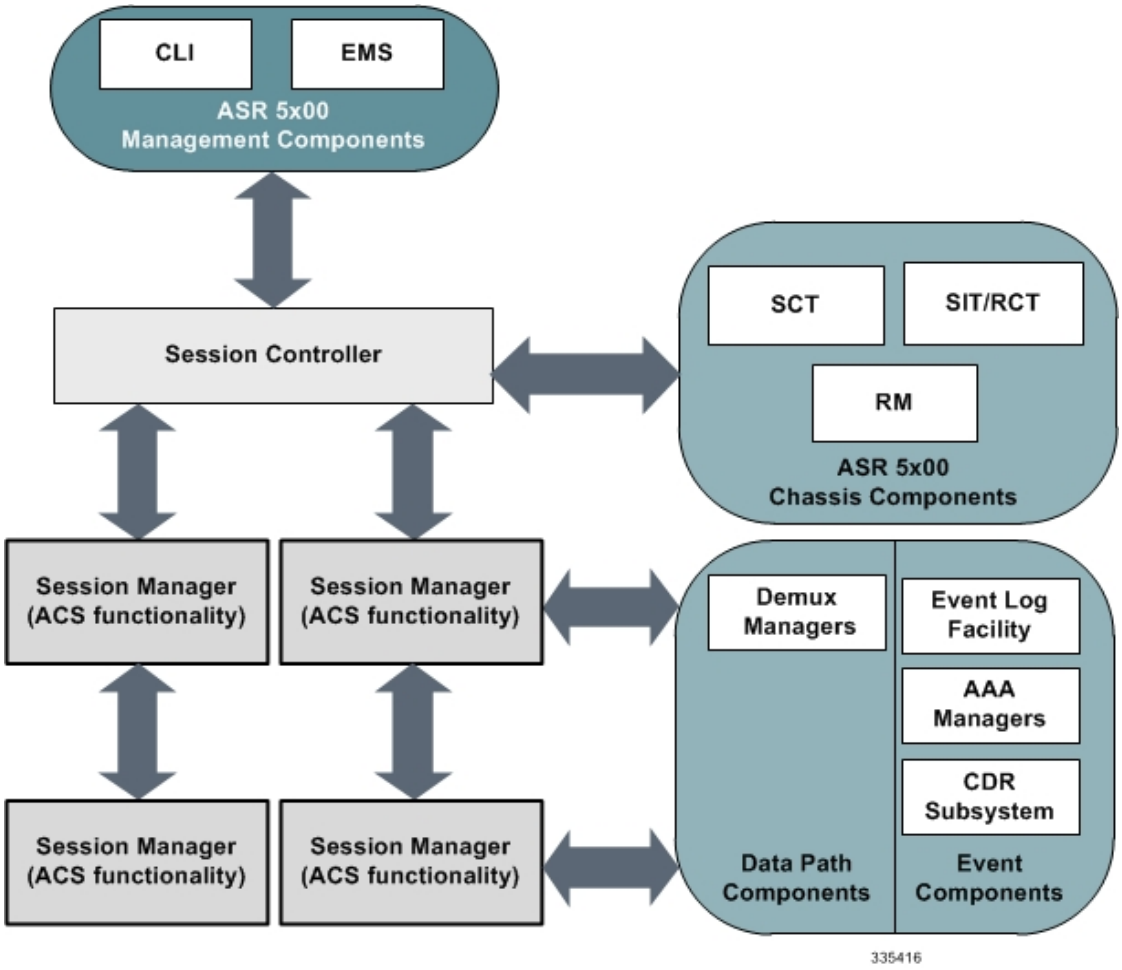
The following figure shows a typical example of ECS deployment in a mobile data environment.

Figure 5: Deployment of ECS in a Mobile Data Network



The following figure depicts the ECS architecture managed by the Session Controller (SessCtrl) and Session Manager (SessMgr) subsystems.

Figure 6: ECS Architecture



Service-Scheme Framework

The Service-scheme framework is introduced to disassociate the dependency with rulebase/PCRF and associate them with policies on the basis of rulebase name, APN name, IMSI range, and so on. This feature offers reduced PCRF dependency that can in turn reduce Gx signaling and integration of any third-party PCRF. The service-scheme framework configuration introduces several CLI commands in support of this feature.

With the previous implementation, the features/policies are always associated with the rulebase and any new proprietary feature for individual subscriber or certain set of subscribers requires support from PCRF. The service-scheme framework helps in overriding this feature behavior for subscribers without involving PCRF. The user can also update the policies specific to subscribers based on pre-configured events.

The subscribers will be classified on the basis of rulebase name, APN name, v-APN name, and so on. These conditions can also be used in combination. If multiple set of conditions are defined for a set of subscribers then conditions with higher priority will be applied for subscriber's selection.

The two main constructs for the new policy framework are listed below:

- Subscriber-base: This helps in associating subscribers with service-scheme based on rule-base, APN name, v-APN name, and so on.
- Service-scheme: This helps in associating trigger actions based on trigger conditions that can be applied on different events at call-setup time, location-update time, flow creation time or any other event triggered through control or data path.

Notes:

- Conflicting actions between PCRF and service-scheme framework against the same trigger events must not be configured.
- If service-scheme is deleted while the call is active, then no new triggers will be processed but existing trigger actions will be applicable for the call duration.
- Any change in the classification of the subscriber will not modify the existing trigger actions for the current active call.
- Any configuration change under subscriber-class condition will be evaluated only for new calls.
- After SR/ICR, the framework will re-evaluate trigger condition configured only under "sess-setup" trigger event.
- Any change under trigger events related to trigger condition and trigger action will depend on the type of trigger event.
  - For session-setup trigger event, the change will be reflected on new calls.
  - For location-update trigger event, the change will be reflected whenever the subscriber changes location.

Refer to the *Configuring Service-scheme Framework* section in the *Enhanced Charging Service Configuration* chapter for more details.

Enhanced Features and Functionality

This section describes enhanced features supported in ECS.

Important The features described in this section may be licensed Cisco features. 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 the *Software Management Operations* chapter in the *System Administration Guide*.

Content Filtering Support

ECS provides offline content filtering support and in-line static and dynamic content filtering support to control static and dynamic data flow and content requests.

Content Filtering Server Group Support

ECS supports external Content Filtering servers through Internet Content Adaptation Protocol (ICAP) implementation between ICAP client and Active Content Filter (ACF) server (ICAP server).

ICAP is a protocol designed to support dynamic content filtering and/or content insertion and/or modification of Web pages. Designed for flexibility, ICAP allows bearer plane nodes such as firewalls, routers, or systems running ECS to interface with external content servers such as parental control (content filtering) servers to provide content filtering service support.

In-line Content Filtering Support

Content Filtering is a fully integrated, subscriber-aware in-line service available for 3GPP and 3GPP2 networks to filter HTTP and WAP requests from mobile subscribers based on the URLs in the requests. This enables operators to filter and control the content that an individual subscriber can access, so that subscribers are inadvertently not exposed to universally unacceptable content and/or content inappropriate as per the subscribers' preferences. Content Filtering uses Deep Packet Inspection (DPI) capabilities of ECS to discern HTTP and WAP requests.

Important For more information on Content Filtering support, refer to the *Content Filtering Services Administration Guide*.

Implementation of AES Encryption

URL redirection is used for user equipment (UE) self-activation, along with pre-paid mobile broadband and other projects.

In the current implementation, when a URL redirection occurs, additional dynamic fields such as MSISDN, IMEI, and username can be appended to the redirection URL for use by the IT portal during the account activation process. StarOS currently supports URL encryption of attributes within the redirection by using Blowfish (64 and 128 bit keys) encryption. It also provides the ability to encrypt either single or multiple concatenated plain text fields. However, Blowfish is no longer considered robust and thus operator now has the option to augment the security of these redirection parameters with a more robust encryption based on AES Encryption.

For URL encryption, AES is an additional option along with Blowfish. The operator has flexibility of choosing the encryption mechanism— Blowfish or AES. This is achieved using CLI and there are no changes done to the dynamic fields. The operator can have different encryption for different rules configurable using CLI.

AES encryption is available for 128 and 256 bit keys. For AES encryption with CBC mode of operation, a key-phrase is taken as configurable field from the operator. This key phrase is internally converted to a 128/256 bit key. An additional field value ("salt") is also allowed as a configurable field. This configurable field is optional.

Security of the subscriber sensitive attributes is enhanced with a more robust encryption algorithm. This helps protect subscriber specific information sent to different servers, thus helping operators to adhere to regulatory policies.

For more information on these commands, see the *ACS Charging Action Configuration Mode Commands* chapter in the *Command Line Interface Reference*.

IP Readdressing

The IP Readdressing feature enables redirecting unknown gateway traffic based on the destination IP address of the packets to known/trusted gateways.

IP Readdressing is configured in the flow action defined in a charging action. IP readdressing works for traffic that matches particular ruledef, and hence the charging action. IP readdressing is applicable to both uplink and downlink traffic. In the Enhanced Charging Subsystem, uplink packets are modified after packet inspection, rule matching, and so on, where the destination IP/port is determined, and replaced with the readdress IP/port just before they are sent out. Downlink packets (containing the readdressed IP/port) are modified as soon as they are received, before the packet inspection, where the source IP/port is replaced with the original server IP/port number.

For one flow from an MS, if one packet is re-addressed, then all the packets in that flow will be re-addressed to the same server. Features like DPI and rule-matching remain unaffected. Each IP address + port combination will be defined as a ruledef.

In case of IP fragmentation, packets with successful IP re-assembly will be re-addressed. However, IP fragmentation failure packets will not be re-addressed.

New hierachy approach has also been provided for selecting the server in case of server list configured under charging-action. This helps the operator to specify list of DNS servers in the order of preference. In hierarchy based approach, queries are redirected as per primary, secondary, and tertiary selection. Both round-robin and hierarchy based server selection approaches would be applicable for both IPv4 and IPv6 based servers. An additional CLI is provided that enables you to select from hierarchy or round-robin approach for server selection. See the *Configuring IP Readdressing* for more information.

Next-hop Address Configuration

ECS supports the ability to set the next-hop default gateway IP address as a charging action associated with any ruledef in a rulebase. This functionality provides more flexibility for service based routing allowing the next-hop default gateway to be set after initial ACL processing. This removes need for AAA to send the next-hop default gateway IP address for CC opted in subscribers.

In 15.0 and later releases, ECS behaves such that rule matching is not done for partial HTTP request if HTTP analysis is enabled.

Assume ECS has received partial HTTP GET packet where URL is not complete, and there are a few URL based rules configured. At this point of time, ECS will not be in a position to match proper rule as complete URL information is not available. When packet where request is completed, is received by ECS, proper rule matching is possible. Earlier partial packets and bytes of this request will be charged accordingly.

Also, this does not apply to post-processing rules. Post-processing rules are matched for all the packets, irrespective of the packet is partial or not. If the customer wants to configure actions like next-hop forwarding or ip-readdressing, then that can be configured in post-processing rules.

In releases prior to 15.0, partial packets do not go for post processing rule match. Whereas in 15.0 and later releases, the partial packets go for required rule match. This behavior change is introduced to obtain the correct statistics about the packets.

How it works:

Step 1 The next-hop address is configured in the charging action.

Step 2 Uplink packet sent to ECS is sent for analysis.

Step 3 When the packet matches a rule and the appropriate charging action is applied, the next-hop address is picked from the charging action and is copied to the packet before sending the packet to Session Manager.

Step 4 Session Manager receives the packet with the next-hop address, and uses it accordingly.

Post Processing

The Post Processing feature enables processing of packets even if the rule matching for them has been disabled. This enables all the IP/TCP packets including TCP handshaking to be accounted and charged for in the same bucket as the application flow. For example, delay-charged packets for IP Readdressing and Next-hop features.

- Readdressing of delay-charged initial hand-shaking packets.
- Sending the delay-charged initial packets to the correct next-hop address.
- DCCA—Taking appropriate action on retransmitted packets in case the quota was exhausted for the previous packet and a redirect request was sent.
  - DCCA with buffering enabled—Match CCA rules, charging-action will decide action—terminate flow/redirect
  - DCCA with buffering disabled—Match post-processing rules, and take action
- Content ID based ruledefs—On rule match, if content ID based ruledef and charging action are present, the rule is matched, and the new charging action will decide the action

A ruledef can be configured as a post-processing rule in the ruledef itself using rule-application of the ruledef. A rule can be charging, routing, or a post-processing rule. If the same ruledef is required to be a charging rule in one rulebase and a post-processing rule in another one, then two separate identical ruledefs must be defined.

How the Post-processing Feature Works

The following steps describe how the Post-processing feature works:

Step 1 Charging rule-matching is done on packets and the associated charging-action is obtained.

Step 2 Using this charging-action the disposition-action is obtained.

Step 3 If the disposition action is to either buffer or discard the packets, or if it is set by the ACF, or if there are no post-processing rules, the packets are not post processed. The disposition action is applied directly on the packets. Only if none of the above conditions is true, post processing is initiated.

Step 4 Post-processing rules are matched and the associated charging-action and then the disposition-action obtained through control-charge.

Step 5 If both match-rule and control-charge for post processing succeed, the disposition-action obtained from post-processing is applied. Otherwise, the disposition-action obtained from charging rule-matching is used.

If no disposition action is obtained by matching post-processing rules, the one obtained by matching charging-rules will be applied.

Irrespective of whether post processing is required or not, even if a single post-processing rule is configured in the rulebase, post processing will be done.

The following points should be considered while configuring post-processing rules for next-hop/readdressing.

- The rules will be L3/L4 based.
- They should be configured in post-processing rules' charging actions.

For x-header insertion, there should either be a post-processing rule whose charging-action gives no disposition-action or the packet should not match any of the post-processing rules so that the disposition action obtained from charging-rule matching is applied.

Pre-defined Rule Retention for Rulebase Change Trigger from Charging Action

Rulebase change is triggered from the Gx, Gy and RADIUS CoA external interfaces, and also from charging action by configuring the rulebase change in the charging action definition. With the old implementation, the rulebase change trigger does not retain predefined rules that are common between the current rulebase and destination rulebase. The predefined rules in all triggers could be deactivated and activated even if they existed in the destination rulebase.

With this release, when rulebase change is triggered through charging action, the predefined rules common between the current rulebase and destination rulebase are retained. This change applies only to the charging-action trigger and no rule retention is done for external triggers - Gx, Gy, and RADIUS CoA interfaces.

The following behavior and limitations are applicable with the rulebase change trigger from charging action:

- Rules will be retained only for rulebase change triggered through charging action.
- Rules with matching rule name and charging action name in the destination rulebase will be retained.
- Rule retention will be applied for ADC rules.

  Only the APP-START event notification will be seen for ADC rules as rule is retained on rulebase change.
- Rule retention will be supported for static-and-dynamic rules only from Gx R7 onwards.
- Rules will not be retained for rulebase change triggered through external interfaces - Gx, Gy, and RADIUS CoA.
- Rules will not be retained for SFW and NAT rules.
- Rules will not be retained for UDR and CDR functionality.
- This is a configuration restriction. In a scenario for rulebase change where current rulebase has a rule with configuration in its charging action to change the current rulebase to new rulebase. If there is the same rulename configured with new rulebase in the same charging action, then this scenario could lead to loop of rulebase change. This is the existing behavior and will not be fixed. Hence, this configuration will not be valid.

RADIUS Based Dual Factor Authentication For Mobile Private Network

Dual Factor Authentication has been implemented for Mobile Private Network's (MPN's) mobile devices, most typically for terminals like lottery machine devices, ATMs, and so on. For security reasons, this DFA procedure is followed before traffic can flow normally. The first level authentication happens as part of call setup using RADIUS. While the call is established, the pre-DFA-rulebase that has the configuration to allow only RADIUS and ICMP traffic is used; rest of the traffic is dropped. Until then all the normal traffic is denied and is resumed only after the additional RADIUS based authentication is successful.

The success of RADIUS authentication is determined by a RADIUS analyzer. This analyzer understands the authentication requests and responses especially 'Access-Request' and 'Access-Accept'. Whenever the RADIUS 'Access-Request' message is matched with 'Access-Accept' message, the rulebase is changed to new rulebase

called Post-DFA-rulebase and the existing dedicated bearers are deleted and the same is informed to PCRF. The RADIUS analyzer does not analyze any other message but only the 'Access-Request', 'Access-Accept', and the 'Access-Reject'.

For the Dual Factor Authentication feature to function, the config pre-DFA-rule-base, the RADIUS analyzer, and the post-DFA-rule-base.

For more information on configuring the Radius Analyzer, see *Configuring RADIUS Analyzer* section in the *Enhanced Charging Service Configuration* chapter.

RAN Bandwidth Optimization

When the rule is installed and active, P-GW uses the GBR/MBR assigned in the rule for calculating the GBR / MBR values towards the bearers created. When more than one rule is installed, P-GW adds the GBR / MBR values from all the active and installed rules even if the flow of a certain rule is marked as disabled. This current behavior is in accordance with 3GPP TS standard specification 29.212, and this might result in RAN bandwidth wastage. To avoid this wastage, some optimization is done while calculating MBR and GBR for GBR bearer.

The RAN bandwidth optimization feature provides the ability to configure a list of APNs, for which the optimized calculation of MBR, GBR can be enabled. By default, this optimized calculation should be enabled only for the IMS APN.

This feature further helps optimize the logic of aggregating MBR and GBR values, based on "Flow-Status" AVP value received in the rule definition through RAR. This operation is controlled through a CLI command **ran bandwidth optimize** added in the ACS Rulebase configuration mode.

For more information on this command, see the *ACS Rulebase Configuration Mode Commands* chapter in the *Command Line Interface Reference*.

Selective TFT Suppression for Default Bearer

With this feature, the selected TFT updates can be controlled and sent to the UE. A new CLI command "**tft-notify-ue**" is introduced, which suppresses the selected TFT updates to the UE. This is provided by specific charging-action level option to identify if the appropriate TFT defined in the charging action needs to be sent to the UE or not. This CLI is supported for both default and dedicated bearer.

One more new CLI "**tft-notify-ue-def-bearer**" to suppress TFTs on default bearer has been added, so the operator has the flexibility to configure this per Rulebase and also configure to suppress TFT updates only. This CLI allows sending other QoS updates to the UE and is only controlling TFT related updates. This CLI is supported only for default bearer.

For more information on these commands, see the *ACS Charging Action Configuration Mode Commands* and *ACS Rulebase Configuration Mode Commands* chapters in the *Command Line Interface Reference*.

Service Group QoS Feature

The Service Group QoS feature enables the chassis/PCEF to define and enforce Fair-Usage-Policy (FUP) per subscriber. This enables changing certain charging-action parameters and all QoS-group-of-ruledefs parameters over the Gx interface per individual subscriber session.

In the chassis/PCEF, the Service Group QOS feature enables to:

- Define service-groups that may include unrelated services defined on the chassis/PCEF.

- Dynamically install pre-defined service-groups for a subscriber over Gx.
- Dynamically remove pre-defined service-groups for a subscriber over Gx.
- Dynamically set and change QoS parameters of a service-group for a subscriber over Gx using CCA and RAR messages. QoS parameters of a service-group (FUP-QoS) are:
  - Flow-Rate
  - Flow-Status
  - Volume Threshold
- Apply Flow-Status to a packet-flow progressively at service and service-group levels. The rules for hierarchical enforcement of Flow-Status rule are:
  - Flow-Status Gating at Service Level: If rule indicates "block" or "redirect", then that action is taken. If rule indicates "allow", then next level's gating rule is applied.
  - Flow-Status Gating at Service-Group Level: If rule indicates "block" or "redirect", then that action is taken. If rule indicates "allow", then next level's gating rule is applied.
- Apply Flow-Rate to a packet-flow progressively at service and service-group levels. Maximum Bit-rate and Burst size is defined by Flow-Rate. Meter the traffic to the configured Flow-Rate and based on the output, apply DSCP marking to the packet.

Important The output action of Flow-Rate can be forward, drop, or mark DSCP. Flow-Rate may allow the packet without DSCP marking.

The rules of hierarchical QOS enforcement are:

- Metering at Service Level: Initially, traffic is metered against service-level QOS rule. If the result of metering marks or drops the packet, then the next level metering is not performed.
- Metering at Service-Group Level: If the packet is allowed at service level, then service-group level QOS metering is done. If the result of metering marks or drops the packet, then the next level metering is not performed.

Important The packet is first subjected to Flow-Status enforcement and if allowed by Flow-Status only then Flow-Rate is enforced. Flow-Status enforcement includes applying Flow-Status progressively at service and service-group Levels. If the flow-status at both levels allows the packet to pass only then it is given for flow-rate enforcement, which applies Flow-Rate progressively at service and service-group levels.

- Monitor volume usage of a group-of-services. Multiple group-of-services can share a volume-quota.
- Provide a mechanism to share configured volume threshold of a service-group across all services in that group. This sharing would be dynamic, that is no predefined quota is allocated per service in a service group.

- Generate a notification to PCRF in a CCR-U message, when volume threshold for a group-of-services is crossed. Once a notification is generated, the trigger is disarmed to generate notification. Continue to monitor usage, but do not report further breaches until PCRF explicitly enables threshold-breach notification trigger in a CCA-U message.
- Report volume usage to PCRF in a CCR-U message when the service-group removed is the last using the shared volume-quota.

Configuration Overview

QoS-group-of-ruledefs are statically configured in the CLI, in the Active Charging Service Configuration Mode. The CLI allows addition and removal of charging and dynamic ruledefs to a named QoS-group-of-ruledefs. A single ruledef can belong to multiple QoS-groups. A maximum of 64 QoS-group-of-ruledefs can be configured in the ACS service. Each QoS-group-of-ruledefs can contain up to 128 ruledefs.

PCRF will be aware of all QoS-group-of-ruledefs names and their constituent ruledefs configured on the chassis/PCEF. The PCRF can activate and remove QoS-group-of-ruledefs for a subscriber session over Gx using a proprietary AVP in CCA and RAR messages. This AVP specifies the name of the QoS-group-of-ruledefs to activate or to remove. Individual ruledefs cannot be dynamically added or removed from a predefined QoS-group-of-ruledefs over the Gx interface. Attributes of QoS-group-of-ruledefs (FUP-QoS parameters) cannot be defined in the CLI. These parameters can only be set and changed over the Gx interface. This feature allows setting different QoS parameters for different subscribers for a named QoS-group-of-ruledefs.

The following attributes of QoS-group-of-ruledefs are supported:

- Precedence or Priority: Priority of a QoS-group-of-ruledefs implies priority of applying QoS-parameters of a QoS-group-of-ruledefs to an incoming data packet. If a packet matches a charging rule which is part of multiple QoS-groups activated for the session, then QoS-parameters of the QoS-group-of-ruledefs with highest priority is applied to the packet. A lower priority number indicates higher priority of application of QoS-parameters of that group. Priority of a QoS-group-of-ruledefs is set by PCRF over Gx for each subscriber session.
- Flow-Status: Can be set to Forward, Block, or Redirect.

Important The Append-Redirect option is not supported.

Important Block can be for uplink, downlink, or both uplink and downlink traffic.

- Flow-Rate: Specifies max rate, max burst-size, conform action, and exceed action; individually for uplink and downlink traffic.
- Usage Monitoring Key: A monitoring key, which has an integer value, is set by PCRF over Gx. Volume threshold values are set for this key by PCRF, to perform usage monitoring. Usage is tracked against a monitoring key.
- Volume Thresholds: The PCRF can set volume threshold values for a monitoring key over Gx. An event is reported when thresholds are crossed, and usage is reported at predefined events — such as session termination and when the QoS-group-of-ruledefs removed is the last using the shared volume-quota.

Important In this release, time thresholds are not supported.

- Attributes of QoS-group-of-ruledefs cannot be defined using CLI. These attributes can only be set and changed over Gx. This allows setting different QoS parameters for different subscribers for a named QoS-group-of-ruledefs.
- When a QoS-group-of-ruledefs is activated, its QoS parameters can be set and changed over Gx. This is achieved using a combination of standard and proprietary AVPs.
- The following attributes of charging-action can be set and changed by PCRF over Gx.
  - Flow-Status: Can be set to Forward, Block, or Redirect.

Important ECS does not support the Append-Redirect option.

Important Block can be for uplink, downlink, or both uplink and downlink traffic.

  - Flow-Rate: Specifies max rate, max burst-size, conform action, and exceed action; individually for uplink and downlink traffic.
  - Volume Threshold: Thresholds are set for usage monitoring. PCRF can set threshold for a monitoring key, which is statically defined for a charging-action using CLI. Usage is reported when thresholds are crossed, and at predefined events such as session termination and removal of QoS-group-of-ruledefs.

Important Monitoring-key is not received over Gx for static charging-action. Triggers for threshold breached are same as the usage-reporting for static charging-action.

- Flow-Status and Flow-Rate can be statically defined for a charging action, and thus applied to a ruledef. These parameters may be overridden by PCRF over Gx. Volume-Threshold-Key (monitoring key) can be statically defined for a ruledef in a rulebase. However, its value — the volume quota — can only be set over Gx. Parameters set over Gx will always take precedence over any static configuration.

Important Time-Monitoring over Gx is not supported in this release.

Support for Service-based QoS

As explained earlier, a service can be mapped in ECS to a set-of-ruledefs with the same charging-action applied to them. This section explains the support for QoS control at the charging-action level:

- Flow-Status: In ECS, you can configure a flow-action in a charging-action. If flow-action is not configured for a charging-action, it implies "Forward" action.

Important Flow-Status value of "Append-Redirect" is not supported by ECS.

- Flow-Rate: ECS charging-action supports configuration of bandwidth limits for a flow. Flow limits can be separately configured for uplink and downlink. ECS supports configuration of peak data-rate and burst-size as well as committed data-rate and burst-size, along with corresponding exceed actions. Specification of committed rate and burst-size is optional.

  ECS does not support specifying conform-action (i.e. conform-action is always "Allow"). For exceeding traffic it supports only "Drop" and "Set IP-TOS to 0" as actions. In ECS, traffic matching a flow — both conforming and exceeding, cannot be marked with a specific DSCP mark.

  In ECS, charging-action also contains a Content-Id. Multiple charging-actions can contain the same Content-Id. ECS supports a bandwidth-limiting meter per charging action per subscriber session. This metering is separate from traffic meters that are keyed on Content-Id.

- Volume Thresholds: ECS supports setting and monitoring Volume Threshold per flow using the "monitoring-key" mechanism. Monitoring-key is specified in a rulebase configuration. Monitoring-key is associated with a volume-threshold, which is set over Gx. A single monitoring-key can be specified for multiple ruledefs. This allows sharing of assigned volume quota across all the ruledefs with the same Monitoring-Key ID. To configure service-level volume quota, you can configure the same monitoring-key for all ruledefs that share the same charging action. Monitoring-Key mechanism enables setting and changing Volume-threshold over Gx.

  In ECS, changing QoS parameters at a service level means changing parameters of a charging-action.

  ECS supports three different kinds of ruledefs:

  - Static rules that are defined in the CLI, and are active immediately after they are defined.
  - Pre-defined rules that are defined in the CLI and activated/deactivated over Gx.
  - Dynamic rules which are defined, activated and deactivated over Gx.

  For static and predefined rules, ECS supports updating per-subscriber FUP parameters of a charging-action over Gx. This is achieved using the Charging-Action-Install AVP. Changes to FUP-parameters of dynamic rules are done using the 3GPP-standard Charging-Rule-Definition AVP.

Hierarchical Enforcement of QoS Parameters

When a packet arrives, ECS performs Deep Packet Inspection and rule matching. If the packet matches a rule, Control-Charge processing is performed as defined by the matched rule. Ruledef-level and QoS-group-of-ruledefs level QoS enforcement are performed as part of Control-Charge processing.

It is not mandatory to set QoS parameters for a ruledef over Gx. If QoS parameters are not set over Gx, then static definition, if any, is enforced. Similarly, for a subscriber session it is not mandatory to group ruledefs in one or more QoS-group-of-ruledefs. A subscriber may not have any QoS-group-of-ruledefs configured. Incoming traffic may match a ruledef, which has no associated QoS-group-of-ruledef for that subscriber session. In that case, action is taken based only on the configuration for that ruledef.

Applying Flow-Status

Flow-Status is applied in a hierarchical manner with the following precedence:

1. Flow Gating at charging-action Level: If flow-action in charging rule indicates "block" or "redirect", then that action is taken. If rule indicates "allow", then next level's gating rule is applied.
2. Flow Gating at QoS-Group-of-Ruledefs Level: Flow-Status specified for the matched QoS-group-of-ruledefs is applied.

Applying Flow-Rate

Hierarchy of metering and marking packet follows the precedence:

1. Metering at Charging-Action Level: Flow-Rate at ruledef level is specified in the charging-action associated with the ruledef. Bandwidth metering specified for the charging-action is first applied to every packet. If the packet conforms to specified bandwidth limits, then QoS-group-of-ruledefs level metering will be performed. If the packet exceeds bandwidth limit at charging-action, then specified exceed action will be taken and bandwidth metering at QoS-group-of-ruledefs and subscriber level will not be performed.
2. Metering at QoS-Group-of-Ruledefs Level: If a packet conforms to charging-action bandwidth limits, then QoS-group-of-ruledefs level bandwidth metering will be done. If the packet conforms to specified bandwidth limits, then subscriber-level metering will be performed. If the packet exceeds bandwidth limit at QoS-group, then specified exceed action will be taken.

Monitoring Usage and Reporting Threshold Breaches

Volume usage is tracked at the charging-action level and at QoS-group-of-ruledefs level. If a received packet causes volume threshold to exceed, then a trigger ECS sends a CCR-U message to PCRF with Service-Group-Event AVP indicating the relevant threshold that was crossed. ECS will then disarm the trigger. If the trigger needs to be rearmed, PCRF will explicitly enable it in the CCA-U message.

In 14 and later releases, Time Reporting over Gx is supported. The time usage is tracked at session/flow level and will be reported to PCRF on meeting certain conditions.

FUP Enforcement for Dynamic Rules

The chassis/PCEF supports dynamic rule installation using 3GPP-standards-based AVPs. The Charging-Rule-Definition AVP is used to install dynamic rules and configure charging behavior and QoS parameters. For dynamic rules, charging-action is part of the rule definition, and not a separate named entity. QoS parameters of a dynamic-rule are changed using the same Charging-Rule-Definition AVP. For dynamic rules, per-service QoS control maps to per-dynamic-rule QoS-control.

- For dynamic rules, service-level QoS control is supported using 3GPP-Standard AVPs. For hierarchical enforcement of FUP parameters for a packet matching a dynamic rule, charging-action level parameters are read from the dynamic rule itself. Hierarchical FUP enforcement will otherwise be similar to that for predefined rules.
- Dynamic rule has a name associated with it. This name can be added to statically (CLI) defined QoS-group-of-ruledefs. So, a dynamic rule can be configured to be part of a QoS-group-of-ruledefs. Multiple dynamic rules can be part of a QoS-group-of-ruledefs. QoS control for a QoS-group-of-ruledefs is transparently enforced, irrespective of whether constituent ruledefs are static, predefined, or dynamically installed.

Reporting Statistics and Usage to PCRF

The PCEF reports volume usage to the PCRF in CCR-U and RAR messages at the following events:

- Volume threshold for a charging-action is crossed, and an event trigger for that threshold breach is set by the PCRF.
- Volume threshold for a QoS-group-of-ruledefs is crossed, and an event trigger for that threshold breach is set by the PCRF.
- A QoS-group-of-ruledefs removed is the last using the shared volume-quota.

Monitoring and reporting of time-usage is not supported in this release. Also, packet drops due to enforcement of FUP-QoS parameters is not reported in CDR.

Statistics pertaining to FUP enforcement are available through Show CLI commands for all active sessions.

Delayed enforcement of bandwidth limiting

As per standards, the gateway enforces the bandwidth limiting based on the configured values. A configurable charging action is provided to allow the carrier to not enforce bandwidth limiting on a flow for a certain duration based on a configurable timer. The charging-action for a packet becomes known after rule-match. The **throttle-suppress** CLI command in the ACS Charging Action Configuration Mode can be configured to suppress bandwidth limiting. for the specified "timeout" period.

When the bandwidth limiting feature is turned on for a flow, the following types of bandwidth limiting will be suppressed:

- ITC bandwidth limiting
- Bearer level bandwidth limiting
- QoS-Group level bandwidth limiting
- APN-AMBR bandwidth limiting for downlink packets only

This section describes the bandwidth limiting behavior with various ECS functionalities:

- **Static and Predefined Ruledef/GoR/QGR:** Static and predefined rules / Group-of-ruledefs / QoS group-of-ruledefs are associated with a charging-action. Hence, all types of bandwidth limiting will be suppressed for the configured "timeout" period.
- **Dynamic Rules:** Dynamic-rules will not trigger throttle-suppress on a flow. However, on an ongoing throttle-suppress flow, dynamic-rule if hit, will suppress the dynamic-rule level bandwidth limiting.
- **TCP OOO Packets:** For TCP OOO packets, bearer bandwidth limiting is applied for packets that are transmitted without reordering. For a given flow, when the TCP OOO packets are processed within the time window of Suppress Start Time and Suppress End Time, the bearer bandwidth limiting will be suppressed.
- **ADC Rules:** For ADC Rules, all the rules across all bearers are matched. When an ADC rule (present on the same or different bearer) is matched and suppress bandwidth limiting is configured, the bandwidth limiting will be suppressed. For non-ADC Rules, the rule matching mechanism considered the rules present only on the bearer on which the flow is attached.
- **TRM:** Throttle-suppress is supported in TRM path.
- **Fast Path / Accelerated-ECS:** For Fast Path (FP) / Accelerated-ECS (A-ECS), bandwidth limiting will be suppressed in the same way as is for normal path.
- **DCCA Buffering:** DCCA buffering remains unaffected with the bandwidth limiting feature.

Session Control in ECS

In conjunction with the Cisco ASR 5500 chassis, the ECS provides a high-level network flow and bandwidth control mechanism in conjunction with the Session Control subsystem. ECS Session Control feature uses the interaction between SessMgr subsystem and Static Traffic Policy Infrastructure support of the chassis to provide an effective method to maximize network resource usage and enhancement of overall user experience.

This feature provides the following functionality:

- **Flow Control Functionality**—Provides the ability to define and manage the number of simultaneous IP-based sessions and/or the number of simultaneous instances of a particular application permitted for the subscriber.

  If a subscriber begins a packet data session and system is either pre-configured or receives a subscriber profile from the AAA server indicating the maximum amount of simultaneous flow for a subscriber or an application is allowed to initiate. If subscriber exceeds the limit of allowed number of flows for subscriber or type of application system blocks/redirect/discard/terminate the traffic.

  The following type of flow quotas are available for Flow Control Functionality:

  - **Subscriber-Level Session Quota**—Configurable on a per-rulebase basis
  - **Application-Level Session Quota**—Configurable on a per-charging-action basis

- **Bandwidth Control Functionality**—Allows the operator to apply rate limit to potentially bandwidth intensive and service disruptive applications.

  Using this feature the operator can police and prioritize subscribers' traffic to ensure that no single or group of subscribers' traffic negatively impacts another subscribers' traffic.

  For example, if a subscriber is running a peer-to-peer (P2P) file sharing program and the system is pre-configured to detect and limit the amount of bandwidth to the subscriber for P2P application. The system gets the quota limit for bandwidth from PDP context parameter or individual subscriber. If the subscriber's P2P traffic usage exceeds the pre-configured limit, the Session Control discards the traffic for this subscriber session.

  Session Control feature in ECS also provides the controls to police any traffic to/from a subscriber/application with the chassis.

Support for Splash Pages

The Splash Page support feature helps to distinguish HTTP traffic coming from mobile browsers and redirect the very first flow to a splash page whenever a subscriber attaches to the network. Splash page is the page of a website that the user sees first before being given the option to continue to the main content of the site.

When a subscriber attaches to the network, PCRF installs predefined rule/group of ruledefs towards PCEF to match mobile browser specific flows. On the first match of this rule/group of ruledef, redirect packet containing information of the welcome page where the flow needs to be redirected, is sent to UE and the first request gets terminated. Subsequently the predefined rule/group of ruledef from the list is removed and sends CCR-U with Charging Rule Report (CRR) AVP to PCRF for rule status. The existing **deactivate-predefined-rule** CLI command in the ACS Charging Action configuration mode is used to remove the matched predefined rule/group of ruledef.

In releases prior to 19.2, the redirection functionality was supported in the case when 80% threshold usage is reached for the subscriber and the same rule gets deactivated to ensure one time redirection for the subscriber.

In this release, the functionality is extended to redirect the first mobile browser flow to the splash page whenever subscriber attaches to the network. This feature now supports predefined group of ruledefs in addition to previously supported predefined rules. CLI and Statistics are enhanced to support HTTP-based rule matching in HTTP header.

Support for WebSocket Protocol Identification

This feature extends support for WebSocket Protocol identification.

The Websocket protocol is an independent TCP based protocol. A connection is identified as Websocket through the first HTTP Get Request header after the three way handshake. This packet includes an upgrade header (Upgrade: websocket) and other Websocket headers (Sec-Websocket-*) to upgrade HTTP to Websocket protocol. This helps operators to categorize Websocket traffic and apply different policies for such traffic.

A new CLI **websocket flow-detection** has been implemented at rulebase level to detect the Websocket protocol. The Websocket protocol identification can be enabled or disabled with the new CLI Websocket protocol.

For more information on these commands, see the *ACS Rulebase Configuration Mode Commands* chapter in the *Command Line Interface Reference*.

How it Works

This section describes how Websocket Protocol Identification feature works.

If the Websocket detection is enabled in the rulebase, the ECS parser looks for the following fields in the HTTP Get header fields **Host**, **Upgrade**, **Connection**, **Sec-Websocket-Key**, **Origin**, and **Sec-Websocket-Version**. If these headers are present, the TCP connection is upgraded to a Websocket connection. A ruledef can be defined to identify the HTTP GET request for the websocket and rate it in a certain way. The subsequent data that is transferred through the websocket is also billed the same way as the first packet.

Important You need to enable HTTP analysis to detect Websockets, and Websocket connections cannot be detected on secure-HTTP connections.

TCP Proxy

The TCP Proxy feature enables the ASR 5500 to function as a TCP proxy. TCP Proxy along with other capabilities enables the ASR 5500 to transparently split every TCP connection passing through it between sender and receiver hosts into two separate TCP connections, and relay data packets from the sender host to the receiver host via the split connections. Any application that needs to modify the TCP payload or manage TCP connection establishment and tear down uses the TCP Proxy feature.

TCP Proxy is enabled dynamically based on specified conditions. When TCP proxy is started dynamically on a flow, the original client (MS) first starts the TCP connection with the final server. ECS keeps on monitoring the connection. Based on any rule-match/charging-action, it may happen that the connection will be proxied automatically. This activity is transparent to original client and original server. The functional/charging behavior of ECS for that particular connection before the dynamic proxy is started is exactly same as when there is no proxy.

TCP Proxy impacts post-recovery behavior and the charging model. With TCP Proxy, whatever packets are received from either side is charged completely. The packets that are sent out from the ECS are not considered for charging. This approach is similar to the behavior of ECS without proxy.

The following packets will be charged at ECS:

- Uplink packets received at Gn interface
- Downlink packets received at Gi interface

The following packets will not be considered for charging:

- Uplink packets forwarded/sent out by ECS/Stack on the Gi interface
- Downlink packets forwarded/sent out by ECS/Stack on the Gn interface

ECS supports bulk statistics for the TCP Proxy feature. For details see the *ECS Schema Statistics* chapter of the *Statistics and Counters Reference*.

Flow Admission Control

The Flow Admission Control feature controls the number of flows required to be proxied. It restricts admission of new calls based on the current resource usage, thus preventing system hog and service degradation to existing subscribers.

The number of flows required to be proxied will greatly depend on the deployment scenario. Operators have the provision to configure an upper bound on the memory used by proxy flows. This is specified as a percentage of the Session Manager memory that may be used for proxy flows. When memory utilization by existing proxy flows reaches this value, no further flows will be proxied.

Operators can also set a limit on the number of flows that can be proxied per subscriber. This would exercise Fair Usage policy to a certain extent. No credit usage information by proxy is communicated to the Session Manager.

TCP Proxy Behavior and Limitations

The following are behavioral changes applicable to various ECS features and on other applications after enabling TCP Proxy.

- TCP Proxy Model: Without TCP Proxy, for a particular flow, there is only a single TCP connection between subscriber and server. ECS is a passive entity with respect to flows and the packets received on ingress were sent out on egress side (except in case where some specific actions like drop are configured through CLI) transparently.

  With TCP Proxy, a flow is split into two TCP connections — one between subscriber and proxy and another between chassis and server.

- Ingress Data Flow to Proxy: For all uplink packets, ingress flow involves completing the following steps and then enters the Gn side TCP IP Stack of proxy:

  1. IP Analysis (support for IP reassembly)
  2. Shallow/Deep Packet TCP Analysis (support for TCP OOO)
  3. Stateful Firewall Processing
  4. Application Detection and Control Processing

5. DPI Analysis
6. Charging Function (including rule-matching, generation of various records, and applying various configured actions)

For all downlink packets, ingress flow would involve completing the following steps, and then enters the Gi side TCP IP Stack of proxy:

1. IP Analysis (support for IP reassembly)
2. Network Address Translation Processing
3. Shallow/Deep Packet TCP Analysis (support for TCP OOO)
4. Stateful Firewall Processing
5. Application Detection and Control Processing
6. DPI Analysis
7. Charging Function (including rule-matching, generation of various records, and applying various configured actions)

- Egress Data Flow from Proxy: All egress data flow is generated at proxy stack. For uplink packets, egress data flow would involve the following and then are sent out of the chassis:

  1. IP Analysis
  2. Shallow/Deep Packet TCP Analysis
  3. Stateful Firewall processing
  4. Network Address Translation processing

  For downlink packets, egress data flow would involve the following and then are sent out of the chassis:

  1. IP Analysis
  2. Shallow/Deep Packet TCP Analysis
  3. Stateful Firewall processing

On enabling TCP Proxy the behavior of some ECS features will get affected. For flows on which TCP Proxy is enabled it is not necessary that all the packets going out of the Gn (or Gi) interface are the same (in terms of number, size, and order) as were on Gi (or Gn).

- IP Reassembly: If the fragments are successfully reassembled then DPI analysis is done on the reassembled packet.

  Without TCP Proxy, fragmented packets will go out on the other side. With TCP proxy, normal (non-fragmented) IP packets will go out on the other side (which will not be similar to the incoming fragmented packets).

  With or without TCP Proxy, if fragment reassembly was not successful, then all the fragments will be dropped except under the case where received fragments were sufficient enough to identify the 5-tuple TCP flow and the flow had TCP Proxy disabled on it.

- TCP OOO Processing: Without TCP Proxy if it is configured to send the TCP OOO packets out (as they come), without TCP proxy such packets were sent out. With TCP Proxy, OOO packets coming from one

side will go in-order on the other side. For proxied flows TCP OOO expiry timer will not be started and hence there will be no specific handling based on any such timeouts. Also, TCP OOO packets will not be sent to other side unless the packets are re-ordered.

In releases prior to 14.0, when TCP Out-of-Order (OOO) packets were received and when there was any error in buffering those packets at ECS due to memory allocation failure, these packets were marked as TCP error packets and the rule matching was done accordingly. These packets were also marked as TCP error packets when the reordering packet was not received before the OOO timeout.

In 14.0 and later releases, in the above mentioned scenarios, the packets are not considered as TCP error and the TCP error flag is not set for OOO packets. So, these packets will not match TCP error related ruledef but match other appropriate ruledefs.

If the customer has configured TCP error related rules, then OOO timeout failure packets and memory allocation failure packets will not match these rules now. It will match normal TCP rules.

- TCP Checksum Validation: Without TCP Proxy TCP Checksum validation is optional (configurable through "transport-layer-checksum verify-during-packet-inspection tcp" CLI command). With TCP Proxy TCP checksum is automatically done irrespective of whether the CLI command is configured or not. If the checksum validation fails, the packet is not processed further and so it does not go for application layer analysis.
- TCP Reset Packet Validation: Without TCP Proxy TCP reset packet is not validated for Seq and ACK number present in the segment and the flow is cleared immediately.

  With TCP Proxy TCP Reset packet validation is done. The flow will be cleared only if a valid TCP Reset segment is arrived. This validation is not configurable.
- TCP Timestamp (PAWS) Validation: Without TCP Proxy timestamp verification is not performed and even if there is any timestamp error, the packet is processed normally and goes for further analysis and rule-matching.

  With TCP Proxy if the connection is in established state, timestamp validation for packets is performed. If TCP timestamp is less than the previous timestamp, the packet is marked TCP error packet and is dropped. The packet is not analyzed further and not forwarded ahead. This packet should match TCP error rule (if configured). This validation is not configurable.
- TCP Error Packets: Without TCP Proxy ECS being a passive entity, most of the errors (unless configured otherwise) were ignored while parsing packets at TCP analyzer and were allowed to pass through. With TCP Proxy TCP error packets are dropped by Gi and Gn side TCP IP stack. However, since the ECS processing is already done before giving the packet to the stack, these packets are charged but not sent out by proxy on the other end.
- Policy Server Interaction (Gx): With TCP Proxy, application of policy function occurs on two separate TCP connections (non-proxy processed packets on Gn/Gi). Only external packets (the ones received from Radio and Internet) will be subject to policy enforcement at the box. This does not have any functional impact.
- Credit Control Interaction (Gy): With TCP Proxy, application of Credit Control function occur on two separate TCP connections (non-proxy processed packets on Gn/Gi). Only external packets (the ones received from Radio and Internet) will be subject to credit control at the box. This does not have any functional impact.
- DPI Analyzer: With TCP Proxy, application of DPI analyzer occurs on two separate TCP connections (non-proxy processed packets on Gn/Gi). Only external packets (the ones received from Radio and

Internet) will be subject to DPI analyzer at the chassis. Any passive analyzer in the path would be buffering packet using the existing ECS infrastructure.

- ITC/BW Control: With TCP Proxy, only incoming traffic is dropped based on bandwidth calculation on ingress side packets. The BW calculation and dropping of packet is be done before sending packet to ingress TCP IP Stack. ToS and DSCP marking will be on flow level. The ToS and DSCP marking can be done only once for whole flow and once the ToS is marked for any packet either due to "ip tos" CLI command configured in the charging action or due to ITC/BW control, it will remain same for the whole flow.
- Next Hop and VLAN-ID: Without TCP Proxy nexthop feature is supported per packet, that is nexthop address can be changed for each and every packet of the flow depending on the configuration in the charging action. With TCP Proxy only flow-level next-hop will be supported. So, once the nexthop address is changed for any packet of the flow, it will remain same for the complete flow. The same is the case for VLAN-ID.
- TCP state based rules: Without TCP Proxy there is only one TCP connection for a flow and the TCP state based rules match to state of subscriber stack. With TCP Proxy there are two separate connections when TCP proxy is enabled. TCP state ("tcp state" and "tcp previous-state") based rules will match to MS state on egress side. Two new rules (tcp proxy-state and tcp proxy-prev-state) have been added to support the existing cases (of TCP state based rules). "tcp proxy-state" and "tcp proxy-prev-state" are the state of the embedded proxy server, that is the proxy ingress-side. These rules will not be applicable if proxy is not enabled.

  Using both "tcp state" and "tcp proxy-state" in the same ruledef is allowed. If proxy is enabled, they would map to Gi-side and Gn-side, respectively. If TCP Proxy is not enabled, the "tcp proxy-state" and "tcp proxy-prev-state" rules will not be matched because proxy-state will not be applicable.

  Since TCP state and previous-state rules are now matched based on state on Gi side connection, ECS will not be able to support all the existing use-cases with the existing configuration. New ruledefs based on the new rules (tcp proxy-state and tcp proxy-prev-state) need to be configured to support existing use cases. Note that even by configuring using new rules; all use-cases may not be supported. For example, detection of transition from TIME-WAIT to CLOSED state is not possible now.
- TCP MSS: TCP IP Stack always inserts MSS Field in the header. This causes difference in MSS insertion behavior with and without TCP Proxy.
  - TCP CFG MSS limit-if-present: If incoming SYN has MSS option present, in outgoing SYN and SYN-ACK MSS value is limited to configured MSS value (CFG MSS)
  - TCP CFG MSS add-if-not-present: If incoming SYN does not have MSS option present, in outgoing SYN and SYN-ACK MSS configured MSS value is inserted (CFG MSS)
  - TCP CFG MSS limit-if-present add-if-not-present: If incoming SYN has MSS option present, in outgoing SYN and SYN-ACK MSS value is limited to configured MSS value (CFG MSS), OR if incoming SYN does not have MSS option present, in outgoing SYN and SYN-ACK MSS configured MSS value is inserted (CFG MSS).
- Flow Discard: Flow discard occurring on ingress/egress path of TCP Proxy would be relying on TCP-based retransmissions. Any discard by payload domain applications would result in data integrity issues as this might be charged already and it may not be possible to exclude packet. So it is recommended that applications in payload domain (like dynamic CF, CAE readdressing) should not be configured to drop packets. For example, dynamic content filtering should not be configured with drop action. If drop is absolutely necessary, it is better to use terminate action.

- DSCP/IP TOS Marking: Without TCP Proxy DSCP/IP TOS marking is supported per packet, that is IP TOS can be changed for each and every packet of the flow separately based on the configuration. With TCP Proxy flow-level DSCP/IP TOS marking is supported. So, once the IP TOS value is changed for any packet of the flow, it will remain same for the complete flow.
- Redundancy Support (Session Recovery and ICSR): Without TCP Proxy after recovery, non-syn flows are not reset. With TCP Proxy session recovery checkpointing is bypassing any proxied flows (currently on NAT flows support recovery of flows). If any flow is proxied for a subscriber, after recovery (session recovery or ICSR), if any non-syn packet is received for that subscriber, ECS sends a RESET to the sender. So, all the old flows will be RESET after recovery.
- Charging Function: Application of charging function would occur on two separate TCP connections (non proxy processed packets on Gn/Gi). Only external packets (the ones received from Radio and Internet) shall be subject to Policy enforcement at the box. Offline charging records generated at charging function would pertain to different connections hence.

Dynamic Disabling of TCP Proxy

TCP proxy can be dynamically disabled to reduce the performance overhead on CPU and memory resources. This enables applications to use proxy only when required.

Dynamic disabling is achieved by merging the TCP connections. Before dynamic disabling occurs, the packets are added to a TCP stack with a full proxy connection. Once proxy is disabled dynamically, the TCP stack and proxy are removed from the data processing path and the packets are forwarded without buffering.

Disabling of TCP proxy dynamically occurs only after the following conditions are met:

- There is no data to be delivered by ECS to the peer.
- The flow control buffers do no contain any data.
- There is no data to be read by ECS.

Limitations for Dynamically Disabling TCP Proxy

This section lists known limitations to disabling TCP proxy dynamically:

- TCP proxy cannot be disabled when one end of the TCP supports time stamp and other does not.
- Dynamic disabling does not work when both sides of the TCP have different MSS negotiated.
- Toggling the proxy on the same connection might reduce TCP performance.
- TCP proxy can only be disabled when both ends of TCP are in connected states.
- Multiple connections (1:n) connections cannot be joined together.
- TCP proxy can only be disabled when the conditions outlined for dynamic disabling is achieved (when there is no unAcked data in the network).

Time and Flow-based Bearer Charging in ECS

ECS supports Time-based Charging (TBC) to charge customers on either actual consumed time or total session time usage during a subscriber session. TBC generates charging records based on the actual time difference between receiving the two packets, or by adding idle time when no packet flow occurs.

ECS also supports Flow-based Charging (FBC) based on flow category and type.

PDP context charging allows the system to collect charging information related to data volumes sent to and received by the MS. This collected information is categorized by the QoS applied to the PDP context. FBC

integrates a Tariff Plane Function (TPF) to the charging capabilities that categorize the PDP context data volume for specific service data flows.

Service data flows are defined by charging rules. The charging rules use protocol characteristics such as:

- IP address
- TCP port
- Direction of flow
- Number of flows across system
- Number of flows of a particular type

FBC provides multiple service data flow counts, one each per defined service data flow. When FBC is configured in the ECS, PDP context online charging is achieved by FBC online charging using only the wildcard service data flow.

When further service data flows are specified, traffic is categorized, and counted, according to the service data flow specification. You can apply wildcard to service data flow that do not match any of the specific service data flows.

The following are the chargeable events for FBC:

- **Start of PDP context**—Upon encountering this event, a Credit Control Request (CCR) starts, indicating the start of the PDP context, is sent towards the Online Charging Service. The data volume is captured per service data flow for the PDP context.
- **Start of service data flow**—An interim CCR is generated for the PDP context, indicating the start of a new service data flow, and a new volume count for this service data flow is started.
- **Termination of service data flow**—The service data flow volume counter is closed, and an interim CCR is generated towards the Online Charging Service, indicating the end of the service data flow and the final volume count for this service data flow.
- **End of PDP context**—Upon encountering this event, a CCR stop, indicating the end of the PDP context, is sent towards the Online Charging Service together with the final volume counts for the PDP context and all service data flows.
- **Expiration of an operator configured time limit per PDP context**—This event triggers the emission of an interim CCR, indicating the elapsed time and the accrued data volume for the PDP context since the last report.
- **Expiration of an operator configured time limit per service data flow**—The service data flow volume counter is closed and an interim CCR is sent to the Online Charging Service, indicating the elapsed time and the accrued data volume since the last report for that service data flow. A new service data flow container is opened if the service data flow is still active.
- **Expiration of an operator configured data volume limit per PDP context**—This event triggers the emission of an interim CCR, indicating the elapsed time and the accrued data volume for the PDP context since the last report.
- **Expiration of an operator configured data volume limit per service data flow**—The service data flow volume counter is closed and an interim CCR is sent to the Online Charging Service, indicating the elapsed time and the accrued data volume since the last report for that service data flow. A new service data flow container is opened if the service data flow is still active.

- **Change of charging condition**—When QoS change, tariff time change are encountered, all current volume counts are captured and sent towards the Online Charging Service with an interim CCR. New volume counts for all active service data flows are started.
- **Administrative intervention** by user/service also force trigger a chargeable event.

The file naming convention for created xDRs (EDR/UDR/FDRs) are described in Impact on xDR File Naming, on page 54.

Time-of-Day Activation/Deactivation of Rules

Within a rulebase, ruledefs/groups-of-ruledefs are assigned priorities. When packets start arriving, as per the priority order, every ruledef/group-of-ruledefs in the rulebase is eligible for matching regardless of the packet arrival time. By default, the ruledefs/groups-of-ruledefs are active all the time.

The Time-of-Day Activation/Deactivation of Rules feature uses time definitions (timedefs) to activate/deactivate static ruledefs/groups-of-ruledefs such that they are available for rule matching only when they are active.

Important The time considered for timedef matching is the system's local time.

How the Time-of-Day Activation/Deactivation of Rules Feature Works

The following steps describe how the Time-of-Day Activation/Deactivation of Rules feature enables charging according to the time of the day/time:

Step 1 Timedefs are created/deleted in the ACS Configuration Mode.

A maximum of 10 timedefs can be created in an ECS service.

Step 2 Timedefs are configured in the ACS Timedef Configuration Mode. Within a timedef, timeslots specifying the day/time for activation/deactivation of rules are configured.

A maximum of 24 timeslots can be configured in a timedef.

Step 3 In the ACS Rulebase Configuration Mode, timedefs are associated with ruledefs /groups-of-ruledefs along with the charging action.

One timedef can be used with several ruledefs/group-of-ruledefs. If a ruledef/group-of-ruledefs does not have a timedef associated with it, it will always be considered as active.

Step 4 When a packet is received, and a ruledef/group-of-ruledefs is eligible for rule matching, if a timedef is associated with the ruledef/group-of-ruledefs, before rule matching, the packet-arrival time is compared with the timeslots configured in the timedef. If the packet arrived in any of the timeslots configured in the associated timedef, rule matching is undertaken, else the next ruledef/group-of-ruledefs is considered.

This release does not support configuring a timeslot for a specific date.

If, in a timeslot, only the time is specified, that timeslot will be applicable for all days.

If for a timeslot, "start time" > "end time", that rule will span the midnight. That is, that rule is considered to be active from the current day until the next day.

If for a timeslot, "start day" > "end day", that rule will span over the current week till the end day in the next week.

In the following cases a rule will be active all the time:

- A timedef is not configured in an action priority
- A timedef is configured in an action priority, but the named timedef is not defined
- A timedef is defined but with no timeslots

URL Filtering

The URL Filtering feature simplifies using rule definitions for URL detection.

The following configuration is currently used for hundreds of URLs:

```
ruledef HTTP://AB-WAP.YZ
      www url starts-with HTTP://CDAB-SUBS.OPERA-MINI.NET/HTTP://AB-WAP.YZ
      www url starts-with HTTP://AB-WAP.YZ
      multi-line-or all-lines
      exit
```

In the above ruledef:

- The HTTP request for the URL "http://ab-wap.yz" is first sent to a proxy "http://cdab-subs.opera-mini.net/".
- The URL "http://cdab-subs.opera-mini.net/" will be configured as a prefixed URL.

Prefixed URLs are URLs of the proxies. A packet can have a URL of the proxy and the actual URL contiguously. First a packet is searched for the presence of proxy URL. If the proxy URL is found, it is truncated from the parsed information and only the actual URL (that immediately follows it) is used for rule matching and EDR generation.

The group-of-ruledefs can have rules for URLs that need to be actually searched (URLs that immediately follow the proxy URLs). That is, the group-of-prefixed-URLs will have URLs that need to be truncated from the packet information for further ECS processing, whereas, the group-of-ruledefs will have rules that need to be actually searched for in the packet.

URLs that you expect to be prefixed to the actual URL can be grouped together in a group-of-prefixed-URLs. A maximum of 64 such groups can be configured. In each such group, URLs that need to be truncated from the URL contained in the packet are specified. Each group can have a maximum of 10 such prefixed URLs. By default, all group-of-prefixed-URLs are disabled.

In the ECS rulebase, you can enable/disable the group-of-prefixed-URLs to filter for prefixed URLs.

Important A prefixed URL can be detected and stripped if it is of the type "http://www.xyz.com/http://www.abc.com". Here, "http://www.xyz.com" will be stripped off. But in "http://www.xyz.com/www.abc.com", it cannot detect and strip off "http://www.xyz.com" as it looks for occurrence of "http" or "https" within the URL.

Accounting and Charging Interfaces

ECS supports different accounting and charging interfaces for prepaid and postpaid charging and record generation.

Important Some feature described in this section are licensed Cisco features. 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 the *Software Management Operations* chapter in the *System Administration Guide*.

Accounting Interfaces for Postpaid Service: ECS supports the following accounting interfaces for postpaid subscribers

- Remote Authentication Dial-In User Service (RADIUS) Interface
- GTPP Accounting Interface

Accounting and Charging Interface for Prepaid Service: ECS supports the following Credit Control Interfaces for prepaid subscribers

- RADIUS Prepaid Credit Control interface
- Diameter Prepaid Credit Control Application (DCCA) Gy Interface
- Diameter Gx interface

Charging Records in ECS: ECS provides the following charging records for postpaid and prepaid charging

- GGSN-Call Detail Records (G-CDRs)
- Enhanced GGSN-Call Detail Records (eG-CDRs)
- Event Detail Records (EDRs)
- Usage Detail Records (UDRs)

GTPP Accounting

ECS enables the collection of counters for different types of data traffic, and including that data in CDRs that is sent to a Charging Gateway Function (CGF).

For more information on GTPP accounting, 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 *GTPP Interface Administration and Reference*.

RADIUS Accounting and Credit Control

The Remote Authentication Dial-In User Service (RADIUS) interface in ECS is used for the following purposes:

- **Subscriber Category Request**—ECS obtains the subscriber category from the AAA server (either prepaid or postpaid) when a new data session is detected. The AAA server used for the subscriber category request can be different from the AAA server used for service authorization and accounting.
- **Service Access Authorization**—ECS requests access authorization for a specific subscriber and a newly detected data session. The AAA server is the access Policy Decision Point and the ECS the Policy Enforcement Point.

- **On-line Service Accounting (Prepaid)**—ECS reports service usage to the AAA server. The AAA server acts as a prepaid control point and the ECS as the client. Accounting can be applied to a full prepaid implementation or just to keep ECS updated of the balance level and trigger a redirection if the subscriber balance reaches a low level.

Diameter Accounting and Credit Control

The Diameter Credit Control Application (DCCA) is used to implement real-time online or offline charging and credit control for a variety of services, such as network access, messaging services, and download services.

In addition to being a general solution for real-time cost and credit control, DCCA includes these features:

- **Real-time Rate Service Information:** DCCA can verify when end subscribers' accounts are exhausted or expired; or deny additional chargeable events.
- **Support for Multiple Services:** DCCA supports the usage of multiple services within one subscriber session. Multiple service support includes:
  - The ability to identify and process the service or group of services that are subject to different cost structures.
  - Independent credit control of multiple services in a single credit control sub-session.

Gx Interface Support

The Gx interface is used in IMS deployment in GPRS/UMTS networks. Gx interface support on the system enables wireless operators to intelligently charge the services accessed depending on the service type and parameters with rules. It also provides support for IP Multimedia Subsystem (IMS) authorization in a GGSN service. The goal of the Gx interface is to provide network-based QoS control as well as dynamic charging rules on a per bearer basis for an individual subscriber. The Gx interface is in particular needed to control and charge multimedia applications.

Important For more information on Gx interface support, see the *Gx Interface Support* appendix in the administration guide for the product that you are deploying.

Gy Interface Support

The Gy interface provides a standardized Diameter interface for real-time content-based charging of data services. It is based on the 3GPP standards and relies on quota allocation.

It provides an online charging interface that works with the ECS Deep Packet Inspection feature. With Gy, customer traffic can be gated and billed in an "online" or "prepaid" style. Both time- and volume-based charging models are supported. In all these models, differentiated rates can be applied to different services based on shallow or deep-packet inspection.

Gy is a Diameter interface. As such, it is implemented atop, and inherits features from, the Diameter Base Protocol. The system supports the applicable base network and application features, including directly connected, relayed or proxied DCCA servers using TLS or plain text TCP.

In the simplest possible installation, the system will exchange Gy Diameter messages over Diameter TCP links between itself and one "prepay" server. For a more robust installation, multiple servers would be used.

These servers may optionally share or mirror a single quota database so as to support Gy session failover from one server to the other. For a more scalable installation, a layer of proxies or other Diameter agents can be introduced to provide features such as multi-path message routing or message and session redirection features.

The Diameter Credit Control Application (DCCA) which resides as part of the ECS manages the credit and quota for a subscriber.

Important For more information on Gy interface support, see the *Gy Interface Support* appendix in the administration guide for the product that you are deploying.

Event Detail Records (EDRs)

Event Detail Records (EDRs) are usage records with support to configure content information, format, and generation triggers by the system administrative user.

EDRs are generated according to explicit action statements in rule commands. Several different EDR schema types, each composed of a series of analyzer parameter names, are specified in EDR. EDRs are written at the time of each event in CSV format. EDRs are stored in timestamped files that can be downloaded via SFTP from the configured context.

EDRs are generated on per flow basis, and as such they catch whatever bytes get transmitted over that flow including retransmitted.

EDR format

The EDRs can be generated in comma separated values (CSV) format as defined in the traffic analysis rules.

Important In EDRs, the maximum field length for normal and escaped strings is 127 characters. If a field's value is greater than 127 characters, in the EDR it is truncated to 127 characters. In 15 and later releases, an optional filter "length" is supported for HTTP URL and User-Agent fields which when added will allow the user to configure length from 1 to 255 for these fields in EDRs. For more information, see the **rule-variable** command in the *Command Line Interface Reference*. In 17 and later releases, the allowed length for HTTP URL is 1 through 4095. For more information, see the **rule-variable** command in the *Command Line Interface Reference*.

In 21.1 and later releases, a maximum of 75 EDR attribute fields can be configured in an EDR record. The limit is expanded from 50 fields up to 75 fields.

Flow-overflow EDR

Flow-overflow EDR or Summary FDR is a feature to count the data bytes from the subscriber that are missed due to various reasons in ECS.

In case any condition that affects the callline (FLOW end-condition like hagr, handoff) occurs, flow-overflow EDR generation is enabled, an extra EDR is generated. Based on how many bytes/packets were transferred from/to the subscriber for which ECS did not allocate data session. This byte/packet count is reflected in that extra EDR. This extra EDR is nothing but "flow-overflow" EDR or Summary FDR.

The extra EDR is generated if all of the following is true:

- Subscriber affecting condition occurs (session-end, hand-off, hagr)
- Flow-overflow EDR generation is enabled

- EDR generation on session-end, hand-off or hagr is enabled
- Number of bytes/packets for flow-overflow EDR is non-zero.

The bytes/packet count will be printed as a part of "sn-volume-amt" attribute in the EDR. Hence, this attribute must be configured in the EDR format.

EDR Generation in Flow-end and Transaction Complete Scenarios with sn-volume Fields

"sn-volume-amt" counters will be re-initialized only when the fields are populated in EDRs. For example, consider the following two EDR formats:

```
edr-format edr1
     rule-variable http url priority 10
     attribute sn-volume-amt ip bytes uplink priority 500
     attribute sn-volume-amt ip bytes downlink priority 510
     attribute sn-volume-amt ip pkts uplink priority 520
     attribute sn-volume-amt ip pkts downlink priority 530
     attribute sn-app-protocol priority 1000
     exit
edr-format edr2
     rule-variable http url priority 10
     attribute sn-app-protocol priority 1000
     exit
```

"sn-volume-amt counters" will be re-initialized only if these fields are populated in the EDRs. Now if edr2 is generated, these counters will not be re-initialized. These will be re-initialized only when edr1 is generated. Also, note that only those counters will be re-initialized which are populated in EDR. For example, in the following EDR format:

```
edr-format edr3
     rule-variable http url priority 10
     attribute sn-volume-amt ip bytes uplink priority 500
     attribute sn-volume-amt ip bytes downlink priority 510
     attribute sn-app-protocol priority 1000
     exit
```

If edr3 is generated, only uplink bytes and downlink bytes counter will be re-initialized and uplink packets and downlink packets will contain the previous values till these fields are populated (say when edr1 is generated).

For the voice call duration for SIP reporting requirements, ECS SIP analyzer keeps timestamp of the first INVITE that it sees. It also keeps a timestamp when it sees a 200 OK for a BYE. When this 200 OK for a BYE is seen, SIP analyzer triggers creation of an EDR of type ACS_EDR_VOIP_CALL_END_EVENT. This will also be triggered at the time of SIP flow termination if no 200 OK for BYE is seen. In that case, the last packet time will be used in place of the 200 OK BYE timestamp. The EDR generation logic calculates the call duration based on the INVITE and end timestamps, it also accesses the child RTP/RTCP flows to calculate the combined uplink/downlink bytes/packets counts and sets them in the appropriate fields.

The HTTP URL and HTTP User Agent can be configured in the EDR in two methods:

- Default, where no length is defined in the EDR configuration. The following is an example EDR configuration where the corresponding variable name for HTTP URL/HTTP User Agent will be "http-url" and "http-user-agent" respectively.

  ```
  rule-variable http url priority 2
  ```

  ```
  rule-variable http user-agent priority 2
  ```

- Length for HTTP URL/HTTP User Agent is defined in the EDR configuration. The following is an example EDR configuration where the corresponding variable name for HTTP URL/HTTP User Agent

will be "http-url-2000" and "http-user-agent-100" respectively. The length for HTTP URL is any number between 1 and 4095 (xyz) that translates into a EDR attribute "http-url-xyz", where xyz is the number entered by the user. The length for HTTP User Agent is any number between 1 and 255 (xyz) that translates into a EDR attribute "http-user-agent-xyz", where xyz is the number entered by the user.

```
rule-variable http url length 2000 priority 4
rule-variable http user-agent length 100 priority 4
```

Usage Detail Records (UDRs)

Usage Detail Records (UDRs) contain accounting information based on usage of service by a specific mobile subscriber. UDRs are generated based on the content-id for the subscriber, which is part of charging action. The fields required as part of usage data records are configurable and stored in the System Configuration Task (SCT).

UDRs are generated on any trigger of time threshold, volume threshold, handoffs, and call termination. If any of the events occur then the UDR subsystem generates UDRs for each content ID and sends to the CDR module for storage.

UDR format

The UDRs are generated in Comma Separated Values (CSV) format as defined in the traffic analysis rules.

Charging Methods and Interfaces

This section provides an overview of the Charging methods and interfaces.

Prepaid Credit Control

Prepaid billing operates on a per content-type basis. Individual content-types are marked for prepaid treatment. A match on a traffic analysis rule that has a prepaid-type content triggers prepaid charging management.

In prepaid charging, ECS performs the metering function. Credits are deducted in real time from an account balance or quota. A fixed quota is reserved from the account balance and given to the system by a prepaid rating and charging server, which interfaces with an external billing system platform. The system deducts volume from the quota according to the traffic analysis rules. When the subscriber's quota gets to the threshold level specified by the prepaid rating and charging server, system sends a new access request message to the server and server updates the subscriber's quota. The charging server is also updated at the end of the call.

ECS supports the following credit control applications for prepaid charging:

- **RADIUS Credit Control Application**—RADIUS is used as the interface between ECS and the prepaid charging server. The RADIUS Prepaid feature of ECS is separate to the system-level Prepaid Billing Support and that is covered under a different license key.
- **Diameter Credit Control Application**—The Diameter Credit Control Application (DCCA) is used to implement real-time credit control for a variety of services, such as networks access, messaging services, and download services.

In addition to being a general solution for real-time cost and credit control, DCCA includes the following features:

- **Real-time Rate Service Information**—DCCA can verify when end subscribers' accounts are exhausted or expired; or deny additional chargeable events.

- **Support for Multiple Services**—DCCA supports the usage of multiple services within one subscriber session. Multiple service support includes:
  - The ability to identify and process the service or group of services that are subject to different cost structures.
  - Independent credit control of multiple services in a single credit control sub-session.

Postpaid

In a postpaid environment, the subscribers pay after use of the service. AAA/RADIUS server is responsible for authorizing network nodes to grant access to the user, and the CDR system generates G-CDRs/eG-CDRs/EDRs/UDRs for billing information on pre-defined intervals of volume or per time.

Important G-CDRs and eG-CDRs are only available in UMTS networks.

ECS also supports FBC and TBC methods for postpaid billing. For more information on FBC and TBC in ECS, see Time and Flow-based Bearer Charging in ECS, on page 37.

Prepaid Billing in ECS

In a prepaid environment, the subscribers pay for service prior to use. While the subscriber is using the service, credit is deducted from subscriber's account until it is exhausted or call ends. The prepaid charging server is responsible for authorizing network nodes to grant access to the user, as well as grant quotas for either time connected or volume used. It is up to the network node to track the quota use, and when these use quotas run low, the network node sends a request to the prepaid server for more quota.

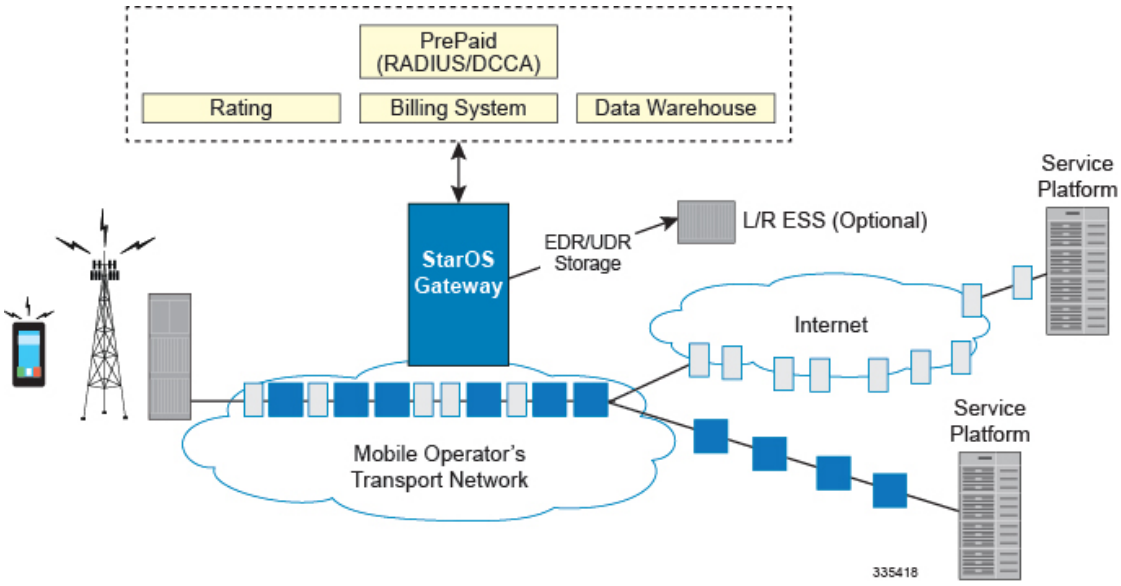
If the user has not used up the purchased credit, the server grants quota and if no credit is available to the subscriber the call will be disconnected. ECS and DCCA manage this functionality by providing the ability to set up quotas for different services.

Prepaid quota in ECS is implemented using RADIUS and DCCA as shown in the following figure.

How ECS Prepaid Billing Works

The following figure illustrates a typical prepaid billing environment with system running ECS.

Figure 7: Prepaid Billing Scenario with ECS



Credit Control Application (CCA) in ECS

This section describes the credit control application that is used to implement real-time credit-control for a variety of end user services such as network access, Session Initiation Protocol (SIP) services, messaging services, download services, and so on. It provides a general solution to the real-time cost and credit control.

CCA with RADIUS or Diameter interface uses a mechanism to allow the user to be informed of the charges to be levied for a requested service. In addition, there are services such as gaming and advertising that may debit from a user account.

How Credit Control Application (CCA) Works for Prepaid Billing

The following figure and steps describe how CCA works with in a GPRS/UMTS or CDMA-2000 network for prepaid billing.

Figure 8: Prepaid Charging in GPRS/UMTS/CDMA-2000 Networks

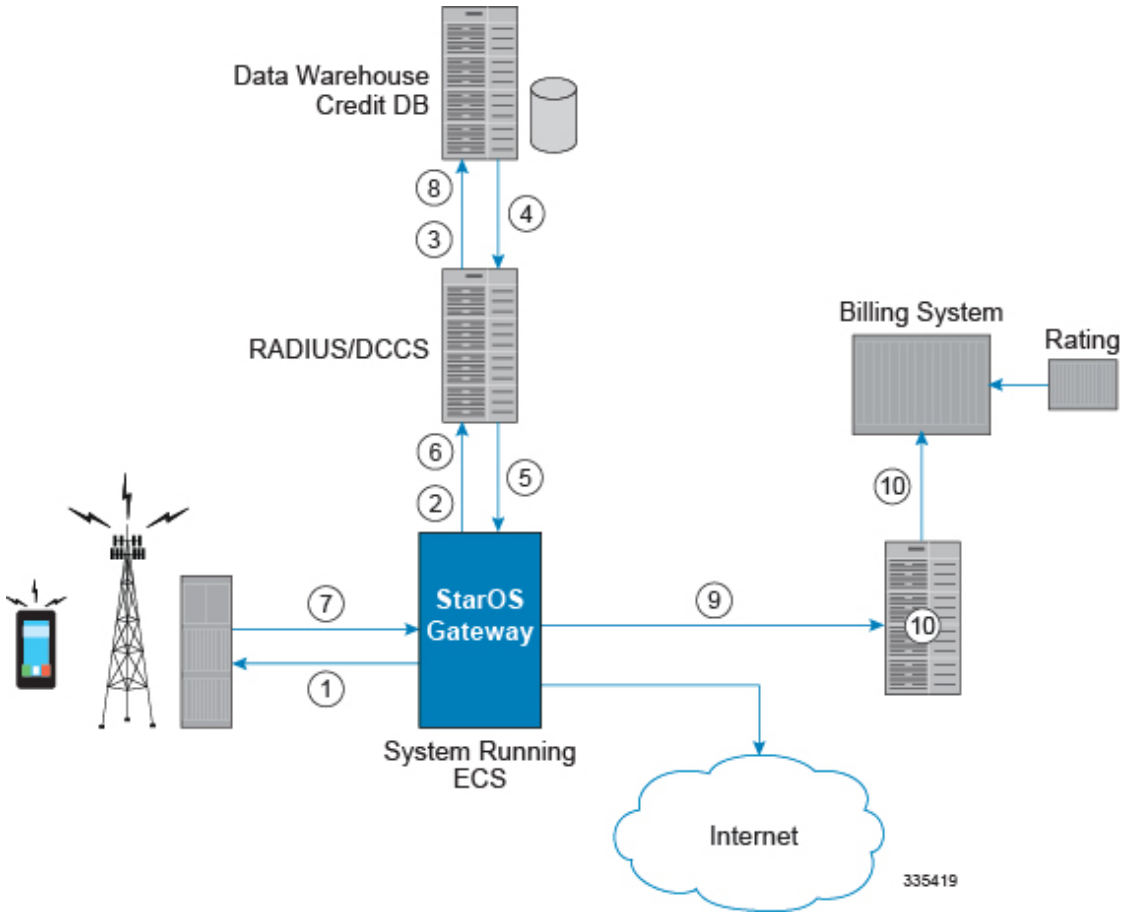


Table 2: Prepaid Charging in GPRS/UMTS/CDMA-2000 Networks

Step No.	Description
1	Subscriber session starts.
2	System sends request to CCA for subscriber's quota.
3	CCA sends request to Data Warehouse (DW) credit quota for subscriber.
4	Credit Database in DW sends pre-configured amount of usage limit from subscriber's quota to CCA. To reduce the need for multiple requests during subscriber's session configured amount of usage limit a major part of available credit quota for subscriber is set.
5	CCA sends the amount of quota required to fulfill the subscriber's initial requirement to the system.

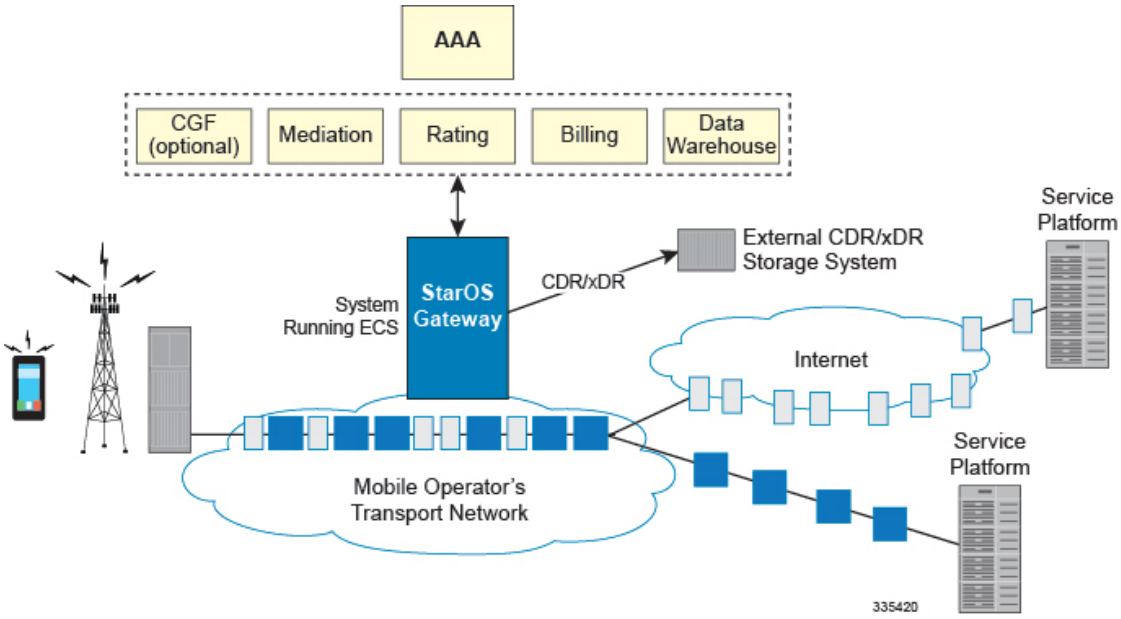
Step No.	Description
6	When the initial amount of quota runs out, system sends another request to the CCA and the CCA sends another portion of available credit quota.
7	Subscriber session ends after either quota exhausts for subscriber or subscriber terminates the session.
8	CCA returns unused quota to DW for update to subscribers Credit DB.
9	EDRs and UDRs are periodically SFTPd from system memory to the external storage, if deployed or to billing system directly as they are generated. Or, if configured, pushed to the external storage at user-configurable intervals.
10	The external storage periodically sends records to the billing system or charging reporting and analysis system.

Postpaid Billing in ECS

This section describes the postpaid billing that is used to implement offline billing processing for a variety of end user services.

The following figure shows a typical deployment of ECS for postpaid billing system.

Figure 9: Postpaid Billing System Scenario with ECS



How ECS Postpaid Billing Works

This section describes how the ECS postpaid billing works in the GPRS/UMTS and CDMA-2000 Networks.

ECS Postpaid Billing in GPRS/UMTS Networks

The following figure and steps describe how ECS works in a GPRS/UMTS network for postpaid billing.

Figure 10: Postpaid Billing with ECS in GPRS/UMTS Network

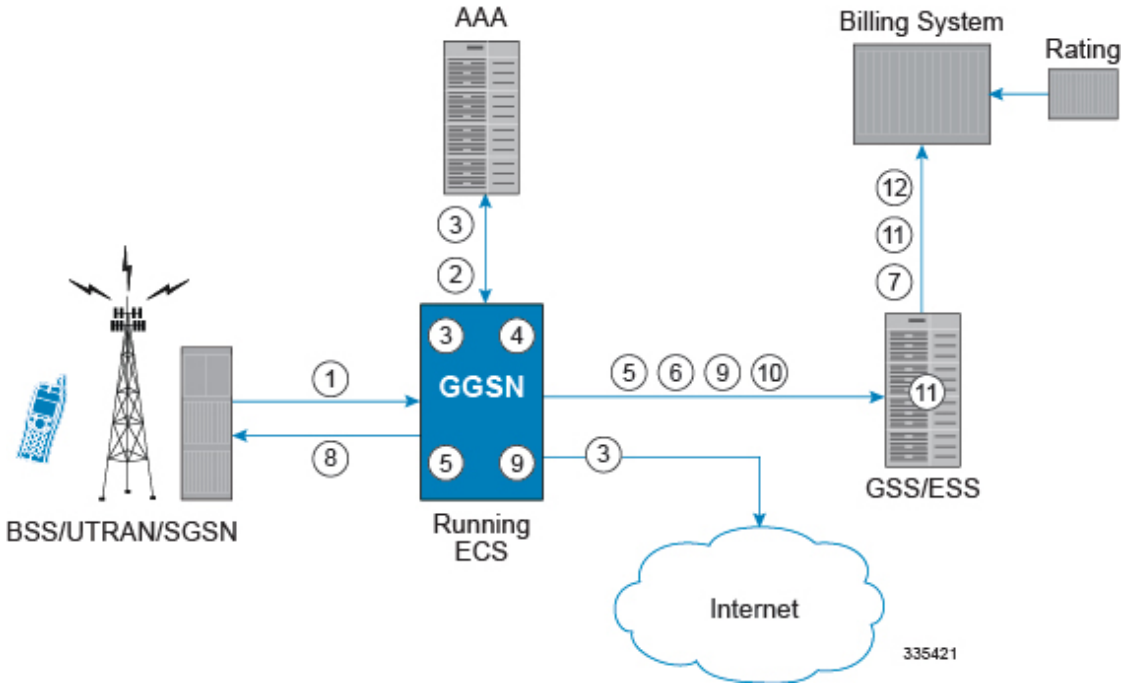


Table 3: Postpaid Billing with ECS in GPRS/UMTS Network

Step No.	Description
1	The subscriber initiates the session.
2	After subscriber authentication and authorization, the system starts the session.
3	Data packet flow and accounting starts.
4	System periodically generates xDRs and stores them to the system memory.
5	System generates G-CDRs/eG-CDRs and sends them to billing system as they are generated.
6	The billing system picks up the CDR files periodically.
7	Subscriber session ends after subscriber terminates the session.

Step No.	Description
8	The system stores the last of the xDRs to the system memory and final xDRs are SFTPd from system memory to external storage, if deployed or to billing system directly.
9	System sends the last of the G-CDRs/eG-CDRs to the billing system.
10	File Generation Utility, FileGen in external storage periodically runs to generate G-CDRs/eG-CDRs files for billing system and send them to the billing system.
11	The billing system picks up the xDR files from the external storage periodically.

ECS Postpaid Billing in CDMA-2000 Networks

The following figure and steps describe how ECS works within a CDMA-2000 network for postpaid billing.

Figure 11: Postpaid Billing with ECS in CDMA-2000 Network

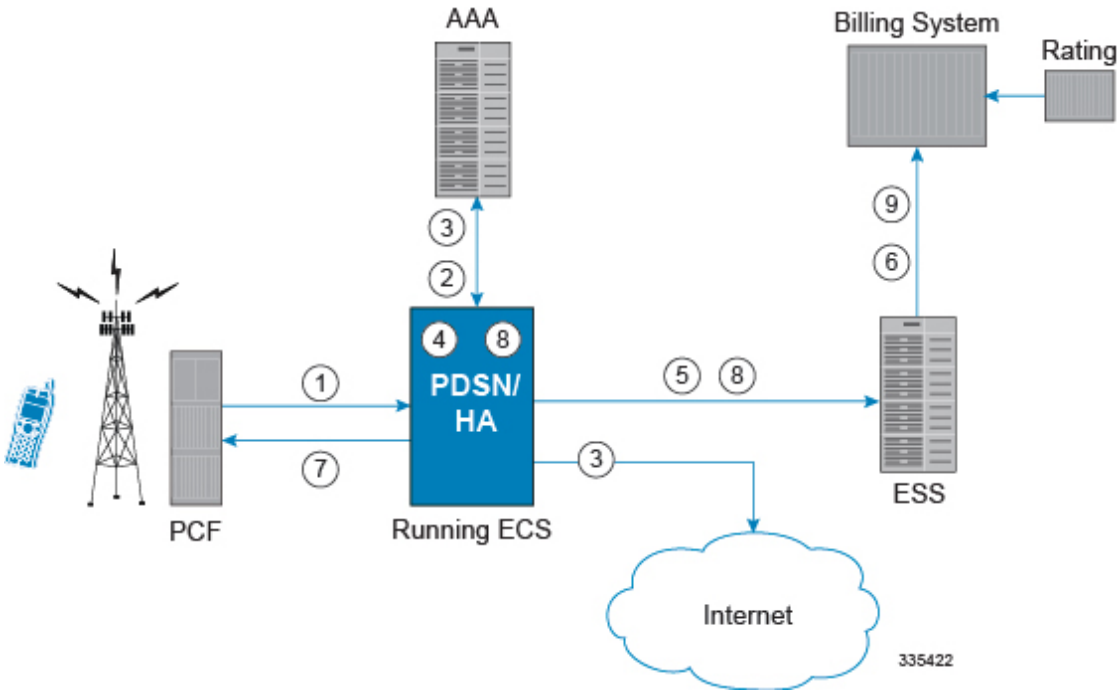


Table 4: Postpaid Billing with ECS in CDMA-2000 Network

Step No.	Description
1	The subscriber initiates the session.
2	After subscriber authentication and authorization, the system starts the session.

Step No.	Description
3	Data packet flow and accounting starts.
4	System periodically generates xDRs and stores them to the system memory.
5	EDRs/UDRs are periodically SFTPd from system memory to external storage, if deployed or to billing system directly as they are generated.
6	The billing system picks up the xDR files from the external storage periodically.
7	Subscriber session ends after subscriber terminates the session.
8	The system stores the last of the xDRs to the system memory and final xDRs are SFTPd from system memory to the external storage, if deployed or to billing system directly.
9	The external storage finally sends xDRs to the billing system.

External Storage

Important For information on availability/support for external storage, contact your Cisco account representative.

The external storage is a high availability, fault tolerant, redundant solution for short-term storage of files containing detail records (UDRs/EDRs/FDRs (xDRs)). To avoid loss of xDRs on the chassis due to overwriting, deletion, or unforeseen events such as power or network failure or unplanned chassis switchover, xDRs are off-loaded to external storage for storage and analysis to avoid loss of charging and network analysis information contained in the xDRs.

The xDR files can be pulled by the external storage from the chassis, or the chassis can push the xDR files to the external storage using SFTP protocol. In the Push mode, the external storage URL to which the CDR files need to be transferred to is specified. The configuration allows a primary and a secondary server to be configured. Configuring the secondary server is optional. Whenever a file transfer to the primary server fails for four consecutive times, the files will be transferred to the secondary server. The transfer will switch back to the original primary server when:

- Four consecutive transfer failures to the secondary server occur
- After switching from the primary server, 30 minutes elapses

In the push transfer mode, the following can be configured:

- Transfer interval—A time interval, in seconds, after which the CDRs are pushed to the configured IP periodically. All the files that are completed before the PUSH timer expires are pushed.

- Remove file after transfer—An option to keep or remove the CDR files on the hard disk after they are transferred to the external storage successfully.

The system running with ECS stores xDRs on an external storage, and the billing system collects the xDRs from the external storage and correlates them with the AAA accounting messages using 3GPP2-Correlation-IDs (for PDSN) or Charging IDs (for GGSN).

System Resource Allocation

ECS does not require manual resource allocation. The ECS subsystem automatically allocates the resources when ECS is enabled on the chassis. ECS must be enabled on the chassis before configuring services.

Redundancy Support in ECS

This section describes the redundancy support available in ECS to recover user sessions and charging records in the event of software/hardware failure.

Caution Persistent data flows are NOT recoverable during session recovery.

Important Redundancy is not available in the current version of the Cisco® XT2 platform.

Intra-chassis Session Recovery Interoperability

Intra-chassis session recovery is coupled with SessMgr recovery procedures.

Intra-chassis session recovery support is achieved by mirroring the SessMgr and AAAMgr processes. The SessMgrs are paired one-to-one with the AAAMgrs. The SessMgr sends checkpointed session information to the AAAMgr. ECS recovery is accomplished using this checkpointed information.

Important In order for session recovery to work there should be at least four packet processing cards, one standby and three active. Per active CPU with active SessMgrs, there is one standby SessMgr, and on the standby CPU, the same number of standby SessMgrs as the active SessMgrs in the active CPU.

There are two modes of session recovery, one from task failure and another on failure of CPU or packet processing card.

Recovery from Task Failure

When a SessMgr failure occurs, recovery is performed using the mirrored "standby-mode" SessMgr task running on the active packet processing card. The "standby-mode" task is renamed, made active, and is then populated using checkpointed session information from the AAAMgr task. A new "standby-mode" SessMgr is created.

Recovery from CPU or Packet Processing Card Failure

When a PSC, PSC2, or PPC hardware failure occurs, or when a planned packet processing card migration fails, the standby packet processing card is made active and the "standby-mode" SessMgr and AAAMgr tasks on the newly activated packet processing card perform session recovery.

Inter-chassis Session Recovery Interoperability

The system supports the simultaneous use of ECS and the Inter-chassis Session Recovery feature. When both features are enabled, ECS session information is regularly checkpointed from the active chassis to the standby as part of normal Service Redundancy Protocol processes. For more information on the Inter-chassis Session Recovery feature, refer to the *System Administration Guide*.

In the event of a manual switchover, there is no loss of accounting information. All xDR data from the active chassis is moved to a customer-configured external storage before switching over to the standby. This data can be retrieved at a later time. Upon completion of the switchover, the ECS sessions are maintained and the "now-active" chassis recreates all of the session state information including the generation of new xDRs.

In the event of an unplanned switchover, all accounting data that has not been written to the external storage is lost. Note that either the external storage can pull the xDR data from the chassis, or the chassis can push the xDR files to a configured external storage at user-configured intervals. For more information, see External Storage, on page 52. Upon completion of switchover, the ECS sessions are maintained and the "now-active" chassis recreates all of the session state information including the generation of new xDRs.

Regardless of the type of switchover that occurred, the names of the new xDR files will be different from those stored in the /records directory of packet processing card RAM on the "now-standby" chassis. Also, in addition to the file name, the content of many of the fields within the xDR files created by the "now-active" chassis will be different. ECS manages this impact with recovery mechanism. For more information on the differences and how to correlate the two files and other recovery information, see Impact on xDR File Naming, on page 54.

Inter-chassis Session Recovery Architecture

Inter-chassis redundancy in ECS uses Flow Detail Records (FDRs) and UDRs to manage the switchover between Active-Standby system. xDRs are moved between redundant external storage server and Active-Standby systems.

Session Recovery Improvements

In StarOS releases prior to 14.0, there were only 10 PCC rules that were recovered per bearer in the event of a session manager crash. In 14.0 and later releases, this limit has been increased to 24. That is, up to 24 PCC rules can be recovered post ICSR.

With the increase in the limit of PCC rules that can be recovered, the rules are not lost and hence the charging applied to the end users are not impacted.

Impact on xDR File Naming

The xDR file name is limited to 256 characters with the following syntax:

basename_ChargSvcName_ timestamp_SeqNumResetIndicator_FileSeqNumber

where:

- *basename*—A global configurable text string that is unique per system that uniquely identifies the global location of the system running ECS.
- *ChargSvcName*—A system context-based configurable text string that uniquely identifies a specific context-based charging service.
- *timestamp*—Date and time at the instance of file creation. Date and time in the form of "MMDDYYYYHHmmSS" where HH is a 24-hour value from 00-23.
- *SeqNumResetIndicator*—A one-byte counter used to discern the potential for duplicated FileSeqNumber with a range of 0 through 255, which is incremented by a value of 1 for the following conditions:
  - Failure of an ECS software process on an individual packet processing card
  - Failure of a system such that a second system takes over according to the Inter-chassis Session Recovery feature
  - File Sequence Number (FileSeqNumber) rollover from 999999999 to 0
- *FileSeqNumber*—Unique file sequence number for the file with nine-digit integer having range from 000000000 to 999999999. It is unique on each system.

With inter-chassis session recovery, only the first two fields in the xDR file names remain consistent between the active and standby chassis as these are parameters that are configured locally on the chassis. Per inter-chassis session recovery implementation requirements, the two chassis systems must be configured identically for all parameters not associated with physical connectivity to the distribution node.

The fields "timestamp", "SeqNumResetIndicator", and "FileSeqNumber" are all locally generated by the specific system through CDR subsystem, regardless of whether they are in an Inter-chassis Session Recovery arrangement or not.

- The "timestamp" value is unique to the system generating the actual xDRs and generated at the time the file is opened on the system.
- The SeqNumResetIndicator is a unique counter to determine the number of resets applied to FileSeqNumber. This counter is generated by CDR subsystem and increment the counter in event of resets in FileSeqNumber. This is required as "timestamp" field is not sufficient to distinguish between a unique and a duplicate xDR.

As such, the "SeqNumResetIndicator" field is used to distinguish between xDR files which have the same "FileSeqNumber" as a previously generated xDR as a result of:

- Normal operation, for example a rollover of the "FileSeqNumber" from maximum limit to 0.
- Due to a failure of one of the ECS processes running on a packet processing card card.
- Failure of the system (that is, Inter-chassis Session Recovery switchover).

In any scenario where the "FileSeqNumber" is reset to 0, the value of the "SeqNumResetIndicator" field is incremented by 1.

- The value of the "FileSeqNumber" is directly linked to the ECS process that is generating the specific xDRs. Any failure of this specific ECS process results in resetting of this field to 0.

Impact on xDR File Content

The following scenarios impact the xDR file content:

- On failure of an active chassis:

  On system startup, xDR files are generated in accordance with the standard processes and formats. If the system fails at any time it results in an inter-chassis session recovery switchover from active to standby and the following occurs depending on the state of the call/flow records and xDR file at the time of failure:

  - Call/flow records that were being generated and collected in system memory prior to being written out to /records directory on packet processing card RAM are not recoverable and therefore are lost.
  - Closed xDRs that have been written out to records directory on packet processing card RAM but that have yet to be retrieved by the external storage are recoverable.
  - Closed xDRs that have been retrieved and processed by the external storage have no impact.

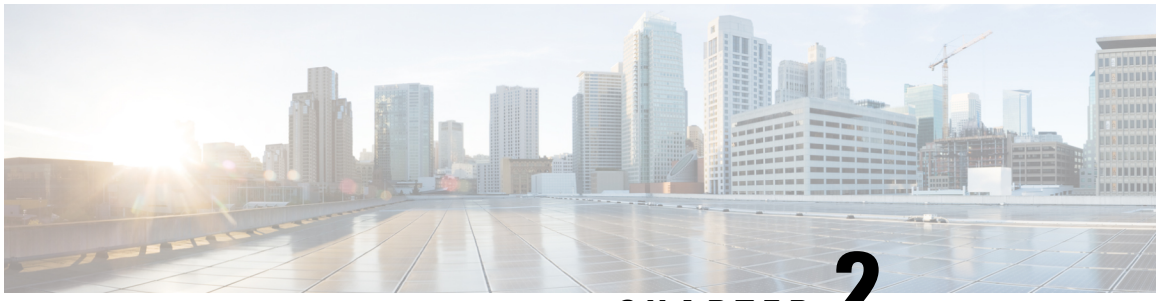
- On the activation of a Standby chassis:

  Upon detection of a failure of the original active chassis, the standby chassis transits to the active state and begins serving the subscriber sessions that were being served by the now failed chassis. Any subsequent new subscriber session will be processed by this active chassis and will generate xDRs per the standard processes and procedures.

  However, this transition impacts the xDRs for those subscribers that are in-progress at the time of the transition. For in progress subscribers, a subset of the xDR fields and their contents are carried over to the newly active chassis via the SRP link. These fields and their contents, which are carried over after an Inter-chassis Session Recovery switchover, are as follows:

  - HA-CORRELATION-ID
  - PDSN-CORRELATION-ID (PDSN only)
  - PDSN-NAS-IP-ADDRESS (PDSN only)
  - PDSN-NAS-ID (PDSN only)
  - USERNAME
  - MSID
  - RADIUS-NAS-IP-ADDRESS

  All remaining fields are populated in accordance with the procedures associated with any new flow with the exceptions that, the field "First Packet Direction" is set to "Unknown" for all in-progress flows that were interrupted by the switchover and the field "FDR Reason" is marked as a PDSN Handoff and therefore is set to a value of "1" and corresponding actions are taken by the billing system to assure a proper and correct accounting of subscriber activities.



CHAPTER 2

Enhanced Charging Service Configuration

This chapter describes how to configure the Enhanced Charging Service (ECS) functionality, also known as Active Charging Service (ACS).

The following topics are covered in this chapter:

Initial Configuration

Initial configuration includes the following:

Step 1 Install the ECS license as described in Installing the ECS License, on page 58.

Step 2 Create the ECS administrative user account as described in Creating the ECS Administrative User Account, on page 57.

Step 3 Enable ECS as described in Enabling Enhanced Charging Service, on page 58.

Step 4 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*.

Important 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.

Creating the ECS Administrative User Account

At least one administrative user account with ECS privileges must be configured on the system. This is the account that is used to log on and execute ECS-related commands. For security purposes, it is recommended that these user accounts be created along with general system functionality administration.

To create the ECS administrative user account, use the following configuration:

```
configure
  context local
    administrator <user_name> password <password> ecs
    end
```

Notes:

- Aside from having ECS capabilities, an ECS Administrator account also has the same capabilities and privileges as any other system-level administrator account.
- You can also create system ECS user account for a config-administrator, operator, or inspector. ECS accounts have the same system-level privileges of normal system accounts except that they have full ECS command execution capability. For example, an ECS account has rights to execute every command that a regular administrator can in addition to all of the ECS commands.
- Note that only Administrator and Config-administrator level users can provision ECS functionality. Refer to the *Configuring System Settings* chapter of the *System Administration and Configuration Guide* for additional information on administrative user privileges.

Installing the ECS License

The ECS in-line service is a licensed Cisco product. Separate session and feature licenses may be required. Contact your Cisco account representative for detailed information on licensing requirements.

For information on installing and verifying licenses, refer to the *Managing License Keys* section of the *Software Management Operations* chapter in the *System Administration Guide*.

Enabling Enhanced Charging Service

Enhanced charging must be enabled before configuring charging services.

To enable Enhanced Charging Service, use the following configuration:

```
configure
  require active-charging
    context local
      interface <interface_name>
        ip address <ip_address/mask>
        exit
      server ftpd
      end
```

Notes:

- The **require active-charging** command must be configured before any services are configured, so that the resource subsystem can appropriately reserve adequate memory for ECS-related tasks. After configuring this command, the configuration must be saved and the system rebooted in order to allocate the resources for ECS on system startup.

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.

Configuring the Enhanced Charging Service

A charging service has analyzers that define which packets to examine and ruledefs (ruledefs) that define what packet contents to take action on and what action to take when the ruledef is matched. Charging services are configured at the global configuration level and are available to perform packet inspection on sessions in all contexts.

To configure the Enhanced Charging Service:

Step 1 Create the ECS service as described in Creating the Enhanced Charging Service, on page 59.

Step 2 Configure a ruledef as described in Configuring Rule Definitions, on page 59.

Step 3 Create a charging action as described in Configuring Charging Actions, on page 61.

Step 4 Define a rulebase as described in Configuring Rulebase, on page 62.

Step 5 *Optional*. Define a rulebase list in the ACS configuration mode and configure the rulebase list in an APN, as described in Configuring Rulebase Lists, on page 63.

Step 6 *Optional*. Enable dynamic collection of ruledef statistics as described in Configuring Ruledef Statistics Collection, on page 63.

Step 7 Set EDR formats as described in Setting EDR Formats.

Step 8 Set UDR formats as described in Setting UDR Formats.

Step 9 Enable charging record retrieval as described in Enabling Charging Record Retrieval, on page 65.

Step 10 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*.

Important 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.

Creating the Enhanced Charging Service

To create an Enhanced Charging Service, use the following configuration:

```
configure
  active-charging service <ecs_service_name>
  end
```

Notes:

- In this release, only one ECS service can be created in a system.

Configuring Rule Definitions

To create and configure a ruledef use the following configuration:

```
configure
   active-charging service <ecs_service_name>
      ruledef <ruledef_name>
         <protocol> <expression> <operator> <condition>
         rule-application { charging | post-processing | routing }
         end
```

Notes:

- If the same ruledef is to be used for charging in one rulebase and for post-processing in another, then two separate identical ruledefs should be defined.
- For information on all the protocol types, expressions, operators, and conditions supported, refer to the *ACS Ruledef Configuration Mode Commands* chapter of the *Command Line Interface Reference*.
- The **rule-application** command specifies the ruledef type. By default, if not specified, the system considers a ruledef as a charging ruledef.
- In 14.1 and earlier releases, a maximum of 10 rule expressions (rule-lines) can be added in one ruledef.

  In 15.0 and later releases, a maximum of 32 rule expressions (rule-lines) can be added in one ruledef.

Verifying your Configuration

To verify your configuration, in the Exec Mode, enter the following command:

```
show active-charging
ruledef { all | charging | name ruledef_name | post-processing | routing }
```

Configuring Group of Ruledefs

A group-of-ruledefs enables grouping rules into categories, so that charging systems can base the charging policy on the category.

To create and configure a group-of-ruledefs, use the following configuration:

```
configure
   active-charging service <ecs_service_name>
      group-of-ruledefs <ruledef_group_name>
         add-ruledef priority <priority> ruledef <ruledef_name>
         group-of-ruledefs-application { charging | content-filtering |
gx-alias | post-processing }
         end
```

Notes:

- In releases prior to 20.1: A maximum of 128 ruledefs can be added to a group-of-ruledef.

  In 20.1 and later releases: A maximum of 512 ruledefs can be added to a group-of-ruledef.
- In 14.1 and earlier releases, a maximum of 64 group-of-ruledefs can be configured.

  In 15.0 and later releases, a maximum of 128 group-of-ruledefs can be configured.

  In 20.1 and later releases, a maximum of 384 group-of-ruledefs can be configured.

Important The total number of ruledefs supported for all GoRs must be used with caution due to the high memory impact. Any modifications to the ruledef or GoR configurations beyond the WARN state of the SCT Task memory may have adverse impact on the system.

- The **group-of-ruledefs-application** command specifies the group-of-ruledef type. By default, if not specified, the system considers a group-of-ruledef as a charging group-of-ruledef.

Verifying your Configuration

To verify your configuration, in the Exec Mode, enter the following command:

```
show active-charging group-of-ruledefs name <ruledef_group_name>
```

Configuring Charging Actions

Charging actions are used with rulebases and must be created before a rulebase is configured.

To create a charging action, use the following configuration:

```
configure
  active-charging service <ecs_service_name>
    charging-action <charging_action_name>
      content-id <content_id>
      retransmissions-counted
      billing-action { create-edrs { charging-edr <charging_edr_format_name>
 | reporting-edr <reporting_edr_format_name> } + [ wait-until-flow-ends ] |
egcdr | exclude-from-udrs | radius | rf } +
      end
```

Notes:

- Up to eight packet filters can be specified in a charging action.

Verifying your Configuration

To verify your configuration, in the Exec Mode, enter the following command:

```
show active-charging charging-action name <charging_action_name>
```

Configuring IP Readdressing

Readdressing of packets based on the destination IP address of the packets enables redirecting unknown gateway traffic to known/trusted gateways. This is implemented by configuring the re-address server in the charging action.

To configure the IP Readdressing feature, use the following configuration:

```
configure
  active-charging service <ecs_service_name>
    charging-action <charging_action_name>
      flow action readdress server  ipv4_address/ipv6_address [
```

```
discard-on-failure ]  [ dns-proxy-bypass ]  [ port port_number [
discard-on-failure ] [ dns-proxy-bypass ] ]
       end
```

To configure the IP Readdressing feature when the readdress server-list is defined under charging-action, use the following configuration:

```
configure
  active-charging service service_name
    charging-action  charging_action_name
      flow action readdress server-list  server_list_name [ hierarchy ] [
round-robin ] [ dns-proxy-bypass ] [ discard-on-failure ]
       end
```

Configuring Next Hop Address

To configure the Next Hop Address configuration feature, use the following configuration:

```
configure
  active-charging service <ecs_service_name>
    charging-action <charging_action_name>
      nexthop-forwarding-address <ip_address>
      end
```

Configuring Rulebase

A rulebase specifies which protocol analyzers to run and which packets are analyzed. Multiple rulebases may be defined for the Enhanced Charging Service. A rulebase is basically a subscriber's profile in a charging service.

To create and configure a rulebase, use the following configuration:

```
configure
  active-charging service <ecs_service_name>
    rulebase <rulebase_name>
      flow end-condition { content-filtering | hagr | handoff |
normal-end-signaling | session-end | tethering-signature-change |
url-blacklisting | timeout } [ flow-overflow ] + { charging-edr
<charging_edr_format_name> | reporting-edr <reporting_edr_format_name> }
      billing-records udr udr-format <udr_format_name>
     action priority <action_priority> { [ dynamic-only | static-and-dynamic
 | timedef <timedef_name> ] { group-of-ruledefs <ruledef_group_name> | ruledef
<ruledef_name> } charging-action <charging_action_name> [ monitoring-key
<monitoring_key> ] [ description <description> ] }
      route priority <route_priority> ruledef <ruledef_name> analyzer <analyzer>
 [ description <description> ]
      rtp dynamic-flow-detection
      udr threshold interval <interval>
      cca radius charging context <context> group <group_name>
      cca radius accounting interval <interval>
      end
```

Notes:

- When R7 Gx is enabled, "static-and-dynamic" rules behave exactly like "dynamic-only" rules. That is, they must be activated explicitly by the PCRF. When Gx is not enabled, "static-and-dynamic" rules behave exactly like static rules.
- In release 20.2, the **tethering-signature-change** keyword is added to create an EDR with the specified EDR format whenever a flow ends due to tethering signature change.

Verifying your Configuration

To verify your configuration, in the Exec Mode, enter the following command:

```
show active-charging rulebase name <rulebase_name>
```

Configuring Rulebase Lists

To create a rulebase list, use the following configuration:

```
configure
  active-charging service <ecs_service_name>
    rulebase-list <rulebase_list_name> <rulebase_name>[ <rulebase_name> + ]
    exit
```

Configuring a Rulebase List in an APN

To configure the rulebase list that was created in the ACS configuration mode in an APN, use the following configuration:

```
configure
  context <context_name>
    apn <apn_name>
      active-charging rulebase-list <rulebase_list_name>
      exit
```

Verifying your configuration

To verify your configuration for the rulebase list and APN, in the Exec mode, enter the following command:

```
show configuration
```

To verify your APN configuration, in the Exec mode, enter the following command:

```
show configuration apn <apn_name>
```

Configuring Ruledef Statistics Collection

To dynamically enable ruledef statistics collection in the ACS Configuration mode, use the following configuration:

```
configure
  active-charging service <ecs_service_name>
    statistics-collection { all | ruledef { all | charging | firewall |
 post-processing } }
    [ no ] statistics-collection
    end
```

Notes:

- By default, no statistics will be maintained.
- The **[no] statistics-collection all** and **[no] statistics-collection ruledef all** commands will result in the same output.
- If the command is not configured, statistics collection will not be enabled and the following error message will be displayed in the **show active-charging sessions full** CLI — "statistics collection disabled; not collecting <*charging/firewall/postprocessing*> ruledef stats".

To dynamically enable ruledef statistics collection in the Exec mode, use the following configuration:

```
statistics-collection active-charging { all | charging | firewall |
post-processing { callid call_id | imsi imsi_number } }
[ no ] statistics-collection active-charging { callid call_id | imsi
imsi_number }
```

Notes:

- The ruledef statistics will be maintained for a bearer only if this command is configured. By default, the statistics will not be maintained.
- If the command is not configured, statistics collection will not be enabled and the following error message will be displayed in the **show active-charging sessions full** CLI — "statistics collection disabled; not collecting <*charging/firewall/postprocessing*> ruledef stats".

To view subscriber statistics, in the Exec Mode, use the following command:

```
show active-charging subscribers { acsmgr instance instance_id | all | callid
 call_id | full | imsi imsi_number | rulebase rulebase_name }
```

Setting EDR Formats

ECS generates postpaid charging data files which can be retrieved from the system periodically and used as input to a billing mediation system for postprocessing.

EDRs are generated according to action statements in rule commands.

Up to 32 different EDR schema types may be specified, each composed of up to 32 fields or analyzer parameter names. The records are written at the time of each rule event in a comma-separated (CSV) format.

Important If you have configured RADIUS Prepaid Billing, configuring charging records is optional.

To set the EDR formats use the following configuration:

```
configure
  active-charging service <ecs_service_name>
    edr-format <edr_format_name>
      attribute <attribute> { [ format { MM/DD/YY-HH:MM:SS |
MM/DD/YYYY-HH:MM:SS | YYYY/MM/DD-HH:MM:SS | YYYYMMDDHHMMSS | seconds } ]
 [ localtime ] | [ { ip | tcp } { bytes | pkts } { downlink | uplink } ]
 priority <priority> }
      rule-variable <protocol> <rule> priority <priority>
      event-label <event_label> priority <priority>
      delimiter { comma | tab }
      end
```

For information on EDR format configuration and rule variables, refer to the *EDR Format Configuration Mode Commands* chapter of the *Command Line Interface Reference*.

Verifying your Configuration

To verify your configuration, in the Exec Mode, enter the following command:

```
show active-charging edr-format name <edr_format_name>
```

Setting UDR Formats

ECS generates postpaid charging data files which can be retrieved from the system periodically and used as input to a billing mediation system for postprocessing.

UDRs are generated according to action statements in rule commands. Up to 32 different UDR schema types may be specified, each composed of up to 32 fields or analyzer parameter names. The records are written thresholds in a comma-separated (CSV) format.

Important If you have configured RADIUS Prepaid Billing, configuring charging records is optional.

To set the UDR format, use the following configuration:

```
configure
  active-charging service <ecs_service_name>
    udr-format <udr_format_name>
      attribute <attribute> { [ format { MM/DD/YY-HH:MM:SS |
MM/DD/YYYY-HH:MM:SS | YYYY/MM/DD-HH:MM:SS | YYYYMMDDHHMMSS | seconds } ]
 [ localtime ] | [ { bytes | pkts } { downlink | uplink } ] ] priority
<priority> }
      end
```

Important For information on UDR format configuration and rule variables, refer to the *UDR Format Configuration Mode Commands* chapter of the *Command Line Interface Reference*.

Verifying your Configuration

To verify your configuration, in the Exec Mode, enter the following command:

```
show active-charging udr-format name <udr_format_name>
```

Enabling Charging Record Retrieval

To retrieve charging records you must configure the context that stores the charging records to accept SFTP connections.

To enable SFTP, use the following configuration:

```
configure
  context local
```

```
    administrator <user_name> [ encrypted ] password <password>
    config-administrator <user_name> [ encrypted ] password <password>
    exit
  context <context_name>
    ssh generate key
    server sshd
    subsystem sftp
    end
```

Notes:

- You must specify the **sftp** keyword to enable the new account to SFTP into the context to retrieve record files.

Optional Configurations

This section describes the following optional configuration procedures:

- Configuring a Rulebase for a Subscriber, on page 66
- Configuring a Rulebase within an APN, on page 66
- Configuring Charging Rule Optimization, on page 67

Important 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.

Configuring a Rulebase for a Subscriber

This section describes how to apply an existing rulebase to a subscriber. For information on how to configure rulebases, see Configuring Rulebase, on page 62.

To configure a rulebase for a subscriber, use the following configuration:

```
configure
  context <context_name>
    subscriber name <subscriber_name>
      active-charging rulebase <rulebase_name>
      end
```

Configuring a Rulebase within an APN

This section describes how to configure an existing rulebase within an APN for a GGSN. For information on how to configure rulebases, see Configuring Rulebase, on page 62.

Important This information is only applicable to GGSN networks.

To configure a rulebase in an APN, use the following configuration:

```
configure
  context <context_name>
    apn <apn_name>
      active-charging rulebase <rulebase_name>
      end
```

Configuring Charging Rule Optimization

This section describes how to configure the internal optimization level for improved performance when the system evaluates each instance of the **action** CLI command.

To configure the rule optimization level, use the following configuration:

```
configure
  active-charging service <ecs_service_name>
    rulebase <rulebase_name>
      charging-rule-optimization { high | low | medium }
      end
```

Notes:

- In StarOS 14.0 and later releases, the **charging-rule-optimization** command is deprecated. Rule optimization is always enabled with the optimization level set to high as standard behavior.
- In 11.0 and later releases, the **medium** option is deprecated.
- Both the **high** and **medium** options cause reorganization of the entire memory structure whenever any change is made (for example, addition of another **action** CLI command).
- The **high** option causes allocation of a significant amount of memory for the most efficient organization.

Configuring Service-scheme Framework

The Service Scheme configuration is required to configure and enable features such as Response-based Charging, Response-based TRM, and Location QoS Override features for a subscriber. The configuration commands described in this section can be used to configure the service-scheme framework.

The following topics are covered in this section:

Configuring Subscriber Base

To configure the Active Charging Service (ACS) subscriber base, use the following configuration:

```
configure
  active-charging service service_name
    subscriber-base subs_base_name
      priority priority subs-class subs_class_name bind service-scheme
serv_scheme_name
      end
```

Notes:

- The **priority** command is used to assign priority to the service-scheme association within a subscriber base. This priority must be unique within a subscriber base.

Configuring Subscriber Class

To configure the Active Charging Service (ACS) subscriber class, use the following configuration:

```
configure
   active-charging service service_name
      subs-class subs_class_name
         [ no ] any-match operator condition
         [ no ] apn operator apn_name
         [ no ] multi-line-or all-lines
         [ no ] rulebase operator rulebase_name
         [ no ] v-apn operator v_apn_name
         end
```

Notes:

- The **any-match** command is used to enable or disable the wildcard configuration.
- The **apn** command is used to specify the APN name as a condition.
- The **multi-line-or** command is used to check if the OR operator must be applied to all lines in a subscriber class.
- The **rulebase** command is used to specify the rulebase name as a condition.
- The **v-apn** command is used to specify the virtual APN name as a condition.

Configuring Service Scheme

To enable the association of service-scheme based on subscriber class, use the following configuration:

```
configure
   active-charging service service_name
      service-scheme service_scheme_name
         [ no ] trigger { loc-update | sess-setup }
         end
```

Notes:

- The **trigger** command is used to configure the set of trigger events such as session-setup, location-update that will be handled under the service-scheme.

Configuring Service Scheme Trigger

To configure the set of triggers that will be handled under the associated service-scheme, use the following configuration:

```
configure
   active-charging service service_name
      service-scheme service_scheme_name
         [ no ] trigger { loc-update | sess-setup }
           priority priority trigger-condition trigger_condn_name trigger-action
 trigger_action_name
          end
```

Notes:

- The **priority** command is used to assign priority to the trigger events configured in service-scheme. The priority must be unique within a trigger.

Configuring Trigger Action

To configure the Active Charging Service (ACS) trigger actions, use the following configuration:

```
configure
  active-charging service service_name
    trigger-action trigger_action_name
      [ no ] charge-request-to-response http { all | connect | delete |
 get | head | options | post | put | trace }
      [ no ] throttle-suppress
      [ no ] transactional-rule-matching response http { all | connect
| delete | get | head | options | post | put | trace }
      end
```

Notes:

- The **charge-request-to-response** command is added in support of the Response-based Charging feature to delay charging till the HTTP response for the configured HTTP request method(s).
- The **throttle-suppress** command is added in support of the Location based QoS Override feature to perform throttle suppression to provide unlimited bandwidth based on trigger condition matched.
- The **transactional-rule-matching** command is added in support of the Response-based TRM feature to delayw-c engagement of TRM till the HTTP response for the configured HTTP request method(s).

Configuring Trigger Condition

To configure Active Charging Service (ACS) trigger conditions, use the following configuration:

```
configure
  active-charging service service_name
    trigger-condition trigger_condn_name
      [ no ] any-match operator condition
      [ no ] local-policy-rule = local_policy_rule
      [ no ] multi-line-or all-lines
      end
```

Notes:

- The **any-match** command is used to analyze all flows created after event activation.
- The **local-policy-rule** command is used to specify the local-policy rule within ECS for enabling trigger condition.
- The **multi-line-or** command is used to check if the OR operator must be applied to all lines in a trigger-condition.

Verifying Service Scheme Configuration

Use the following commands in the Exec Mode to verify your configuration:

- This command is used to display the number of subscribers associated with the configured service-scheme:

  show active-charging service-scheme { all | name *serv_scheme_name* **| statistics [name** *serv_scheme_name* **] } [service name** *service_name* **] [| { grep** *grep_options* **| more }]**

- This command displays the service-scheme selected for the particular subscriber:

  show active-charging subscribers full all

- This command displays information about the trigger action(s) configured in a service.

  show active-charging trigger-action { all | name *trigger_action_name* **[service** *acs_service_name* **] } [| { grep** *grep_options* **| more }]**

- This command displays information about the trigger condition(s) configured in a service.

  show active-charging trigger-condition { { all | name *trigger_condn_name* **[service** *acs_service_name* **] } | statistics [name** *trigger_condn_name* **] } [| { grep** *grep_options* **| more }]**

Sample Configuration

Use the sample configuration to enable features based on service-scheme framework for a subscriber.

```
configure
  active-charging service s1
    subscriber-base default
      priority 10 subs-class class1 bind service-scheme scheme1
      priority 20 subs-class class2 bind service-scheme scheme2
    #exit
    subs-class class1
      apn = cisco.com
      rulebase = plan1
      v-apn != some_virtual_apn
      multi-line-or
    #exit
    subs-class class2
      any-match
    #exit
    service-scheme scheme1
      trigger sess-setup
        priority 10 trigger-condition tc1 trigger-action ta1
        priority 20 trigger-condition tc2 trigger-action ta2
      #exit
    #exit
    service-scheme scheme2
      trigger sess-setup
        priority 10 trigger-condition tc1 trigger-action ta1
      #exit
```

```
#exit
trigger-condition tc1
  any-match
#exit
trigger-action ta1
  transactional-rule-matching response http all
  charge-request-to-response http all
#exit
trigger-action ta2
  throttle-suppress
#exit
```

Configuring Enhanced Features

The configuration examples in this section are optional and provided to cover the most common uses of ECS in a live network.

The following topics are covered in this section:

Configuring Prepaid Credit Control Application (CCA)

This section describes how to configure the Prepaid Credit Control Application for Diameter or RADIUS.

Important To configure and enable Diameter and DCCA functionality with ECS, you must obtain and install the relevant license on the chassis. Contact your Cisco account representative for detailed information on licensing requirements.

Important Before configuring Diameter or RADIUS CCA, you must configure AAA parameters. For more information, 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*.

To configure Prepaid Credit Control Application:

Step 1 Configure the Prepaid Credit Control Application for Diameter or RADIUS as described in Configuring Prepaid CCA for Diameter or RADIUS, on page 72.

Step 2 Configure the required Prepaid Credit Control Mode:

- Configuring Diameter Prepaid Credit Control Application (DCCA), on page 74
- Configuring RADIUS Prepaid Credit Control Application, on page 76

Important 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.

Configuring Prepaid CCA for Diameter or RADIUS

To configure the Prepaid Credit Control Application for Diameter or RADIUS, use the following configuration:

```
configure
  active-charging service <ecs_service_name>
    charging-action <charging_action_name>
     cca charging credit [ preemptively-request | rating-group <coupon_id>
 ]
      exit
    credit-control [ group <group_name> ]
      mode { diameter | radius }
      quota time-threshold { <absolute_value> | percent <percent_value> }
      quota unit-threshold { <absolute_value> | percent <percent_value> }
      quota volume-threshold { <absolute_value> | percent <percent_value> }
      end
```

Notes:

- <*ecs_service_name*> must be the name of the Enhanced Charging Service in which you want to configure Prepaid Credit Control Application.
- <*charging_action_name*> must be the name of the charging action for which you want to configure Prepaid Credit Control Application.
- *Optional:* To configure the redirection of URL for packets that match a ruledef and action on quota request timer, in the Charging Action Configuration Mode, enter the following command. This command also specifies the redirect-URL action on packet and flow for Session Control functionality.

  In 12.2 and later releases: **flow action redirect-url** <*redirect_url*> **[[encryption { blowfish128 | blowfish64 } | { { aes128 | aes256 } [salt] } } [encrypted] key** <*key*> **] [clear-quota-retry-timer] [first-request-only [post-redirect { allow | discard | terminate }]]**

  In 12.1 and earlier releases: **flow action redirect-url** <*redirect_url*> **[clear-quota-retry-timer]**

  The following example shows the redirection of a URL for packets that match a ruledef:

```
charging-action http-redirect
   content-id 3020
   retransmissions-counted
   billing-action exclude-from-udrs
   flow action redirect-url
"http://10.1.67.214/cgi-bin/aoc.cgi077imsi=#bearer.calling-station-id#

&url=http.url#&acctsessid=#bearer.acct-session-id#&correlationid=#bearer.correlation-id#

   &username=#bearer.user-name#&ip=#bearer.served-bsa-addr#&subid=#bearer.subscriber-id#

   &host=#http.host#&httpuri=#http.uri#" clear-quota-retry-timer
   end
```

- *Optional:* To configure credit control quota related parameters, use the following configuration:

```
configure
   active-charging service <ecs_service_name>
      rulebase <rulebase_name>
         cca quota { holding-time <holding_time> content-id <content_id> |
retry-time <retry_time> [ max-retries <max_retries> ] }
        cca quota time-duration algorithm { consumed-time <consumed_time>
 [ plus-idle ] [ content-id <content_id> ] | continuous-time-periods
<seconds> [ content-id <content_id> ] | parking-meter <seconds> [ content-id
 <content_id> ] }
        end
```

<*rulebase_name*> must be the name of the rulebase in which you want to configure Prepaid Credit Control configurables.

- *Optional:* To define credit control rules for quota state and URL redirect match rules with RADIUS AVP, use the following configuration:

```
configure
   active-charging service <ecs_service_name>
      ruledef <ruledef_name>
         cca quota-state <operator> { limit-reached | lower-bandwidth }
         cca redirect-indicator <operator> <indicator_value>
         end
```

<*ruledef_name*> must be the name of the ruledef that you want to use for Prepaid Credit Control Application rules.

cca redirect-indicator configuration is a RADIUS-specific configuration.

- *Optional:* This is a Diameter-specific configuration. To configure the failure handling options for credit control session, in the Credit Control Configuration Mode, use the following configuration:

```
configure
   active-charging service <ecs_service_name>
      credit-control [ group <group_name> ]
         failure-handling { ccfh-session-timeout <session_timeout> | {
initial-request | terminate-request | update-request } { continue [
go-offline-after-tx-expiry | retry-after-tx-expiry ] |
retry-and-terminate [ retry-after-tx-expiry ] | terminate }
        end
```

- *Optional:* To configure the triggering option for credit re-authorization when the named values in the subscriber session changes, use the following configuration:

```
configure
  active-charging service <ecs_service_name>
    credit-control [ group <group_name> ]
      trigger type { cellid | lac | qos | rat | sgsn } +
      end
```

- *Optional:* This is a Diameter-specific configuration. If the configuration is for 3GPP network, to configure the virtual or real APN name to be sent in Credit Control Application (CCA) message, use the following configuration:

```
configure
  active-charging service <ecs_service_name>
    credit-control [ group <group_name> ]
      apn-name-to-be-included { gn | virtual }
      end
```

Configuring Diameter Prepaid Credit Control Application (DCCA)

This section describes how to configure the Diameter Prepaid Credit Control Application.

Important To configure and enable Diameter and DCCA functionality with ECS, you must obtain and install the relevant license on the chassis. Contact your Cisco account representative for detailed information on licensing requirements.

Important It is assumed that you have already fully configured the AAA parameters, and Credit Control Application as described in Configuring Prepaid Credit Control Application (CCA), on page 71 for Diameter mode. For information on configuring AAA parameters, 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*.

To configure Diameter Prepaid Credit Control Application, use the following configuration.

```
configure
  active-charging service <ecs_service_name>
    credit-control [ group <cc_group_name> ]
      mode diameter
      diameter origin endpoint <endpoint_name>
      diameter dictionary <dcca_dictionary>
      diameter peer-select peer peer_name [ realm realm_name ] [
secondary-peer secondary_peer_name [ realm realm_name ] ] [ imsi-based { { prefix
 | suffix } imsi/prefix/suffix_start_value } [ to imsi/prefix/suffix_end_value ] ] [
 msisdn-based { { prefix | suffix } msisdn-based/prefix/suffix_start_value } [
to msisdn-based/prefix/suffix_end_value ] ]
      end
```

Notes:

- Diameter peer configuration set with the **diameter peer-select** command can be overridden by the **dcca peer-select peer** command in the APN Configuration mode for 3GPP service networks, and in Subscriber Configuration mode in other service networks.
- The specific Credit Control Group to be used for subscribers must be configured in the APN Configuration Mode using the **credit-control-group** <*cc_group_name*> command.
- *Optional:* To configure the maximum time, in seconds, to wait for a response from Diameter peer, in the Credit Control Configuration Mode, enter the following command:

  diameter pending-timeout <*duration*>
- *Optional:* To configure Diameter Credit Control Session Failover, in the Credit Control Configuration Mode, enter the following command:

  diameter session failover

  When enabled, in the event of failure, failure handling action is based on the **failure-handling** CLI.
- *Optional:* If you want to configure the service for IMS authorization in 3GPP service network, you can configure dynamic rule matching with Gx interface and dynamic rule matching order in rulebase, use the following configuration:

```
configure
   active-charging service <ecs_service_name>
      rulebase <rulebase_name>
         dynamic-rule order { always-first | first-if-tied }
         action priority <action_priority> { [ dynamic-only |
static-and-dynamic | timedef <timedef_name> ] { group-of-ruledefs
<ruledef_group_name> | ruledef <ruledef_name> } charging-action
<charging_action_name> [ monitoring-key <monitoring_key> ] [ description
<description> ] }
         end
```

- *Optional:* To configure Diameter group AVP Requested-Service-Unit for Gy interface support to include a sub-AVP in CCRs using volume, time, and unit specific charging, in the Rulebase Configuration Mode, enter the following command:

  cca diameter requested-service-unit sub-avp { time cc-time <*duration*> **| units cc-service-specific-units** <*charging_unit*> **| volume { cc-input-octets** <*bytes*> **| cc-output-octets** <*bytes*> **| cc-total-octets** <*bytes*> **} + }**
- Ensure the Diameter endpoint parameters are configured. For information on configuring Diameter endpoint, 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*.

Configuring Peer-Select in Subscriber Configuration Mode (Optional)

This section describes how to configure Diameter peer-select within a subscriber configuration.

Important The **dcca peer-select** configuration completely overrides all instances of **diameter peer-select** configured within the Credit Control Configuration Mode for an Enhanced Charging Service.

To configure DCCA peers within a subscriber configuration, use the following configuration:

```
configure
  context <context_name>
    subscriber name <subscriber_name>
      dcca peer-select peer <host_name> [ [ realm <realm_name> ] [
secondary-peer <host_name> [ realm <realm_name> ] ] ]
      end
```

Configuring Peer-Select in APN Configuration Mode (Optional)

This section describes how to configure Diameter peer-select within an APN configuration.

Important This information is only applicable to GGSN networks.

Important The **dcca peer-select** configuration completely overrides all instances of **diameter peer-select** configured within the Credit Control Configuration Mode for an Enhanced Charging Service.

To configure DCCA peers within an APN, use the following configuration:

```
configure
  context <context_name>
    apn <apn_name>
      dcca peer-select peer <host_name> [ [ realm <realm_name> ] [
secondary-peer <host_name> [ realm <realm_name> ] ] ]
      end
```

Configuring RADIUS Prepaid Credit Control Application

RADIUS prepaid billing operates on a per content-type basis. Individual content-types are marked for prepaid treatment. When a traffic analysis rule marked with prepaid content-types matches, it triggers prepaid charge management.

Important The RADIUS Prepaid feature of ECS has no connection to the system-level Prepaid Billing Support or the 3GPP2 Prepaid features that are enabled under different licenses.

Important It is assumed that you have already fully configured the AAA parameters, and Credit Control Application as described in Configuring Prepaid Credit Control Application (CCA), on page 71 for RADIUS mode. For information on configuring AAA parameters, 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*.

To configure RADIUS Prepaid Charging with Enhanced Charging, use the following configuration.

```
configure
  active-charging service <ecs_service_name>
```

```
credit-control [ group <group_name> ]
  mode radius
  exit
rulebase <rulebase_name>
  cca radius charging context <vpn_context> [ group <group_name> ]
  end
```

Notes:

- <*rulebase_name*> must be the name of the rulebase in which you want to configure Prepaid Credit Control configurables.
- <*vpn_context*> must be the charging context in which the RADIUS parameters are configured:
- *Optional:* To specify the accounting interval duration for RADIUS prepaid accounting, in the ACS Rulebase Configuration Mode, enter the following command:

  cca radius accounting interval <*interval*>
- *Optional:* To specify the user password for RADIUS prepaid services, in the ACS Rulebase Configuration Mode, enter the following command:

  cca radius user-password [encrypted] password <*password*>
- Ensure the RADIUS server parameters are configured. For more information, 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*.

Configuring Redirection of Subscriber Traffic to ECS

User traffic is directed through the ECS service inspection engine by using Access Control List (ACL) mechanism to selectively steer subscriber traffic.

To configure redirection of subscriber traffic to ECS:

Step 1 Create an ECS ACL as described in Creating an ECS ACL.

Step 2 Apply an ACL to an individual subscriber as described in Applying an ACL to an Individual Subscriber, on page 78.

Step 3 Apply an ACL to the subscriber named default as described in Applying an ACL to the Subscriber Named default, on page 78.

Step 4 Apply the ACL to an APN as described in Applying the ACL to an APN, on page 78.

Important 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.

Creating an ECS ACL

To create an ACL to use in steering subscriber traffic through ECS, use the following configuration:

```
configure
  context <context_name>
    ip access-list <access_list_name>
      redirect css service <ecs_service_name> <keywords> <options>
      end
```

Notes:

- *<ecs_service_name>* must be the enhanced charging service's name; no CSS service needs to be configured.

Applying an ACL to an Individual 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 information 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*.

To apply an ACL to an individual subscriber, use the following configuration:

```
configure
  context <context_name>
    subscriber name <subscriber_name>
      ip access-group <acl_name> [ in | out ]
      end
```

Applying an ACL to the Subscriber Named default

To apply an ACL to the default subscriber, use the following configuration:

```
configure
  context <context_name>
    subscriber default
      ip access-group <acl_name> [ in | out ]
      end
```

Applying the ACL to an APN

To apply an ACL to an APN, use the following configuration:

Important This information is only applicable to UMTS networks.

```
configure
  context <context_name>
    apn <apn_name>
      ip access-group <acl_name> [ in | out ]
      end
```

Configuring GTPP Accounting

For information on configuring GTPP accounting, 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*.

Configuring EDR/UDR Parameters

This section provides an example configuration to configure EDR/UDR file transfer and file properties parameters, including configuring hard disk support on SMC card on ASR 5500, transfer modes, transfer interval, etc.

To configure EDR/UDR file parameters:

```
configure
   context <context_name>
      edr-module active-charging-service  [ charging | reporting ]
         cdr { purge { storage-limit <storage_limit> | time-limit <time_limit> }
 [ max-files <max_records_to_purge> ] | push-interval <push_interval> |
push-trigger space-usage-percent <trigger_percentage> |
remove-file-after-transfer | transfer-mode { pull [ module-only ] | push
 primary { encrypted-url <encrypted_url> | url <url> } [ [ max-files <max_records>
 ] [ module-only ] [ secondary { encrypted-secondary-url
<encrypted_secondary_url> | secondary-url <secondary_url> } ] [ via local-context
 ] + ] | use-harddisk }
         file [ charging-service-name { include | omit } ] [ compression {
 gzip | none } ] [ current-prefix <string> ] [ delete-timeout <seconds> ] [
directory <directory_name> ] [ edr-format-name ] [ exclude-checksum-record
] [ field-separator { hyphen | omit | underscore } ] [ file-sequence-number
 rulebase-seq-num ] [ headers ] [ name <file_name> ] [ reset-indicator ] [
 rotation [ num-records <number> | time <seconds> | volume <bytes> ] ] [
sequence-number { length <length> | omit | padded | padded-six-length |
unpadded } ] [ storage-limit <limit> ] [ single-edr-format ] [ time-stamp
{ expanded-format | rotated-format | unix-format } ] [ trailing-text
<string> ] [ trap-on-file-delete ] [ xor-final-record ] +
         exit
      udr-module active-charging-service
         file [ charging-service-name { include | omit } ] [ compression {
 gzip | none } ] [ current-prefix <string> ] [ delete-timeout <seconds> ] [
directory <directory_name> ] [ exclude-checksum-record ] [ field-separator
{ hyphen | omit | underscore } ] [ file-sequence-number rulebase-seq-num
 ] [ headers ] [ name <file_name> ] [ reset-indicator ] [ rotation [
num-records <number> | 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 ] [
udr-seq-num ] [ xor-final-record ] +
         end
```

Notes:

- The **cdr** command keywords can be configured either in the EDR or the UDR Configuration Mode. Configuring in one mode prevents the configurations from being applied in the other mode.

- If the **edr-module active-charging-service** command is configured without the **charging** or **reporting** keywords, by default the EDR module is enabled for charging EDRs.
- When the configured threshold limit is reached on the hard disk drive, the records that are created dynamically in the */mnt/hd-raid/data/records/* directory are automatically deleted. Files that are manually created should be deleted manually.
- The **use-harddisk** keyword is only available on the ASR 5500.

Verifying your Configuration

To view EDR-UDR file statistics, in the Exec Mode, enter the following command:

```
show active-charging edr-udr-file statistics
```

Pushing EDR/UDR Files Manually

To manually push EDR/UDR files to the configured external storage, in the Exec mode, use the following command:

```
cdr-push { all | local-filename file_name }
```

NOTES:

- Before you can use this command, the CDR transfer mode and file locations must be set to push in the EDR/UDR Module Configuration Mode.
- The **cdr-push** command is available in the Exec Mode.
- <*file_name*> must be absolute path of the local file to push.

Retrieving EDR and UDR Files

To retrieve UDR or EDR files you must SFTP into the context that was configured for EDR or UDR file generation.

This was done with the FTP-enabled account that you configured in Enabling Charging Record Retrieval, on page 65.

The following commands use SFTP to log on to a context named **ECP** as a user named **ecpadmin**, through an interface configured in the ECS context that has the IP address *192.168.1.10* and retrieve all EDR or UDR files from the default locations:

```
sftp -oUser=ecpadminECP 192.168.1.10:/records/edr/*
sftp -oUser=ecpadminECP 192.168.1.10:/records/udr/*
```

Configuring RADIUS Analyzer

This section describes how to configure the RADIUS Analyzer. When a call is established, the pre-DFA-rulebase uses the traffic that has been authenticated by the RADIUS server. Until then all the normal traffic is denied and is resumed only after the additional RADIUS based authentication is successful. The success of RADIUS authentication is determined by a RADIUS analyzer.

To configure the RADIUS Analyzer, use the following configuration:

```
configure
  active-charging service  service_name
    ruledef  ruledef_name
      [ no ] radius [ any-match < != | = > < FALSE | TRUE  > | error < != | = >
 <FALSE | TRUE  > | state < != | = > < auth-req-rcvd | auth-rsp-fail |  auth-rsp-success
  > ]
      end
```

Notes:

- **radius:** RADIUS related configuration.
- **any-match:** This command allows you to define rule expressions to match all RADIUS packets.
- **error:** This command allows you to define rule expressions to match for errors in RADIUS packets and errors in the RADIUS analyzer.
- **state:** This command allows you to define rule expressions to match the current state of an RADIUS session.

Sample Radius Analyzer Configuration

This section describes how to configure the RADIUS Analyzer feature.

To configure the RADIUS Analyzer, use the following sample configuration:

```
configure
  active-charging service s1
    ruledef rt_radius
      udp dst-port = 1812
      rule-application routing
      exit
    ruledef  radius_accept
      radius state = auth-rsp-success
      exit
```

Sample Dual Factor Authentication Configuration

This section describes how to configure the the Dual Factor Authentication (DFA) feature.

To configure the DFA Analyzer, use the following sample configuration:

```
configure
  active-charging service s1
    rulebase pre-dfa-rulebase
      action priority 1 ruledef radius_server_radius_traffic
charging-action do_nothing
     action priority 2 ruledef radius_server_icmp_traffic charging-action
 do_nothing
      action priority 3 ruledef  radius_accept charging-action
change_rbase
      action priority 100 ruledef catch_all charging-action drop
      route priority 1 ruledef rt_radius analyzer radius
      exit
    rulebase post-dfa-rbase
    exit
```

Configuring Post Processing Feature

This section describes how to configure the Post-processing feature to enable processing of packets even if rule matching for them has been disabled.

To configure the Post-processing feature, use the following configuration:

```
configure
   active-charging service <ecs_service_name>
      ruledef <ruledef_name>
         <protocol> <expression> <operator> <condition>
         rule-application post-processing
         exit
      charging-action <charging_action_name>
         ...
         exit
      rulebase <rulebase_name>
       action priority <action_priority> { [ dynamic-only | static-and-dynamic
 | timedef <timedef_name> ] { group-of-ruledefs <ruledef_group_name> | ruledef
<ruledef_name> } charging-action <charging_action_name> [ monitoring-key
<monitoring_key> ] [ description <description> ] }
        post-processing priority <priority> ruledef <ruledef_name>
charging-action <charging_action_name>
        ...
        end
```

Notes:

- In the ACS Rulebase Configuration Mode, the ruledef configured for post-processing action must have been configured for post processing in the ACS Ruledef Configuration Mode.
- If the same ruledef is required to be a charging rule in one rulebase and a post-processing rule in another rulebase, then two separate identical ruledefs must be defined.
- Post processing with group-of-ruledefs is not supported.
- Delay charging with dynamic rules is not supported, hence there cannot be dynamic post-processing rules.

Configuring Service Group QoS Feature

To create and configure a QoS-Group-of-Ruledefs, use the following configuration:

```
configure
   active-charging service <ecs_service_name>
     qos-group-of-ruledefs <qos_group_of_ruledefs_name> [ -noconfirm ] [
description <description> ]
        add-ruledef <ruledef_name>
        end
```

Notes:

- To configure flow action in the charging-action, in the ACS Charging Action Configuration Mode, use the **flow action** CLI command.

- To configure bandwidth limits for a flow, in the in the ACS Charging Action Configuration Mode use the **flow limit-for-bandwidth** CLI command.
- To view subscriber statistics and information on dynamic updates to charging parameters per call ID, in the Exec Mode, use the following command:

```
show active-charging
subscribers callid <call_id> charging-updates [ statistics ] [
charging-action [ name <charging_action_name> ] | qos-group [ name
<qos_group_of_ruledefs_name> ] ] [ | { grep <grep_options> | more } ]
```

Configuring Bandwidth Limiting

To suppress throttling at charging-action, bearer and APN level, use the following configuration:

```
configure
   active-charging service service_name
      charging-action charging_action_name
         throttle-suppress [ timeout suppress_timeout ]
         no throttle-suppress
         end
```

Notes:

- **timeout** *suppress_timeout*: Specifies the time for which throttling is suppressed, in seconds. *suppress_timeout* must be an integer from 10 through 300.
- When configured with the **timeout** keyword, bandwidth limiting is suppressed for the mentioned time.
- When configured without the **timeout** keyword, the default value of 30 seconds will apply.
- When **throttle-suppress** is configured, the timeout will take the default value of 30 seconds. The flow will not be throttled for the next 30 seconds.
- When **no throttle-suppress** is configured, bandwidth limiting will continue from the next flow onwards.

Verifying your Configuration

Verify your configuration in the Exec mode using the following commands.

- To verify the configured timeout value for suppress-throttle:

```
show active-charging charging-action name <charging_action_name> [ | { grep
 <grep_options> | more } ]
```

- To verify the number of uplink and downlink bytes that escaped bandwidth limiting due to suppressing functionality:

```
show active-charging charging-action statistics [ name
<charging_action_name> ] [ | { grep <grep_options> | more } ]
```

- To verify the total suppress time and elapsed time:

```
show active-charging flows full all [ | { grep <grep_options> | more } ]
```

- To verify the historical total and current number of flows for which bandwidth limiting is suppressed:

```
show active-charging subsystem all [ | { grep <grep_options> | more } ]
```

Configuring Flow Admission Control

To configure the TCP Proxy Flow Admission Control feature, use the following configuration:

```
configure
   active-charging service <ecs_service_name>
      fair-usage tcp-proxy max-flows-per-subscriber <max_flows>
      fair-usage tcp-proxy memory-share <memory_share>
      end
```

Notes:

- It is not necessary for the Fair Usage feature to be enabled before this configuration.
- <*max_flows*> specifies the maximum number of flows for which TCP Proxy can be used per subscriber. Note that this limit is per Session Manager.
- <*memory_share*> specifies what portion of ECS memory should be reserved for TCP Proxy flows. Note that it is a percentage value.

Verifying your Configuration

To verify your configuration, in the Exec mode, use the following command:

```
show active-charging tcp-proxy statistics [ rulebase <rulebase_name> ] [
verbose ] [ | { grep <grep_options> | more } ]
```

Configuring Time-of-Day Activation/Deactivation of Rules Feature

This section describes how to configure the Time-of-Day Activation/Deactivation of Rules feature to enable charging according to day/time.

To configure the Time-of-Day Activation/Deactivation of Rules feature, use the following configuration:

```
configure
   active-charging service <ecs_service_name>
      ruledef <ruledef_name>
         ...
         exit
      timedef <timedef_name>
       start day { friday | monday | saturday | sunday | thursday | tuesday
 | wednesday } time <hh> <mm> <ss> end day { friday | monday | saturday |
sunday | thursday | tuesday | wednesday } time <hh> <mm> <ss>
         start time <hh> <mm> <ss> end time <hh> <mm> <ss>
         exit
      charging-action <charging_action_name>
         ...
         exit
      rulebase <rulebase_name>
         action priority <action_priority> timedef <timedef_name> {
group-of-ruledefs <ruledef_group_name> | ruledef <ruledef_name> } charging-action
 <charging_action_name> [ description <description> ]
         ...
         end
```

Notes:

- In a timeslot if only the time is specified, that timeslot will be applicable for all days.
- If for a timeslot, "start time" > "end time", that rule will span the midnight, which means that rule is considered to be active from the current day till the next day.
- If for a timeslot, "start day" > "end day", that rule will span over the current week till the end day in the next week.
- In the following cases a rule will be active all the time:
  - A timedef is not configured in an action priority
  - A timedef is configured in an action priority, but the named timedef is not defined
  - A timedef is defined but with no timeslots

Verifying your Configuration

To verify your configuration, in the Exec Mode, enter the following command:

show active-charging timedef name <*timedef_name*>

Configuring Retransmissions Under Rulebase or Service Level CLI

To enable retransmission under Rulebase or Service Level base, use the following configuration:

```
configure
     active-charging service <ecs_service_name>
          rulebase <rulebase name>
               retransmissions-counted
              end
```

Notes:

- Use the **no retransmission counted** command to disable the retransmission counted feature.

To verify your configuration, in the Exec mode, enter the following command:

```
show active-charging rulebase name <rulebase_name>
```

Configuring Websockets

To enable the websocket flow detection feature, use the following configuration:

```
configure
  active-charging service <ecs_service_name>
    rulebase <rulebae name>
      websocket flow-detection  <protocol>
      end
```

Notes:

Use the **no websocket flow-detection** command or **default websocket flow-detection** command to disable websocket flow detection.

To verify your configuration, in the Exec mode, enter the following command:

```
show active-charging rulebase name <rulebase_name>
```

Configuring URL Filtering Feature

This section describes how to configure the URL Filtering feature to simplify rules for URL detection.

To create a group-of-prefixed-URLs, use the following configuration:

```
configure
  active-charging service <ecs_service_name>
    group-of-prefixed-urls <prefixed_urls_group_name>
    end
```

To configure the URLs to be filtered in the group-of-prefixed-URLs, use the following configuration:

```
configure
  active-charging service <ecs_service_name>
    group-of-prefixed-urls <prefixed_urls_group_name>
      prefixed-url <url_1>
      ...
      prefixed-url <url_10>
      end
```

To enable or disable the group in the rulebase for processing prefixed URLs, use the following configuration:

```
configure
  active-charging service <ecs_service_name>
    rulebase <rulebase_name>
      url-preprocessing bypass group-of-prefixed-urls
<prefixed_urls_group_name>
      ...
      url-preprocessing bypass group-of-prefixed-urls
<prefixed_urls_group_name>
      end
```

Notes:

- A maximum of 64 group-of-prefixed-urls can be created and configured.
- A maximum of 10 prefixed URLs can be configured in each group-of-prefixed-urls.
- In a rulebase, multiple group-of-prefixed-urls can be configured to be filtered.

Verifying your Configuration

To verify your configuration, in the Exec Mode, enter the following command:

```
show active-charging group-of-prefixed-urls name <prefixed_urls_group_name>
```

Configuring AES Encryption

This section describes how to redirect the flow to the redirect-url and encrypt the dynamic fields by using either blowfish encryption or AES encryption.

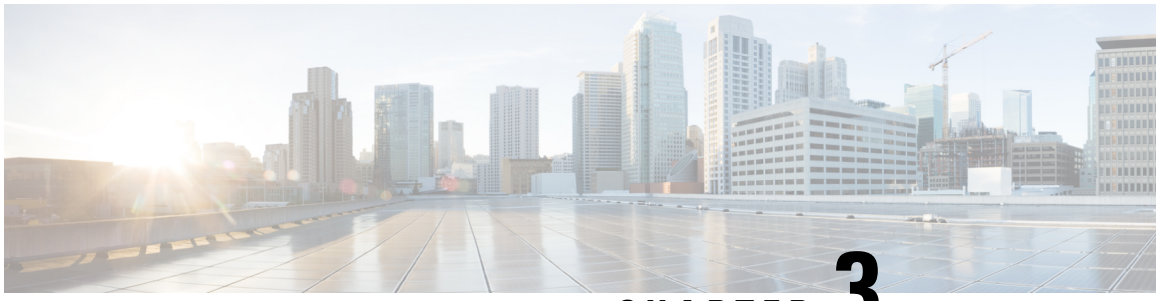
The flow action redirect-url specifies ASR to return a redirect response to the subscriber, and terminate the TCP connections (to the subscriber and server). The subscriber's Web browser automatically sends the original HTTP packet to the specified URL. Redirection is only possible for certain types of HTTP packets (for example, GET requests), which typically are only sent in the uplink direction. If the flow is not HTTP, the redirect-url option is ignored, that is the packet is forwarded normally, except for SIP. For SIP, a Contact header with the redirect information is inserted. The redirect-url consists of the redirect url and may additionally include one or more dynamic fields. Earlier, the dynamic fields could be encrypted using 128 and 256 bit blowfish encryption. The new functionality provides the additional AES-CBC encryption of the dynamic fields as well.

To redirect-URL action on packet and flow for Session Control functionality, use this configuration.

```
configure
   active-charging service <ecs_service_name>
     flow action redirect-url redirect_url [ encryption { blowfish128 |
blowfish64 | { { aes128 | aes256 } [ salt ] } } [ encrypted ] key key ] ]

     end
```

Notes:

- **aes128:** Specifies to use AES-CBC encryption with 128 bit key for encrypting the dynamic fields
- **aes256:** Specifies to use AES-CBC encryption with 256 bit key for encrypting the dynamic fields.
- **salt:** Specifies to use salt with AES-CBC encryptions of the dynamic fields



CHAPTER 3

Dedicated Bearer Creation by Service Flow Detection

This feature introduces the ability to create dedicated bearer for specific service flows based on the LAC/TAC configured on the P-GW.

Feature Information

Summary Data

Status	New Functionality
Introduced-In Release	21.2
Modified-In Release(s)	Not Applicable
Applicable Product(s)	GGSN, P-GW
Applicable Platform(s)	ASR 5500
Default Setting	Disabled
Related CDETS ID(s)	CSCvc99538
Related Changes in This Release	Not Applicable

Related Documentation	Command Line Interface Reference ECS Administration Guide Statistics and Counters Reference Statistics and Counters Reference - Counter Descriptions

Revision History

Important Revision history details are not provided for features introduced before release 21.2.

Revision Details	Release	Release Date
New in this release.	21.2	April 27, 2017

Feature Description

This feature introduces the ability to create dedicated bearer for specific service flows based on the LAC/TAC configured on the P-GW. The service flows are specified by destination IP addresses and the location is specified by TAI/ECGI/CGI. When the UE moves out from the location, this dedicated bearer is release. The dedicated bearer is created with support from the local policy on some specific service flow and subscriber location.

A new CLI keyword **activate-predef-rule** has been added to the CLI command **trigger-action** to create a dedicated bearer when enterprise user enters in corporate area and starts some specific service.

Important This feature requires standard local-policy license. Contact your Cisco account or support representative for detailed licensing information.

How It Works

This section describes how this feature works:

- On a call set-up or on location change event, local-policy sends the list of rules to be activated and deleted to the P-GW.
- P-GW then caches the list of activated rules at subscriber level.
- It then does rule matching on the flow and evaluates the trigger-condition. If the condition matches, the configured trigger action is performed, which in this case is to activate predef-rule. This initiates a new bearer request towards the UE.
- On location change event, local-policy again send the list of rules to be added or deleted towards the P-GW. If the bearer is created on any of the deleted rules, then the delete bearer request is initiated towards the UE.

Limitations

Following are the limitations of this feature:

- Service flow is identified based on the destination IP address. So, in this case if delay-charging feature is enabled then for the TCP flows the trigger conditions are evaluated only when the first application packet is received.
- If the CLI command **flow control-handshaking charge-to-application** is configured under the rulebase and if the rule matches, then the bearer request is sent on the first application packet received on that flow.
- The predef rule defined for a bearer creation must be configured in the action priority line under the rulebase. The service flow rule and the predef rule both need to be present in the same rulebase.
- The activate bearer trigger action is taken only if local-policy-rule condition is specified in the trigger-condition of service-scheme framework.
- If the create bearer request fails for any reason, then the bearer request is not retried for the UE in that location.
- The session recovery is not supported when local-policy is enabled through the APN. This is because the local-policy rule is not supported in this mode. This feature works with local-policy configured in dual mode or fallback mode. Hence, the operator must configure the dummy IMSA with fallback mechanism.
- Rule-report-status should not be configured under local-policy as in this case P-GW is triggering the bearer creation.
- The multi-line-or all-lines CLI command should not be defined under trigger-condition for **trigger-action activate-predef-rule** in the service-scheme framework.

Configuring Dedicated Bearer

The following section provides the configuration commands to enable or disable the feature.

activate-predef-rule

The CLI keyword **activate-predef-rule** has been added to the command **trigger-action** to create dedicated bearer by service flow detection at specific location.

```
configure
  active-charging service service_name
    trigger-action trigger_action_name
    [ no ] activate-predef-rule
    end
```

Notes:

- **no:** Disables predefine rule or group of rules.
- **activate-predef-rule:** Activates predefine rule or group of rules.

Sample Configuration

This section lists sample local policy configuration, service scheme framework configuration, rule charging configuration, and deletion of bearer due to inactivity.

Local Policy Configuration

```
configure
  local-policy-service LOCAL_PCC
    ruledef ruledef-tai-group
      condition priority 1 tai mcc 214 mnc 365 tac ge 10
    #exit
    ruledef ruledef-ecgi-group
      condition priority 1 ecgi mcc 214 mnc 365 eci match *
    #exit
    actiondef activate_lp_action_tai
      action priority 1 activate-lp-rule name tai_action1
    #exit
    actiondef activate_lp_action_ecgi
      action priority 1 activate-lp-rule name ecgi_action1
    #exit
    eventbase default
      rule priority 1 event new-call ruledef ruledef-tai-group actiondef
activate_lp_action_tai
      rule priority 2 event location-change ruledef ruledef-tai-group
actiondef activate_lp_action_tai
     rule priority 3 event tai-change ruledef ruledef-tai-group actiondef
 activate_lp_action_tai
    rule priority 4 event ecgi-change ruledef ruledef-ecgi-group actiondef
 activate_lp_action_ecgi
    #exit
end

configure
  context source
    ims-auth-service ims-auth
      policy-control
        associate failure-handling-template f1
        associate local-policy-service LOCAL_PCC
      exit
    exit
  exit
end

configure
  failure-handling-template f1
    msg-type any failure-type any action continue local-fallback
end
```

Service Scheme Framework

```
configure
  active-charging service acs
    subscriber-base SB1
     priority 1 subs-class SC1 bind service-scheme SS1
    #exit

    subs-class SC1
      any-match = TRUE
    #exit

    service-scheme SS1
      trigger flow-create
        priority 1 trigger-condition TC1 trigger-action TA1
    #exit

    trigger-condition TC1
      local-policy-rule = tai_action1
      rule-name = ded-bearer-rule
    #exit

   trigger-action TA1
    activate-predef-rule  predef-rule
   #exit
```

Rule-Charging Configuration

```
ruledef ded-bearer-rule
    ip dst-adress = 10.10.10.1
  #exit

  ruledef predef-rule
    tcp either-port = 80
  #exit

  charging-action ch-pkt
     qos-class-identifier 6
     tft packet-filter ip-pkts
  #exit

  charging-action standard
    content-id 5
  #exit

  packet-filter ip-pkts
    ip protocol = 6
  #exit

  rulebase consumer
    action priority 1 ruledef ded-bearer-rule charging-action standard
    action priority 5 dynamic-only ruledef predef-rule charging-action
```

```
ch-pkt
  #exit
```

Deletion of Bearer Due to Inactivity

```
configure
  context source
    apn cisco.com
      timeout bearer-inactivity non-gbr 300 volume-threshold total 1 <<
this will delete non-gbr bearers if idle for 300 seconds
     timeout bearer-inactivity exclude-default-bearer  << needs to be
configured to exclude default bearer if nongbr is defined in above CLI
    #exit
  #exit
end
```

Monitoring and Troubleshooting

This section lists the commands available to monitor the "Dedicated Bearer Creation by Service Flow Detection" feature.

Show Commands

This section lists all the show commands available to monitor this feature.

show active-charging rulebase statistics name prepaid

This command has been modified to display the following output:

```
Service Name: acs

  Rulebase Name: prepaid

     Predefined Rule Retention Statistics:
         Total number of Predefined Retention Succeeded:             0
         Total number of Predefined Retention Failed:                0

     Service Scheme Dedicated Bearer Statistics:
       Predefined Rule Installation Statistics:
         Total Number of Installation Received:                      6
         Total Number of Installation Succeeded:                     6
         Total Number of Installation Failed:                        0

       Predefined Rule Removal Statistics:
         Total Number of Removal Received:                           2
         Total Number of Removal Succeeded:                          2
         Total Number of Removal Failed:                             0
```

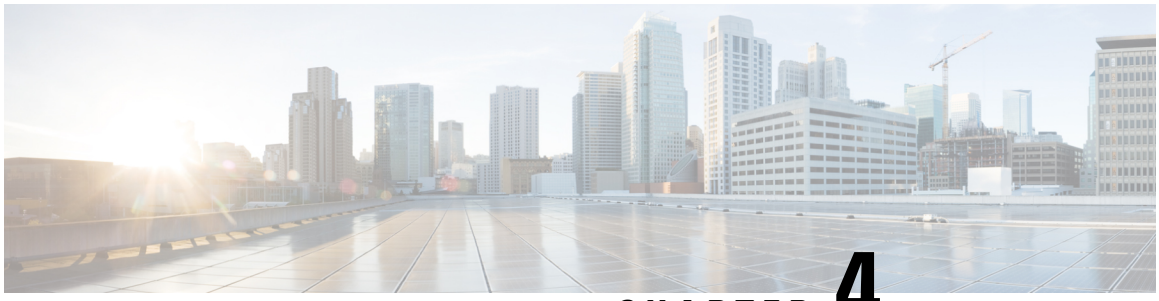
Bulk Statistics

This section lists all the bulk statistics that have been added, modified, or deprecated to support this feature.

ECS Schema

This section displays the bulk stats that have been added to display the number of bearers created and deleted for bearers created through the service-scheme framework.

- servschm-predef-rule-install-received - Number of predefined rules received for installation from Service Scheme
- servschm-predef-rule-install-succeeded - Number of predefined rules succeeded during installation from Service Scheme
- servschm-predef-rule-install-failed - Number of predefined rules installation failed from Service Scheme
- servschm-predef-rule-remove-received - Number of predefined rule removal received from Service Scheme
- servschm-predef-rule-remove-succeeded - Number of predefined rule removal successful from Service Scheme
- servschm-predef-rule-remove-failed - Number of predefined rule removal failed from Service Scheme



CHAPTER 4

Destination-Host AVP in ACR Message

This chapter describes the following topics:

Feature Summary and Revision History

Summary Data

Applicable Product(s) or Functional Area	ECS
Applicable Platform(s)	• ASR 5500 • VPC-DI • VPC-SI
Feature Default	Enabled - Always-on
Related Changes in This Release	Not Applicable
Related Documentation	*ECS Administration Guide*

Revision History

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

Revision Details	Release
In this release, the ACR message (acct-record-number=x) that is sent to the OFCS server will not have the Destination-Host AVP where its previous interim message (acct-record-number=x-1) is written into HDD. Instead, it follows the day-1 behavior like archiving request message.	21.4

Revision Details	Release
First introduced.	Pre 21.2

Feature Changes

After powering off Charging Collection Function (CCF) servers, the P-GW is not sending the destination host in the interim accounting messages. The returned failure message DIAMETER_UNABLE_TO_DELIVER (3002) keeps incrementing until an SRP switchover is performed. This issue is seen when the P-GW does not send the destination host in the interim accounting message when a previous interim message is written to the hard disk drive (HDD).

Previous Behavior

The accounting-request (ACR) message (acct-record-number=x) that is sent to the Offline Charging System (OFCS) server has the Destination-Host AVP where its previous interim message (acct-record-number=x-1) is written to the HDD.

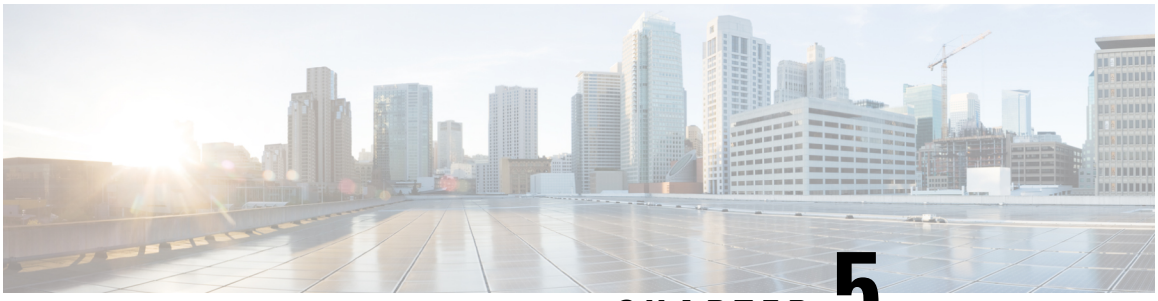
New Behavior

The ACR message (acct-record-number=x) that is sent to the OFCS server will not have the Destination-Host AVP where its previous interim message (acct-record-number=x-1) is written into HDD. Instead, follows the day-1 behavior like archiving request message.

This behavior is applied for a single server and multiple server configuration, Session Recovery scenario, and ICSR scenarios.

Impact on Customer

The new behavior is seen where the HDD is used as the RF interface.



CHAPTER 5

DNS Type Query Support Added to the DNS Analyzer

This chapter describes the following topics:

Feature Summary and Revision History

Summary Data

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

Revision History

Important Revision history details are not provided for features introduced before releases 21.2 and N5.5.

Revision Details	Release
In this release, the DNS analyzer (ECS) is enhanced to query the DNS type. This capability supports zero-rating on A type of queries and DNS tunneling on TXT and NULL type queries.	21.4
First introduced.	Pre 21.2

Feature Changes

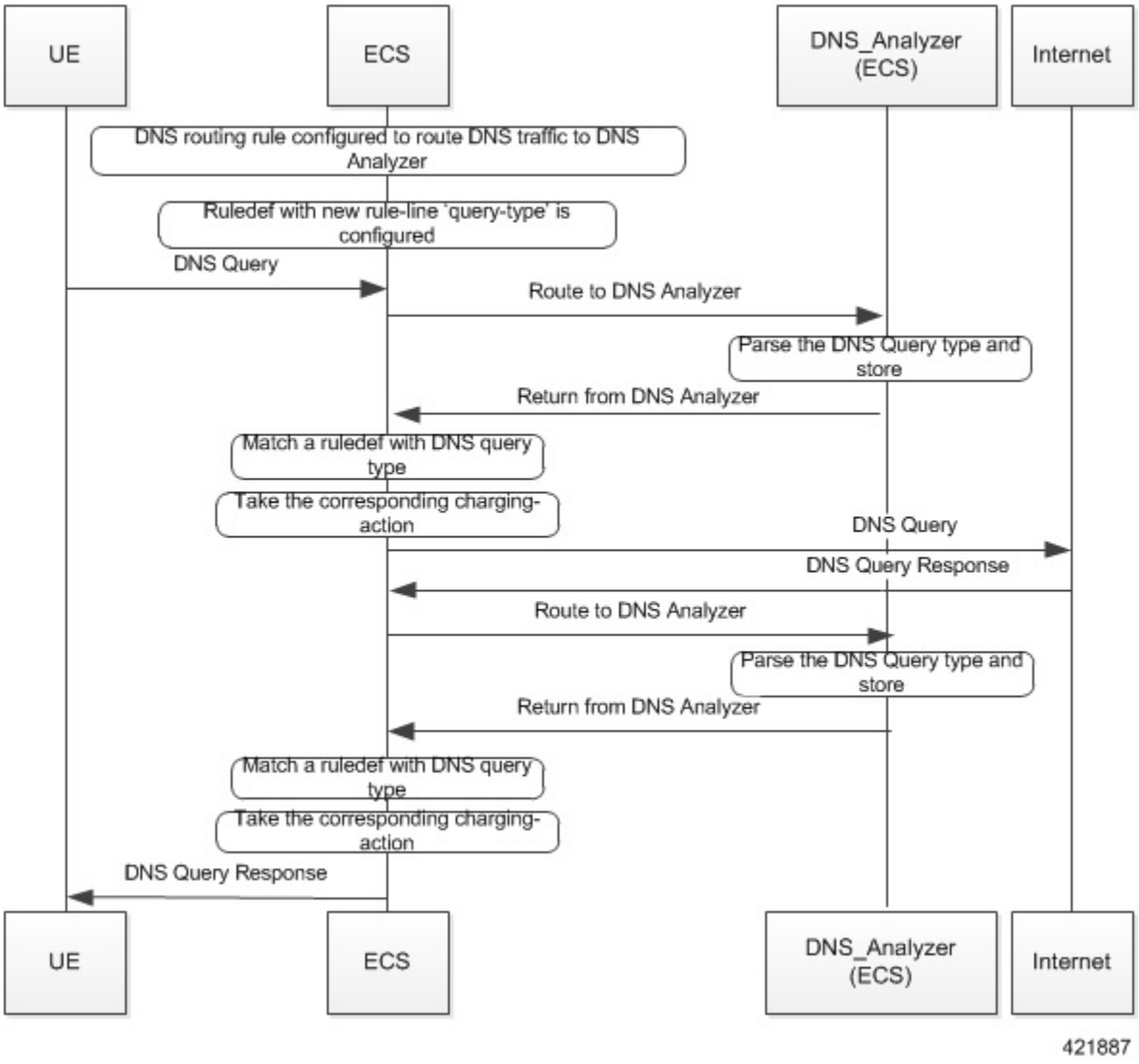
The DNS analyzer currently does not detect the type of DNS query nor does it perform zero-rating on A type DNS queries.

The new command **dns query-type** is introduced to enable the DNS analyzer (ECS) to query the DNS type to counter DNS fraud without huge impacts. This capability supports zero-rating on A type of queries and DNS tunneling on TXT and NULL type queries.

The Rule Match engine is enabled to support matching based on the query type.

The **dns query-type** command defines rule expressions to match the query type in the DNS request messages. This command is added under the ACS Ruledef Configuration Mode.

The following call flow displays how the DNS analyzer (ECS) detects the type of DNS query.



Previous Behavior

DNS query types based rule-matching never occurred. If there were multiple answers, unsupported query-type skipped parsing the complete answer.

New Behavior

The following DNS query types can be configured in a ruledef. These are parsed and rule-matched.

- A
- CNAME
- NS
- PTR
- SRV

- AAAA
- TXT
- ANY
- NULL

If there are multiple answers, unsupported query-type skips parsing only that answer and continues parsing the next answer.

Customer Impact

DNS packets now start matching the query-type ruledefs.

Command Changes

dns query-type

This new command is added under the ACS Ruledef Configuration mode to define rule expressions to match the query type in the DNS request messages.

When enabled, the **dns query-type** CLI supports the following behavior:

- DNS request with only one query is supported.
- DNS response with multiple answers is supported. Query-type corresponding to all the answers is stored and matched to the highest priority ruledef.
- For DNS response with multiple answers, unsupported query-type (mentioned previously) is skipped and parsing continues for remaining answers.
- For TXT and NULL query types, minimal parsing occurs like only a DNS record is created and query-type is stored. Answer-name is not extracted and hence the corresponding EDR field is not populated.
- For NULL query types, response is not parsed and matching is based on the same ruledef as a Request.

```
configure
    active-charging service  service_name
       ruledef  ruledef_name
        [ no ] dns query-type operator query_type
        end
```

Notes:

- **no**: Disables this feature, that is, the query-name ruleline is removed from the DNS protocol.
- **operator:** Specifies how to match.

  operator must be one of the following:

  - **=**: Specifies that the query-name must be equal to the one specified.
  - **!=**: Specifies that the query-name must not be equal to the one specified.

- **query-type**: Specifies the type of queries supported: a, cname, ns, ptr, srv, aaaa, txt, any, and null.
- This CLI is disabled by default.

Performance Indicator Changes

show active-charging analyzer statistics name dns

The output of this command now includes the following new fields (TXT and NULL query types) depending on whether the CLI is enabled or disabled:

```
show active-charging analyzer statistics name dns
ACS DNS Session Stats:
  Total Uplink Bytes:               0   Total Downlink Bytes:               0
  Total Uplink Pkts:                 0   Total Downlink Pkts:                 0
  Unknown OPCODE:             0   Invalid Pkts:                               0

  DNS Over TCP:
    Uplink Bytes:                       0   Downlink Bytes:                     0
    Uplink Pkts:                         0   Downlink Pkts:                        0

  Request:
    A Query Type:                     0   CNAME Query Type:               0
    NS Query Type:                   0   PTR Query Type:                     0
    SRV Query Type:                0   Unknown Query Type:             0
    AAAA Query Type:            0    TXT Query Type:                    0
    NULL Query Type:             0
  Response:
    A Query Type:                     0   CNAME Query Type:               0
    NS Query Type:                   0   PTR Query Type:                       0
    SRV Query Type:                0   Unknown Query Type:             0
    AAAA Query Type:            0    TXT Query Type:                    0
    NULL Query Type:               0
```

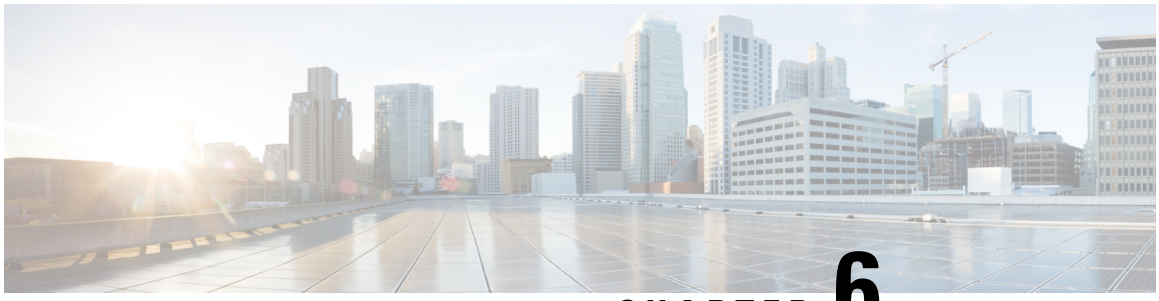
Bulk Statistics

This section lists all the bulk statistics that have been added, modified, or deprecated to support this feature.

ECS Schema

This section displays the new bulk stats that are collected for the new query types:

- dns-req-txt-query—Indicates the number of DNS queries with 'TXT' query type.
- dns-rsp-txt-query—Indicates the number of DNS answers with 'TXT' query type.
- dns-req-null-query—Indicates the number of DNS queries with 'NULL' query type.
- dns-rsp-null-query—Indicates the number of DNS answers with 'NULL' query-type.



CHAPTER 6

DNS Snooping

This chapter describes the DNS Snooping feature and provides detailed information on the following topics:

Feature Summary and Revision History

Summary Data

Applicable Product(s) or Functional Area	ECS
Applicable Platform(s)	• ASR 5500 • VPC-DI • VPC-SI
Feature Default	Disabled - License Required
Related Changes in This Release	Not applicable
Related Documentation	*Command Line Interface Reference*

Revision History

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

Revision Details	Release
First introduced.	Pre 21.2

Feature Description

This section provides an overview of the DNS Snooping feature.

Important In the 12.2 release, the DNS Snooping feature is supported only on the GGSN and P-GW.

ECS, using L7 rules, can be configured to filter subscriber traffic based on domain name. While this works fine for HTTP-based traffic, a subscriber's initial HTTP request may result in additional flows being established that use protocols other than HTTP and/or may be encrypted. Also, a domain may be served by multiple servers, each with its own IP address. This means that using an IP rule instead of an HTTP rule will result in multiple IP rules, one for each server "behind" the domain. This necessitates service providers to maintain a list of IP addresses for domain-based filters.

The DNS Snooping feature enables a set of IP rules to be installed based on the response from a DNS query. The rule in this case contains a fully qualified domain name (for example, m.google.com) or its segment (for example, google) and a switch that causes the domain to be resolved to a set of IP addresses. The rules installed are thus IP rules. Any actions specified in the domain rule are inherited by the resulting IP rules.

When configured, DNS snooping is done on live traffic for every subscriber.

The DNS Snooping feature enables operators to create ruledefs specifying domain names or their segments. On defining the ruledefs, the gateway will monitor all the DNS responses sent towards the UE, and snoop only the DNS response that has q-name or a-name as specified in the rules, and identify all the IP addresses resulting from the DNS response. A table of these IP addresses is maintained per destination context per rulebase per instance and shared across subscribers of the same destination context same rulebase per instance. In case DNS queries made by different subscribers produce different results, all the IP entries in the table are stored based on their Time to Live (TTL) and the configurable timer. The TTL or the timer whichever is greater is used for aging out the IP entry. Dynamic IP rules are created for these IP entries within the same rule having the domain name, applying the same charging action to these dynamic rules. This solution will have the exact IP entries as obtained live from snooping DNS responses. They will be geographically and TTL correct.

License Requirements

DNS Snooping 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 the *Software Management Operations* chapter in the *System Administration Guide*.

Limitations and Dependencies

This section identifies limitations and dependencies for the DNS Snooping feature.

- On a SessMgr kill or card switchover, the dynamic IP rules created based on domain name resolution will be lost. Until a new DNS query is made, the dynamic IP based rules will not be applied. These rules will be recreated on new DNS traffic. So, SessMgr recovery is not supported for these dynamic IP rules.

- The **ip server-domain-name** ruledef can be used as a predefined dynamic rule, static rule, or as a part of group of ruledefs. However, it cannot be used as a dynamic-only rule, as dynamic-only rules apply up to L4 and this is an L7 rule.
- Operators must define valid domain-name servers, the DNS responses from which will be considered correct and snooped and included in the list of dynamic-learnt IP addresses. If the list of valid domain-name servers is not provided, then the DNS responses from all DNS servers will be considered valid and included in the list of learnt IP addresses. Also, in case subscribers make DNS queries to their self-created DNS servers and hack the response being sent, it can result in inclusion of invalid IP addresses in the list. In this case, the IP addresses will be learnt and the traffic may be free-rated or blocked incorrectly depending on the action set. Therefore the above is suggested to avoid attacks on DNS traffic.
- There is a limit on the total number of learnt IP addresses per server-domain-name ruledef for memory and performance considerations. Any more IP addresses across this limit will not be learnt and hence the charging-action will not be applied to these IP addresses. Similarly, there is a limit on the total number of server-domain-name ruledefs that can be configured.
- If same IP address is returned in DNS responses for different DNS q-names (same IP hosting multiple URLs), than while rule matching, the higher priority rule having this learnt-IP address will be matched. This can have undesired rule-matching as explained next.

  For example, if DNS queries for both www.facebook.com and www.cnn.com returned the IP address 162.168.10.2. Here we have allow action for domain www.facebook.com and block or no action for www.cnn.com which is at a lower priority than allow rule. In this if the actual request for www.cnn.com comes than as the server IP is same, it will match the higher priority allow rule for domain www.facebook.com (considering there are no other rule lines or all lines match) and thus, free rated incorrectly. However, this will happen only of same IP address is returned for different q-names, which is rare and cannot be handled.
- In the 12.2 release, the lookup for IPv6 learnt IP addresses will not be optimized. Hash based lookup (optimization) is done for IPv4 address lookup. In a later release Longest Prefixed Match (LPM) based optimization will be considered for both IPv4 and IPv6 learnt IP address matching.

How It Works

This section describes how the DNS Snooping feature works.

ECS allows operators to create ruledefs specifying domain names or their segments using options available in the CLI ruledef syntax (contains, starts-with, ends with, or equal to). This allows operators to match all the traffic going to specified fully qualified domain names as presented by the UE in the DNS queries, or segments of the domain names.

Internally, when a ruledef containing ip server-domain-name keyword is defined and the ruledef is used in a rulebase, an IP table similar to the following is created per rulebase per instance.

Operator	Domain Name	IP Pool Pointer	Associated Ruledef	List of CNAMES
contains	gmail	ip-pool1	domain_google	l.google.com
=	yahoo.com	ip-pool2	domain_yahoo	
starts-with	gmail	ip-pool3	domain_start_gmail	

On definition of the ruledefs, the gateway will monitor all the DNS responses sent towards the UE and will snoop the DNS responses from valid DNS servers. IP addresses (IPv4 and IPv6) resulting from the DNS responses are learnt dynamically and will be used for further rule matching. These dynamic Service Data Flows (SDFs), containing IP addresses, may also be reused by ECS for other subscribers from the same routing instance in order to classify the subscriber traffic.

The dynamic SDFs generated are kept for the TTL specified in the DNS response plus a configurable timer that can be added to the TTL in case the DNS response contains a very small TTL.

Important If the rule created using this feature is removed from the configuration then all the associated dynamic SDFs are removed immediately. The usage incurred by the subscriber for traffic matching the removed SDFs will be reported over the Gy interface when the usage reporting for the corresponding rating group is due.

In case DNS queries made by different subscribers produce different results, all the dynamically generated SDFs are stored based on their TTL and the configured timer.

DNS Snooping supports DNS responses containing nested CNAME responses.

When the DNS response contains nested CNAME record, a list per entry in the IP-table is dynamically allocated to store the CNAME. CNAME is the canonical name of the alias, which means the q-name to which the actual query was made is the alias name and this CNAME is the actual domain name to which the query should be made. So, the IP addresses found in response to CNAME DNS query is stored in the same IP-pool as that of the alias.

Here, either the DNS response to the actual alias contains CNAME record along with its A record or only the CNAME record. In the first case the IP address is already resolved for CNAME and it is included in the learnt IP addresses IP-pool.

In both the scenarios, the list of CNAMES is stored in the same record of the IP-table, which is keyed by operator+domain. By default, the operator for CNAME is "equal". So, while snooping DNS responses, DNS responses for a-name as in the CNAME list will also be snooped and the IP addresses stored in the corresponding IP-pool. This allows the feature to work in case DNS responses have nested CNAME response.

Like IP addresses, even CNAME entries have TTL associated with them. In the same five minute timer, where the aged IP addresses are timed out, the CNAME entries will also be looked at and the expired CNAME entries reference removed from the corresponding entry.

The DNS Snooping feature supports both IPv4 and IPv6 addresses. The following are the maximum limits:

- IPv4 addresses learnt per server-domain-name pattern: 200
- IPv4 addresses learnt per instance across all IPv4 pools: 51200
- IPv6 addresses learnt per server-domain-name pattern: 100
- IPv6 addresses learnt per instance across all IPv6 pools: 25600

Rule matching: While matching rule for IP packets, it will be checked if the source IP address matches any of the entries stored in the IP pools formed as part of DNS snooping. If a match is found, the corresponding ruledef is determined from the IP table. The other rule lines of the rule are matched, and if it is the highest priority rule matched it is returned as a match. The corresponding charging-action is applied. So the same priority as that of the domain name is applied to its corresponding IP addresses, and is matched as a logical OR of the domain or the IP addresses.

Lookup (matching) is performed in learnt IP pools only for the first packet of the ADS as the destination IP address will not change for that flow, and will match the same rule (last rule matched for this ADS flow) for all the packets of the flow. This enables to have the same rule matched even if its IP addresses get aged out when the flow is ongoing.

In 12.3 and earlier releases, the CLI command **show active-charging dns-learnt-ip-addresses statistics sessmgr all** displayed all the configured patterns and rulebase names for each pattern entry, even though the pattern has not learnt any IP address.

When a large number of DNS snooping ruledefs are configured (configured as ip server-domain name under ruledef configuration), the memory allocated for sending this information exceeds the message size limit for messenger calls and hence the crash is observed.

In 14.0 and later releases, the **show active-charging dns-learnt-ip-addresses statistics sessmgr all** CLI command will be displaying only the patterns for which at least one IPv4/IPv6 address is learnt as all other information is available from the configuration.

The following call flow illustration and descriptions explain the working of the DNS Snooping feature.

Figure 12: DNS Snooping Call Flow

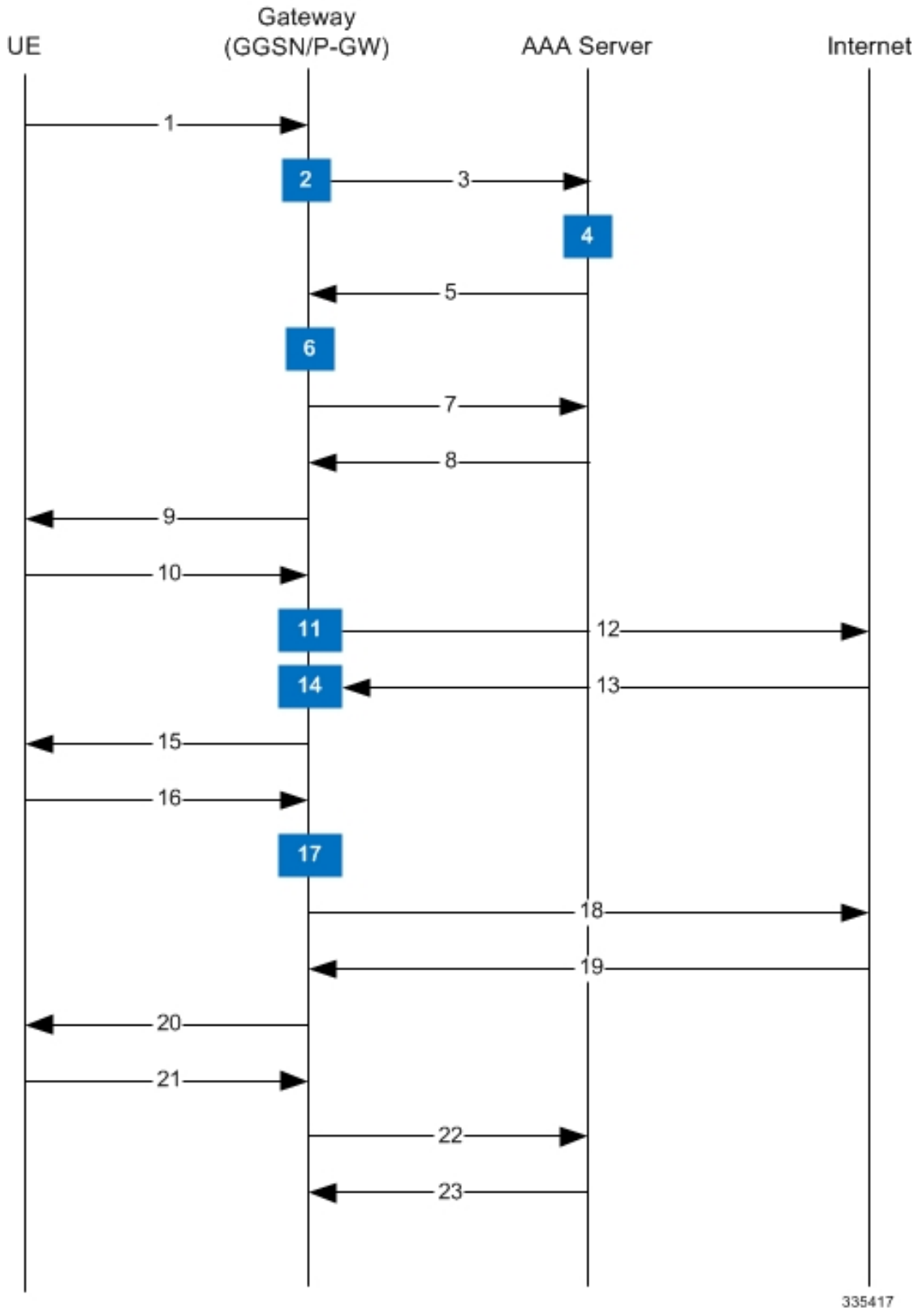

Gateway
(GGSN/P-GW)
UE
AAA Server
Internet
1
2
3
4
5
6
7
8
9
10
11
12
14
13
15
16
17
18
19
20
21
22
23
335417

Table 5: DNS Snooping Call Flow Descriptions

Step No.	Description
1	UE requests the system for registration.
2	System processes UE-related information with ECS subsystem.
3	System sends AAA Access Request to AAA server for UE.
4	The AAA server processes the AAA Access Request from the ECS to create the session, and the Policy Manager in AAA server uses subscriber identification parameters including NAI (usernamedomain), Calling Station ID (IMSI, MSID), and Framed IP Address (HoA) as the basis for subscriber lookup.
5	The Policy Manager and AAA generate and send an Access Accept message including all policy and other attributes to establish the session to ECS. The Policy Manager and/or AAA include following attributes in the Access Accept message: • Filter ID or Access Control List Name: Applied to subscriber session. It typically contains the name of the Content Service Steering (CSS) ACL. The CSS ACL establishes the particular service treatments such as Content Filtering, ECS, Stateful Firewall, VPN, etc. to apply to a subscriber session, and the service order sequence to use in the inbound or outbound directions. Real-time or delay sensitive flows are directly transmitted to the Internet with no further processing required. In this case, no CSS ACL or Filter ID is included in the Access Response. • SN1-Rulebase Name: This custom attribute contains information such as consumer, business name, child/adult/teen, etc. The rulebase name identifies the particular rule definitions to apply. Rulebase definitions are used in ECS as the basis for deriving charging actions such as prepaid/postpaid volume/duration/destination billing and charging data files (EDRs/UDRs). Rulebase configuration is defined in the ACS Configuration Mode and can be applied to individual subscribers, domains, or on per-context basis.
6	ECS creates a new session for UE, and sends the rulebase to ACS subsystem if required.

Step No.	Description
7	ECS sends Accounting-Start messages to the AAA server.
8	The AAA server sends Accounting-Start response message to ECS.
9	ECS establishes data flow with UE.
10	UE requests for data with URL name (DNS query).
11	ECS analyzes the query-name from the subscriber's DNS query, and if it matches the entry in the "DNS URLs to be snooped" list (created when ip server-domain-name rules were defined in rulebase), it marks this request for its response to be snooped.
12	DNS query is sent to the Internet.
13	DNS response is received from the Internet.
14	Based on the various answer records in the response the IP addresses are snooped and included in the "list of learnt IP addresses".
15	DNS response is sent to the UE.
16	Actual URL request comes from the UE.
17	Looking at the server-ip-address of the packet, rule matching will be done based on the "list of learnt IP addresses" and the rules already configured. An action is taken based on the ruledef matched and the charging action configured.
18	If the packet is to be forwarded, it is forwarded to the Internet.
19	A response is received from the Internet.
20	The response is sent to the UE.
21	UE requests for session termination.
22	System sends Accounting-Stop Request to AAA server.
23	AAA server stops accounting for subscriber and sends Accounting-Stop-Response to the system.

Configuring DNS Snooping

Use the following configuration to configure the DNS Snooping feature:

```
configure
   active-charging service <ecs_service_name>
      ip dns-learnt-entries timeout <timeout_period>
      ruledef <ruledef_name>
        ip server-domain-name { = | contains | ends-with | starts-with }
<domain_name/domain_name_segment>
        ...
        exit
      rulebase <rulebase_name>
        action priority <priority> ruledef <ruledef_name> charging-action
<charging_action_name>
        ...
        end
```

Verifying the DNS Snooping Configuration

Enter the following command to check the number of DNS learnt IP-entries per ruleline.

```
show active-charging dns-learnt-ip-addresses statistics sessmgr { all |
instance instance | summary | [ verbose ] }
```

Monitoring and Troubleshooting the DNS Snooping feature

This section provides information regarding bulk statistics, show commands and/or their outputs in support of this feature.

show active-charging dns-learnt-ip-addresses statistics sessmgr instance <instance> verbose

The following fields display the statistics related to the DNS Snooping feature.

- Sessmgr Instance
- Pattern
- Rulebase
- List of CNAMES
- Destination Context
- Total-ipv4-entries
- Ipv4-Entries-flushed
- Ipv4-TTL-replaced

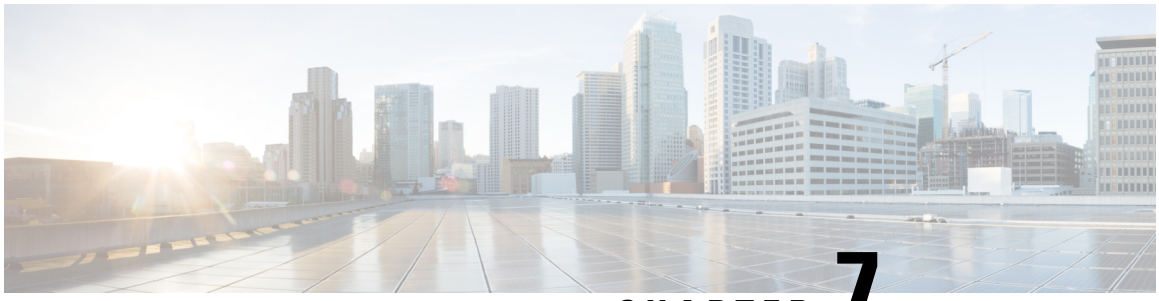
- Ipv4-Overflows
- Total-ipv6-entries
- Ipv6-Entries-flushed
- Ipv6-TTL-replaced
- Ipv6-Overflows
- Ipv4 Address TTL (in secs)
- Ipv6 Address TTL (in secs)
- Summary:
  - Total learnt ipv4 entries
  - Total learnt ipv6 entries

Bulk Statistics

Bulk statistics reporting for the DNS Snooping feature is supported.

The following bulk statistics are available in the ECS schema:

- ecs-dns-learnt-ipv4-entries
- ecs-dns-flushed-ipv4-entries
- ecs-dns-replaced-ipv4-entries
- ecs-dns-overflown-ipv4-entries
- ecs-dns-learnt-ipv6-entries
- ecs-dns-flushed-ipv6-entries
- ecs-dns-replaced-ipv6-entries
- ecs-dns-overflown-ipv6-entries



CHAPTER 7

Enhanced MBR and APR-AMBR Enforcement Support

Feature Summary and Revision History

Summary Data

Applicable Product(s) or Functional Area	• GGSN • P-GW • SAEGW
Applicable Platform(s)	• ASR 5500 • VPC - DI • VPC - SI
Feature Default	Disabled - Configuration Required
Related Changes in This Release	Not Applicable
Related Documentation	• *Command Line Interface Reference* • *ECS Administration Guide* • *Statistics and Counters Reference*

Revision History

Revision Details	Release
First introduced.	21.7

Feature Description

The token replenishment time for maximum bit rate (MBR) is currently hardcoded at 1 ms. This causes the Cisco P-GW to flat out traffic, which causes the RAN to see no burst traffic. Therefore, the RAN scheduler is unable to work efficiently. To improve the efficiency of the RAN scheduler and to cause the RAN scheduler to see burst traffic, it is necessary to increase the token replenishment time. The Enhanced MBR and APR-AMBR Enforcement Support feature addresses this requirement.

How It Works

The new MBR and APR-AMBR enforcement logic is implemented as described in the following sections.

MBR Enforcement Logic

A new token replenishment interval for MBR enforcement is introduced that is configurable at the APN and Global Configuration level. The APN level configuration takes precedence over the Global Configuration level.

The following example describes the change:

```
flow limit-for-bandwidth direction downlink peak-data-rate 4000000 peak-burst-size 500000
violate-action discard
```

The Peak-Data-Rate (MBR) is set at 4 Mbps (4000000/8 = 500 KBps) and Peak-Burst-Size is 500 KBps (500000).

1. Token Replenishment Interval is 1 msec:

   MBR of 4 Mbps means 4000000/1000 = 4000/8 = 500 bytes token is accumulated and allowed every 1 msec. Therefore, 1 packet is passed every 3 ms. Initial burst is 500 KBs (assumption is that each packet size is 1500 bytes).

2. Token Replenishment Interval is 10 msec:

   MBR of 4 Mbps means 4000000/100 = 40000/8 = 5000 bytes token is accumulated and allowed every 10 msecs. Therefore, 3 packets are passed during a 10 ms interval. Initial burst is 500 KBs (assumption is that each packet size is 1500 bytes),

3. Token Replenishment Interval is 100 msec:

   MBR of 4 Mbps means 4000000/10 = 400000/8 = 50000 bytes token is accumulated and allowed every 100 msecs. Therefore, 33 packets are passed during a 100 ms interval. Initial burst is 500 KBs (assumption is that each packet size is 1500 bytes).

4. Token Replenishment Interval is 500 msec:

MBR of 4 Mbps means 4000000/2 = 2000000/8 = 250000 bytes token is accumulated and allowed every 500 msecs. Therefore, 166 packets are passed during a 500 msec interval. Initial burst is 500 KBs (assumption is that each packet size is 1500 bytes).

5. Token Replenishment Interval is 1000 msec (1sec):

   MBR of 4 Mbps means 4000000/1 = 4000000/8 = 500000 bytes token is accumulated and allowed every 1000 msecs. Therefore, 333 packets are passed during a 1 sec interval. Initial burst is 500 KBs (assumption is that each packet size is 1500 bytes).

APN-AMBR Enforcement Logic

A new token replenishment interval for MBR enforcement is introduced that is configurable at the APN Configuration level. For more details, see the "Configuring MBR and APN-AMBR Enforcement" section.

Recommendations

The following is recommended while configuring the token replenishment interval for MBR and APN-AMBR enforcement.

1. To achieve data rate, not more than peak-data-rate at any point of time, it is recommended to configure peak-burst-size equals to peak-data-rate (MBR) in bytes.

   Examples:

   For flow level bandwidth limiting:

   ```
   flow limit-for-bandwidth direction downlink peak-data-rate 4000000 peak-burst-size 500000
    violate-action discard
   ```

   For dynamic rule bandwidth limiting:

   ```
   policy-control burst-size auto-readjust duration 1
   ```

Note Currently, default value of burst-size for dynamic rule bandwidth limiting is 5 times the MBR value.

2. If violate-action is configured as lower-ip-precedence, new MBR enforcement algorithm based on token replenishment interval, forwards packets, with zero ToS marked, if rate is beyond the configured MBR value. This may not improve efficiency of RAN side scheduler. Therefore, it is recommended to use violate-action as discard.
3. The burst size for APN-AMBR should be configured as [ambr (bps) / 8] bytes or if auto-readjust is used, duration should be 1 sec.

   Examples:

   AMBR downlink received from PCRF is 4000000 (4 Mbps)

   ```
   apn-ambr rate-limit direction downlink burst-size 500000 violate-action drop
   ```

   or

   ```
   apn-ambr rate-limit direction downlink burst-size auto-readjust duration 1 violate-action
    drop
   ```

4. For APN-AMBR enforcement, it is recommended to use violate-action as drop.

5. For APN-AMBR violate-action shape, it is recommended to configure token replenishment interval as either 100 ms or 10 ms. No other token replenishment intervals are not supported for APN-AMBR shaping.

Limitations

The following restrictions are applied to the MBR and APN-AMBR enforcement logic:

- After redundancy actions (like inter- and intra-chassis session recovery), new MBR enforcement logic will use token replenishment interval time from the latest configuration during recovery. In other words, if token replenishment interval is changed on the fly, after redundancy action, all existing subscriber session will use the latest configured token replenishment interval.
- If violate-action is configured as lower-ip-precedence, packets are forwarded with zero ToS marked, if the rate is beyond configured MBR value. This may not improve efficiency of RAN side scheduler. Therefore, it is recommended to use violate-action as discard.
- For APN-AMBR violate-action shape, it is recommended to configure token replenishment interval as either 100 ms or 10 ms. No other token replenishment intervals are not supported for APN-AMBR shaping.

Configuring MBR and APN-AMBR Enforcement

The following section provides the configuration commands to enable or disable the feature.

Configuring APN-AMBR Enforcement (APN level)

Use this command to configure token replenishment interval at APN level for APN-AMBR. This command is configured in the APN Configuration Mode.

```
configure
  context context_name
    apn apn_name
      [ default ] apn-ambr rate-limit token-replenishment-interval {
10ms [ multiplication-factor < 2..100 > ] | 100ms }
      end
```

NOTES:

- **default**: Configures default token replenishment interval at APN level for apn-ambr. Default token replenishment interval for apn-ambr is 100 ms.
- **apn-ambr**: Configures apn-ambr attributes for all PDNs of the APN.
- **rate-limit**: Configures rate-limit parameters.
- **token-replenishment-interval**: Configures token-replenishment-interval. The available values range from 10ms to 1000ms (1 sec). Token-replenishment-interval value other than 100 ms or 10 ms is not valid for violate-action shape.
- **multiplication-factor**: Configures multiplication factor of 10 ms as token replenishment interval. Multiplication-factor is configurable only if token replenishment interval is 10 ms.

- The burst size should be configured as [ambr (bps) / 8] bytes or if auto-readjust is used, duration should be 1 sec.
- By default, this CLI is disabled.

Configuring MBR Enforcement (Active Charging Service level)

Use this command to configure token replenishment interval for MBR enforcement at the Active Charging Service level. This command is configured in the ACS Service Configuration Mode.

```
configure
  context context_name
    apn apn_name
      [ no ] policy-control token-replenishment-interval { 10ms [
multiplication-factor < 2..100 > ] }
      end
```

NOTES:

- **no**: Disables token replenishment interval at Active Charging Service level.
- **token-replenishment-interval**: Configures token-replenishment-interval. The available values range from 10 ms to 1000 ms (1 sec).
- **multiplication-factor**: Configures multiplication factor of 10 ms as token replenishment interval. Multiplication-factor is configurable only if token replenishment interval is 10 ms.
- By default, this CLI is disabled.

Configuring MBR Enforcement (APN level)

Use this command to configure token replenishment interval for MBR enforcement at the APN level. This command is configured in the APN Configuration Mode.

```
configure
  context context_name
    apn apn_name
      [ no ] mbr rate-limit token-replenishment-interval { 10ms [
multiplication-factor < 2..100 > ] }

      end
```

NOTES:

- **no**: Disables token replenishment interval at the APN level.
- **mbr**: Configures MBR attributes for all PDNs of the APN.
- **rate-limit**: Configures rate-limit parameters.
- **token-replenishment-interval**: Configures token-replenishment-interval. The available values range from 10 ms to 1000 ms (1 sec).
- **multiplication-factor**: Configures multiplication factor of 10 ms as token replenishment interval. Multiplication-factor is configurable only if token replenishment interval is 10 ms.

- 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 Commands and/or Outputs

The output of the following CLI command has been enhanced in support of the feature.

show apn <apn_name>

This show command CLI now includes the value for the following new field when token replenishment interval is configured for the specified APN at the APN level:

token-replenishment-interval

show configuration (Active Charging Service Level)

This show command CLI now includes the values for the following new fields when token replenishment interval is configured at the Active Charging Service (ACS) level:

- token-replenishment-interval
- multiplication-factor

show configuration (APN level)

This show command CLI now includes the values for the following new fields when token replenishment interval is configured at the APN level:

For MBR Enforcement:

- mbr
- rate-limit
- token-replenishment-interval
- multiplication-factor

For APN-AMBR Enforcement:

- apn-ambr
- rate-limit
- token-replenishment-interval
- multiplication-factor

show configuration verbose (Active Charging Service Level)

This show command CLI now includes the value for the following new field when token replenishment interval is configured at the Active Charging Service (ACS) level:

token-replenishment-interval

show apn <apn_name>

This show command CLI now includes the value for the following new field when token replenishment interval is configured for the specified APN at the APN level:

token-replenishment-interval

show configuration (Active Charging Service Level)

This show command CLI now includes the values for the following new fields when token replenishment interval is configured at the Active Charging Service (ACS) level:

- token-replenishment-interval
- multiplication-factor

show configuration (APN level)

This show command CLI now includes the values for the following new fields when token replenishment interval is configured at the APN level:

For MBR Enforcement:

- mbr
- rate-limit
- token-replenishment-interval
- multiplication-factor

For APN-AMBR Enforcement:

- apn-ambr
- rate-limit
- token-replenishment-interval
- multiplication-factor

show configuration verbose (Active Charging Service Level)

This show command CLI now includes the value for the following new field when token replenishment interval is configured at the Active Charging Service (ACS) level:

token-replenishment-interval

show configuration verbose (APN level)

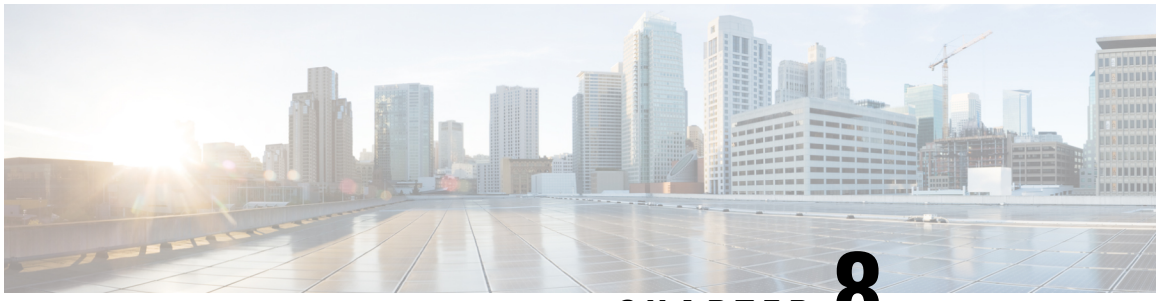
This show command CLI now includes the value for the following new field when token replenishment interval is not configured at the APN level:

For MBR Enforcement:

- mbr
- rate-limit
- token-replenishment-interval

For APN-AMBR Enforcement:

- apn-ambr
- rate-limit
- token-replenishment-interval



CHAPTER 8

Extraction of IPv4 Addresses Embedded in IPv6 Addresses

Feature Summary and Revision History

Summary Data

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

Revision History

Revision Details	Release
With this release, support is added for extraction of IPv4 addresses embedded in IPv6 addresses using the Command Line Interface (CLI).	21.17.16

Feature Description

Learning the IPv4 address, which is embedded in IPv6 address through DNS snooping, requires matching of IPv4 format against the address learnt from the DNS response.

In the release 21.17.16, IPv4 extraction is done by enhancing the existing Command Line Interface (CLI) for Well-known prefix and Network-specific prefix. For more information on prefixes, refer RFC6052 document.

After the required changes are done in the CLI, IPv4 address extraction happens and the lookup of IPv4 address is done using the learnt address pool.

Relationships to other Features

This feature is related to DNS Snooping feature. For more information about DNS Snooping feature, refer the *DNS Snooping* chapter in the *ECS Administration Guide.*

License Requirements

The Extraction of IPv4 Addresses Embedded in IPv6 Addresses requires the same DNS Snooping 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 the *Software Management Operations* chapter in the *System Administration Guide*.

How it Works

The following procedure describes the steps to be followed for IPv4 address extraction:

1. P-GW monitors all responses sent to the UE.
2. P-GW snoops only the DNS response and identifies all the IP addresses resulting from the DNS response.
3. The first data packet from IPv4 device reaches P-GW.
4. The Session Manager receives data indication and routes the packet to the ACS manager.
5. The ACS manager analyzes the packet and assigns data session for the flow.
6. Prefix matching is done based on the configured prefix.

Based on the matching, IPv4 address is extracted and it is stored in the ACS data session. Then, IPv4 address starts the lookup in the IPv4 address pool and if it matches, then the traffic is matched with the DNS snooping rule. If match does not happen, then it starts to check for other rules.

Restrictions

This section identifies the restrictions to be applied in CLI for IPv4 address extraction.

Prefix-Set Restrictions:

- Allows network-specific prefixes, well-known prefixes but restricts other prefixes.
- Restricts configuring multiple mask values under the same prefix-set.

- Restricts prefix removal from prefix-set, if the same prefix-set is associated with rule base-strip CLI.
- Restricts prefix-set removal, if the same prefix-set is associated with rule base-strip CLI.

Rule base Restrictions:

- Allows network-specific prefixes, well-known prefixes but restrict other prefixes.
- Restricts strip CLI configuration, if rulebase prefix length is not matched to the associated prefix-set mask value.
- Restricts strip CLI configuration, if the rule base associated prefix-set is invalid.
- Restricts strip CLI configuration, if the available prefix-set is empty.

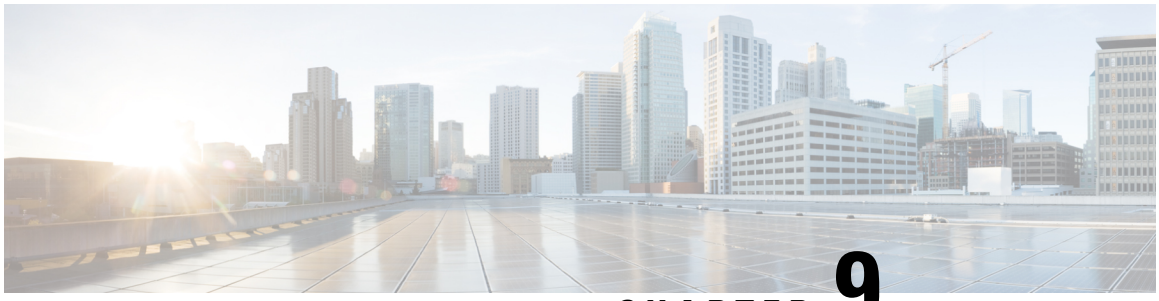
Associating Rulebase to Prefix-Set

Use the following configuration to associate rulebase to the prefix-set.

```
configure
  active-charging service ecs_service_name
    prefix-set prefix_set_name
      exit
    rulebase <rulebase_name>
      strip server-ipv6 prefix_length prefix-set prefix_set_name
      exit
```

NOTES:

- **strip server-ipv6** : Matches the prefix of server IPv6 address with the configured prefixset and prefix length. If match is found then extracts the IPv4 address from the server IPv6 address.
- *prefix_length*: Enter values 32,40,48,56,64 or 96.
- **prefix-set**: Configures the active configuration for Well-known prefix or Netowrk-specific prefix. You can configure a maximum of 10 IPv6 prefixes in a prefix-set.



CHAPTER 9

Flow Aware Packet Acceleration

This chapter describes the Flow Aware Packet Acceleration (FAPA) feature and provides detailed information on the following topics:

Feature Description

The Flow Aware Packet Acceleration (FAPA) feature improves the throughput in terms of PPS, by caching rule matching results of a flow for selected flows so as not to incur the lookup penalty for a large number of packets in that flow. This new accelerated path is capable of performing a full range of basic functions including handling charging, modification of packet headers, and incrementing various counters. The accelerated path dynamically evaluates the current flow state and reverts back to the slow path when the flow cannot be handled on the fast path.

Important A Flow Aware Packet Acceleration license is required on ASR 5500 and VPC platforms.

The acceleration is applied to specific flows without affecting any external interfaces related to Billing, CLI, interfaces, and so on. This feature is an extension of the TRM/FastPath that was introduced in R15.0 for ASR 5500 platform. This feature will be supported when TRM FastPath is enabled on the Rulebase. TRM FastPath works on approximately 50-65 of all packets, including VoLTE, Encrypted and HTTP, in the system for any given call model, with the control path left intact. New changes are in the data path after TRM has cached the rule matching results. FAPA ECS packet path can efficiently process 50+ of data packets in the system, yielding a significant performance gain on ECS data path.

TRM/FP support has been extended beyond rule-matching. Qualifying packets avoid much of the ECS stack for N bytes of volume for a given flow. Only the packets requiring minimal work are qualified for the accelerated path. The work needed for each packet include a subset of flow actions, QoS enforcement, L3/L4 header inspection, TCP sequence number validation, and applicable charging methods.

The FAPA function identifies packets that need only a small amount of processing, and performs only those necessary tasks on these packets. Only those packets that do not require DPI are allowed to enter the Accelerated path.

VoLTE, encrypted, HTTP, HTTPS, RTP and plain TCP/UDP traffic where L7 analysis is not enabled, and so on are all the flows that will get accelerated.

The FAPA functionality is extended for the active charging service with CF/ICAP/BL configured. Simple TCP traffic will be eligible on accelerated path with Static Content Filtering configured in active charging service but not in rulebase. Packets for tethered flows will be counted on accelerated packets. Packets for blacklisted flows will be counted on fast-path packets.

This feature provides the operator with additional capacity on deployed systems without any hardware addition. The operators could get 30-40 of the system capacity based on their traffic pattern and deployed call models.

The following charging formats are supported on the FAPA feature, which gives improved performance for HTTP traffic, if the traffic flow is FAPA eligible:

- EDR
- EGCDR
- Gy
- Rf
- VoGx

The FAPA feature is controlled by the FAPA license, and a CLI at active-charging service. The FAPA path will be functional, only if TRM/FP is enabled and the CLI is configured.

Configuring Flow Aware Packet Acceleration

Use the following configuration to enable Flow Aware Packet Acceleration (FAPA):

```
configure
  active-charging service <service_name>
    [ no ] accelerate-flow
    end
```

Verifying the FAPA Configuration

Enter the following command in the Exec mode to check the "Accelerate Flow" AVP value:

```
show active-charging service name <service_name>
```

Monitoring and Troubleshooting the FAPA feature

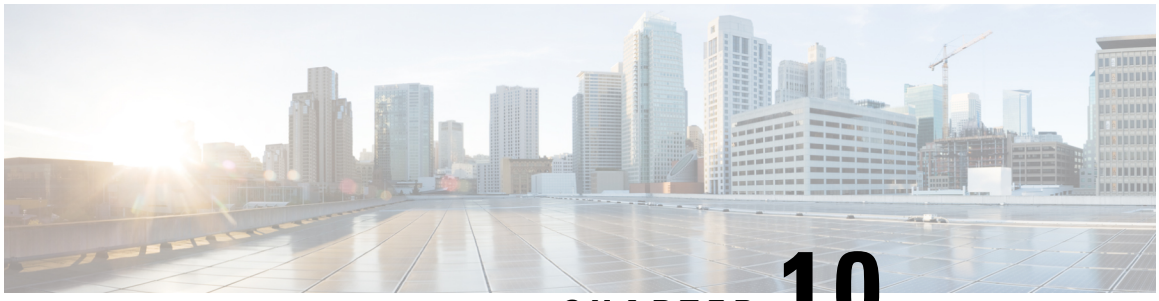
This section provides information regarding show commands and/or their outputs in support of this feature.

Bulk Statistics

The following bulk statistics are supported for the FAPA feature:

- ip-accel-pkts
- udp-accel-pkts
- tcp-accel-pkts

- http-accel-pkts
- https-accel-pkts
- rtp-accel-pkts



CHAPTER 10

Flow Checkpoint Support for ADC Rules

This feature adds Flow Checkpoint support for ADC rules after ICSR/SR switchover.

Feature Information

Summary Data

Status	Modified Functionality
Introduced-In Release	21.1
Modified-In Release(s)	21.2
Applicable Product(s)	GGSN, P-GW, SAEGW
Applicable Platform(s)	ASR 5500
Default Setting	Enabled. This is CLI controlled feature and the same CLI which was used in "Flow Recovery Support for ECS Rules" feature to configure flow-recovery will enable this as well.
Related CDETS ID(s)	CSCvc48853
Related Changes in This Release	Not Applicable
Related Documentation	ECS Administration Guide

Revision History

Important Revision history details are not provided for features introduced before release 21.2.

Revision Details	Release	Release Date
Modified in this release.	21.2	April 27, 2017

Feature Changes

Cisco P-GW supports ICSR/SR check pointing at session level. The Flow Recovery feature was introduced to support flow checkpoint for ECS rules.

This feature adds Flow Checkpoint support for ADC rules after ICSR/SR switchover. Flows of a rule can be recovered if the rule is made eligible. To make any rule eligible for checkpointing or recovery, the rule must be configured in the service scheme framework and identified based on rulename. Any type of rule can be eligible for checkpointing irrespective of the rule definition or protocol type. The rule type can be static/predefined, group-of-ruledef, dynamic, or ADC rule.

When this feature is implemented, the following things are avoided:

- Default rule being matched post recovery or ICSR switchover.
- Flow charging when they are zero rated or not charged.
- Different or incorrect policy being applied to the data.

Behavior Changes

Previous Behavior 1: For configuring multiple rules for flow recovery, a single trigger condition was sufficient as it was possible to add all eligible rules under it along with the delay. If a flow matches any of the configured eligible rules, P-GW started checkpointing the flow after the configured period of delay.

Example: Rule rule01, rule02, and rule03 are eligible for flow recovery. Flow should be checkpointed after 60 sec of flow creation.

```
trigger-condition tc1
rule-name = rule01
rule-name = rule02
rule-name = rule03
delay = 60
#exit
```

The above configuration enabled P-GW to checkpoint flows matching either rule01, rule02, or rule03 after a delay of 60 seconds.

New Behavior 1: With this feature, there is a change in the way flow recovery is configured on the chassis. A new CLI command **multi-line-or all-lines** under trigger condition. Because of this delay cannot be configured along with multiple rules in same trigger condition. If there is a need to configure delay, then you must create multiple trigger condition (one for each rule).

Example with new configuration:

```
trigger-condition tc1
rule-name = rule01
delay = 60
#exit
trigger-condition tc2
rule-name = rule02
delay = 60
#exit
```

```
trigger-condition tc3
rule-name = rule03
delay = 60
#exit
```

The above configuration enables P-GW to checkpoint flows matching either rule01, rule02, or rule03 after a delay of 60 seconds.

Configuration for multiple rules without delay:

```
trigger-condition tc1
rule-name = rule01
rule-name = rule02
rule-name = rule03
multi-line-or all-lines
#exit
```

Previous Behavior 2: ULI was not sent with "successful_resource" allocation CCR-U, if there is no change in the ULI.

New Behavior 2: ULI is sent with "successful_resource" allocation CCR-U, even if there no change in the ULI. Old value of ULI is encoded and sent.

Relationships to Other Features

This sections describes the relationship of the Flow Checkpoint Support for ADC Rules feature with other supported features.

Feature Interaction

The following table describes the impact to inline services with the Flow Checkpoint Support for ADC Rules implementation.

The behavior applies to recovered flows unless explicitly mentioned.

Inline Services	Flow Recovery Impact	Description
CF Static and Dynamic	No	Post recovery, CF redirect and CF blacklist will not work and packets will be allowed.
ICAP	No	Post recovery GET messages for the recovered flow will not be forwarded to ICAP server.
Tethering Detection	No	For IP-TTL based tethering detection, the post recovery flow will be considered as tethered if the first packet post recovery is uplink packet and will be considered as non-tethered.. For OS-DB signature based tethering detection, the post recovery flow will be considered as tethered if the SYN packet received post recovery has matching signature in the database.
Post processing rules	No	URL Redirect for recovered flows will not be supported based on post processing rules. The default action "Discard" will be applied.

Inline Services	Flow Recovery Impact	Description
X-Header insertion	No	X-header insertion will not work for recovered rules.
DSCP/IP-TOS Marking	Yes	Packets will be marked post recovery with correct DSCP/IP-TOS values.
Idle-timeout handling	Yes	The flow is terminated after idle timeout and checkpoint limits are decremented.
TCP based application Protocol	No	Analyzer will not be engaged for recovered flows.
UDP Based application Protocol	Yes	Analyzer will be engaged for recovered flows if configured in rulebase.
ICSR	Yes	Flows will be recovered to match the last match rule pre ICSR and for which checkpoint was successful.
Session Recovery	Yes	Flows will be recovered to match the last match rule pre SR and for which checkpoint was successful.
IPv6	Yes	IPv6 flows will be recovered.

Charging Methods

The following table describes the impact to ECS charging methods with the flow checkpoint support for the ADC Rules feature implementation.

The behavior applies to recovered flows unless explicitly mentioned.

Charging Method	Flow Recovery Impact
eGCDR	No
PGW-CDR	No
VoGx	No
Gy	No
RF	No
EDR	For P2P we will not recover field P2P app-identifier(SNI) and P2P duration
FDR	No
UDR	No

How It Works

This section describes the working of this feature.

- Flows of a rule is eligible for recovery if the rule is made eligible for flow recovery under policy framework service scheme.
- The number of flows checkpointed/recovered is limited at per subscriber call level as per the configuration at active-charging service level.
- The number of flows checkpointed/recovered is limited at per sessmgr instance level which will not be a configurable value.
- The eligible flow(s) is/are checkpointed in the following cases:
  - Post flow establishment, the delay timer configured in the policy framework service scheme for matching rule expires.
  - Limit for number of recovered flows is not reached
- Any flow which has been recovered will continue matching the last matched checkpointed rule irrespective of a higher precedence rule available later. If the last matched checkpointed rule for a flow is removed, will match to rules based on L3/L4 definition.
- Any flow for which delay timer has not expired and which was not checkpointed, will match the rules based on L3/L4 definition after recovery.

Sample Configuration

This section lists sample configuration for configuring P2P rules.

In order to make a P2P rule eligible for checkpoint, the rule needs to be configured in the policy framework service scheme under trigger-condition and then associated with the flow-recovery trigger-action.

Configuring P2P Rule for the Youtube

```
configure
  active-charging service service_1
    ruledef rule_youtube
       p2p protocol = youtube
       p2p traffic-type = file-transfer
    #exit
    rulebase base_1
       action priority 1 ruledef rule_youtube charging-action ca_edr
end
```

Configuring P2P Rule Youtube for Flow Recovery

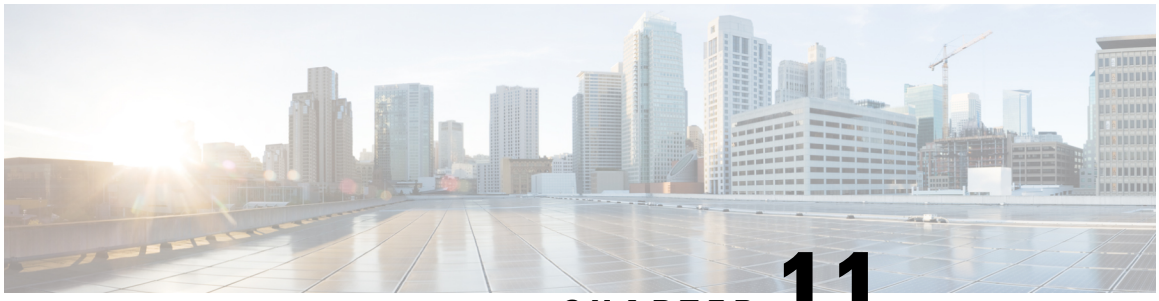
```
active-charging service ACS
    trigger-action action1
      flow-recovery
    #exit
    trigger-condition tc1
      rule-name = rule_youtube
```

```
      delay = 600
    #exit
    service-scheme scheme1
      trigger flow-create
      priority 1 trigger-condition tc1 trigger-action action1
    #exit
  #exit
end
```

Limitations

Due to memory and performance impact only selected P2P field will be recovered for EDR, that is, P2P Protocol, P2P Protocol Group, and P2P Traffic Type.

Important All the limitations mentioned in the "Flow Recovery" feature are applicable for this feature as well.



CHAPTER 11

Flow Recovery Support for ECS Rules

This chapter describes the Flow Recovery feature and provides detailed information on the following topics:

Feature Description

The Flow Recovery feature is introduced to support flow checkpoint for ECS rules. Flows of a rule can be recovered if the rule is made eligible. To make any rule eligible for checkpointing or recovery, the rule must be configured in the service scheme framework and identified based on rulename. Any type of rule can be eligible for checkpointing irrespective of the rule definition or protocol type. The rule type can be static/predefined, group-of-ruledef, dynamic, or ADC rule.

Important Flow Recovery is a licensed Cisco feature requiring a separate feature license. Contact your Cisco account representative for more information.

With the previous implementation, Cisco PCEF supports ICSR/SR check pointing at session level. Session and bearers were recovered as part of session recovery or ICSR switchover but flows were not recovered. Due to this when the packets were received on already established flows after the recovery, the packets were matched to default rules based on L3/L4 definition. This causes billing impact as flows were incorrectly charged.

Important If the flow recovery/checkpointing feature is enabled, then ICSR control outage might have a significant impact.

The enhancements with the Flow Recovery feature are listed below.

- **List of eligible rules for flow recovery:** A configurable trigger action CLI **flow recovery** is added to configure flow-recovery under service scheme. The flows matching the configured rules in service scheme will be recovered. The flows will be checkpointed to AAAMgr when the delay timer for the flow matching these rules expires post the flow establishment.

- **Limit configuration for the number of flows recovered:** The maximum number of flows to be checkpointed or recovered will be limited per subscriber using the **system-limit flow-chkpt-per-call** CLI at active-charging service level. This limit controls the number of flows that can be recovered across rules and bearers for a single call.

  The number of flows checkpointed/recovered will be limited at per SessMgr instance level. This will be a non-configurable value.

- **Rule matching post recovery:** If a flow is recovered to match a checkpointed rule, the incoming packets will continue matching the same rule throughout the lifetime of the flow and the flow will remain checkpointed with the rule. The checkpoint information will be deleted post recovery when:
  - The rule from recovered list is uninstalled for the session
  - The flow idle time-out happens
  - The TCP-based application flow is gracefully terminated when a RST packet is received at the gateway
  - The TCP-based application flow is not terminated when FIN packet is received at the gateway and cleared after the flow idle timeout happens

- **Handoff/ICSR scenario:** After a rule is recovered for control events such that the bearer/rule information is not available (due to delay in EGTP message), the packets for these flows will match the default rule based on L3/L4 definition until transactions are complete. Once the control events are no longer pending, the packets will start matching the last matched rule from flow recovery list.

  Similarly for ICSR, the packets for recovered flows will match the default rules based on L3/L4 definition. Once the information is received from new chassis, the packets will start matching the last matched rule from flow recovery list.

- **KPI and Bulk Statistics:** KPI and Bulk Statistics are added on a per rulename basis for the rules that are eligible for checkpointing. When a rule is configured as eligible for recovery, KPI and bulk statistics for the rule will be displayed. For a rule that is deleted from the service-scheme framework, KPI and bulk statistics will not be displayed. SRP level statistics are also added to maintain the SRP bandwidth required for flow checkpoint.

Relationships to Other Features

This sections describes the relationship of the Flow Recovery feature with other supported features.

Feature Interaction

The following table describes the impact to inline services with the flow recovery implementation.

The behavior applies to recovered flows unless explicitly mentioned.

Inline Services	Flow Recovery Impact	Description
CF Static and Dynamic	No	CF redirect and CF blacklist post recovery will not work and packets will be allowed.
ICAP	No	Post recovery GET messages for the recovered flow will not be forwarded to ICAP server.

Inline Services	Flow Recovery Impact	Description
Tethering Detection	No	For IP-TTL based tethering detection, the post recovery flow will be considered as tethered if the first packet post recovery is uplink packet. For OS-DB signature based tethering detection, the post recovery flow will be considered as tethered if the SYN packet received post recovery has matching signature in the database.
P2P	No	P2P rules are not supported for recovery.
Post processing rules	No	URL Redirect for recovered flows will not be supported based on post processing rules. The default action "Discard" will be applied.
X-Header insertion	No	X-header insertion will not work for recovered rules.
DSCP/IP-TOS Marking	Yes	Packets will be marked post recovery with correct DSCP/IP-TOS values.
Idle-timeout handling	Yes	The flow is terminated after idle timeout and checkpoint limits are decremented.
TCP based application Protocol	No	Analyzer will not be engaged for recovered flows.
UDP Based application Protocol	Yes	Analyzer will be engaged for recovered flows if configured in rulebase.
ICSR	Yes	Flows will be recovered to match the last match rule pre ICSR and for which checkpoint was successful.
Session Recovery	Yes	Flows will be recovered to match the last match rule pre SR and for which checkpoint was successful.
IPv6	Yes	IPv6 flows will be recovered.

Flow Actions

The following table describes the behavior of flow actions with the flow recovery implementation.

This applies to recovered flows unless explicitly mentioned.

Flow Action	Description
Rate Limiting	Per bearer rate limiting, per rule rate limiting, ITC rate limiting, or QG related rate limiting is applied.

Flow Action	Description
Allow/Deny packet	Flow-action discard OR dynamic-rule flow-status enforcement is applied.
URL redirect	Run time URL redirection is not applied for a recovered flow. If the packets of a flow are redirected before recovery and the matching rule is eligible and checkpointed, packets are redirected post recovery as well.
Flow Readdress	Since the flow readdress is identified at SYN packet and SYN packets are not analyzed post recovery, flow readdress will not work post recovery.
Flow Terminate	Flow terminate action is applied for dynamic changes to charging action configuration.
Next Hop	Packets are forwarded to the next hop address.

Charging Methods

The following table describes the impact to ECS charging methods with the flow recovery implementation.

The behavior applies to recovered flows unless explicitly mentioned.

Charging Method	Flow Recovery Impact
eGCDR	No
PGW-CDR	No
VoGx	No
Gy	No
RF	No
EDR	In case of recovery, generated EDR will show "Unknown" for sn-direction because the direction of first packet is not known.
FDR	No
UDR	No

Limitations

This section lists the limitations associated with the Flow Recovery/Checkpoint feature.

- Few calls may be lost if session recovery occurs when the system is experiencing very high flow setup rates.
- Checkpoint failures (failure counters being incremented) might be seen when large number of flows are being check pointed (>12K per seconds). The failed checkpoints are retried again. If a session recovery, unplanned card migration, or unplanned switchover occurs while the checkpoints are being retried, call loss may be seen.
- Disabling of flow recovery feature (via CLI) while active calls (using flow recovery) are present might result in session manager restart.

- Rule Matching – Packets are not matched to L7 rule after bearer movement from dedicated bearer to default bearer during LTE to WiFi HO. The same behavior will be seen post recovery.
- Static rule match on dedicated bearer – After recovery, flow packets will always match the last matched rule. If the TFT are installed on dedicated bearer post recovery, such that the last match checkpointed rule is a static rule but the packets are received on dedicated bearer due to matching TFT, the packets are still matched to static rule but on dedicated bearer. Charging and QoS parameters are applied as per dedicated bearer.
- EDR loss for SR/Unplanned ICSR – EDR is not generated in case of session recovery or unplanned ICSR.
- Parent-child relation loss post recovery – The parent-child relationship post recovery is not maintained for protocols such as RTP and RTSP.
- Lifetime of recovered flows – Lifetime will not be incremented for recovered flows for which no packet is received post recovery and is deleted due to flow idle timeout.
- Service scheme framework – Flow checkpointing is dependent on the service scheme framework. Since this framework is introduced only in release 20.2, flow checkpointing for the existing calls/flows is supported only for the 20.2 build during upgrade scenarios. In all other upgrade scenarios, flow checkpointing will not be supported for the existing calls/flows.

How It Works

This section describes the Flow Recovery configuration. The Service Scheme framework configuration is required to configure and enable this feature for a subscriber.

Use the sample configuration to enable the Flow Recovery feature.

```
configure
  active-charging service s1
    trigger-action ta1
      flow-recovery
    #exit
    trigger-condition tc1
      rule-name = rule01
      delay = 600
    #exit
    service-scheme ss1
        trigger flow-create
          priority 1 trigger-condition tc1  trigger-action ta1
      #exit
    #exit
  end
```

Restrictions

Any TCP-based application protocol sends out mid-flow acknowledgements that are TCP packets. These packets are generally matched to a L3/L4 based rule on that bearer. In such cases a delete checkpoint will be sent whenever the L3/L4 rule is processed for the flow and again a checkpoint to AAAMgr is sent when the L7 rule is processed for the flow. This could result in high number of checkpoints sent/received and delete

checkpoints sent/received. To avoid this scenario, delay charging must be configured for all packets or mid-flow packets in conjunction with flow-recovery for the rule under rulebase using the **flow control-handshaking charge-to-application { all-packets | mid-session-packets }** CLI configuration.

Configuring Flow Recovery Checkpointing

The rules eligible for checkpointing must be configured in the service-scheme framework and identified based on rulename. The rule type can be static/predefined, group-of-ruledef, dynamic or ADC rule. A group-of-ruledef can be independently made eligible for checkpointing even if the constituent ruledefs are not eligible for checkpoint. If both group-of-ruledef and constituent ruledefs are eligible for checkpoint, either one can be checkpointed based on the higher action priority configured in rulebase.

A delay timer to delay the checkpoint of a flow can be configured under the service-scheme trigger condition. This configuration controls the checkpoint sent/received (both checkpointing and deleting checkpoint) for flows which may last for a shorter duration. A new trigger action type "flow recovery" is also added to the service-scheme configuration.

Configuring Flow Recovery

Use the following configuration in the ACS Trigger Action Configuration mode to enable flow recovery for a trigger-action.

```
configure
   active-charging service <service_name>
      trigger-action <trigger_action_name>
         [ no ] flow-recovery
         exit
```

Notes:

- The flows for the rule will be checkpointed as per session level and call level limit.
- To disable flow recovery, configure the following command:

  no flow-recovery

Configuring Delay Checkpointing for flow

Use the following configuration in the ACS Trigger Condition Configuration mode to define the list of rules along with the delay after which the flows can be checkpointed. When configured in conjunction with flow-recovery trigger, the flows for the rule(s) will be checkpointed as per session level and call level limit after the delay timer is expired.

```
configure
   active-charging service <service_name>
      trigger-condition <trigger_condn_name>
         delay = <delay_time>
         rule-name operator <rule_name>
         exit
```

Notes:

- The range of the delay timer to be configured is 1 through 600 seconds. If the "delay" CLI command is not configured under trigger-condition, any flow for the rule will be checkpointed immediately on flow creation.
- **rule-name** *operator <rule_name>*
  - *operator* specifies to match and must be one of the following:
    - =
    - *contains*
    - *ends-with*
    - *starts-with*

    The operators "contains", "starts-with", and "ends-with" cannot be used with dynamic rule names. For dynamic rules, the entire rulename must be mentioned with the "=" operator.
  - *rule_name* must be an alphanumeric string of 1 through 63 characters.
- In any defined trigger-condition, a user can configure upto a maximum of 15 entries.
- To have more rules eligible for flow checkpoint, a user can configure multiple trigger condition(s) associated with the same trigger-action.
- Use the **no rule-name** command to remove the particular rule from the list of eligible rules for flow checkpoint. For wildcard-based rule definition, this command must contain rulename in the same format.
- Use the **no delay** command to checkpoint all eligible rules immediately without any delay.

Enabling trigger for new flows

Use the following configuration in the ACS Service Scheme Configuration mode to enable trigger for every new flow.

```
configure
  active-charging service <service_name>
    service-scheme <serv_scheme_name>
      [ no ] trigger flow-create
      exit
```

Configuring Flow Checkpoint Limit

Use the following configuration in the ACS Configuration mode to control the number of flows that can be checkpointed per call.

```
configure
  active-charging service <service_name>
    system-limit flow-chkpt-per-call <num_flows>
    exit
```

Notes:

- The number of flows that will be allowed to checkpoint will be maintained at call-ID level. The number of flows that can be checkpointed per call are between 1 to 300.
- The default value of the number of flows checkpointed is 10. This value will apply for all the calls across rules and bearers.

Configuring Flow KPI Bulk Statistics

A new bulk statistic schema **flow-kpi** is added to maintain the KPI for flows. These counters will be the sum of the statistics for the corresponding rule(s) across session manager instance.

Important The Flow Recovery feature license is required to display KPI and configure bulk statistics.

Use the following configuration to configure the Flow KPI bulk statistic schema.

```
configure
  bulkstat mode
     file file_number
     flow-kpi schema schema_name format schema_format
     exit
```

Use the following configuration to view the Flow KPI bulk statistics in the Flow KPI schema.

```
show bulkstats variables flow-kpi
```

Verifying the Flow Recovery Configuration

Enter the following command to check the cumulative KPI for each rule across session managers:

```
show active-charging flow-kpi [ all | instance instance_id ]
```

Monitoring and Troubleshooting the Flow Recovery Feature

The sections listed below describe the KPI, Bulk statistics, and SRP level checkpointing statistics that can be used to identify if the checkpoints are sent and successful.

Flow Recovery Show Command(s) and/or Outputs

This section provides information regarding show commands and/or their outputs in support of the Flow Recovery feature.

show active-charging flows all

The output of this command is modified to include a new flag to identify if the flow is a recovered flow or not. The value will display Y for a recovered flow and N for a non-recovered flow. The flag will be displayed if flow recovery license is present.

- Recovered Flow: (Y) - Yes (N) - No

show active-charging flow-kpi all

In support of the Flow Recovery feature, the **show active-charging flow-kpi all** CLI command is added to display the cumulative KPI for each rule across session managers and can also filter the statistics for all rules based on sessmgr instance. KPI are added on a per rule basis.

- Rule Name
- Active Flows
- SR Flow Checkpoint Sent
- SR Flow Checkpoint Received
- GR Flow Checkpoint Sent
- GR Flow Checkpoint Received
- SR Flow Checkpoint Delete Sent
- SR Flow Checkpoint Delete Received
- GR Flow Checkpoint Delete Sent
- GR Flow Checkpoint Delete Received
- Flows of lifetime bucket1
- Flows of lifetime bucket2
- Flows of lifetime bucket3

show active-charging trigger-action all

The following field indicates whether flow recovery is enabled or disabled.

- Flow recovery

show active-charging trigger-condition all

The following fields are added to the output of this command in support of this feature:

- Trigger Action Delay - Displays the delay (in seconds) for application of action.
- Rule-name/GOR - Displays the condition specified for a particular rule/GoR for flow checkpoint.

show srp checkpoint statistics

The SRP checkpoint statistics are added as part of checkpoint and recovery for ECS and ADC flows. Statistics for any flow checkpoint sent and received as well as delete checkpoint sent and received will be maintained.

ACS_FLOW_INFO and DEL_ACS_FLOW_INFO fields are displayed only after the creation/deletion of flow checkpoints respectively. When the flow checkpoints are sent from active chassis to standby chassis, the corresponding ACS_FLOW_INFO micro checkpoint statistics will be incremented. Similarly, after the expiry of idle timeout configured in the active-charging service, the flows will be deleted and the corresponding DEL_ACS_FLOW_INFO micro checkpoint statistics will be incremented.

A sample output is shown below. The rate limited MC checkpoints are displayed under "Checkpoints Rate limited" statistics and will be pegged only for ACS_FLOW_INFO and DEL_ACS_FLOW_INFO micro checkpoints.

```
show srp checkpoint statistics [OR] show srp checkpoint statistics sessmgr all
SRP Session Checkpoint Sent Statistics
======================================
Checkpoints Rate limited-------------------------------------------------------+
Bytes sent---------------------------------------------------------------------+  |
External_audit_in_progress-----------------------------------------+         |  |
Recovery in progress-----------------------------------------------+  |      |  |
Write list congested-----------------------------------------------+  |  |   |  |
Invalid GR state---------------------------------------------------+  |  |  |   |  |
Session not ready for recovery-------------------------+  |  |  |  |   |  |
Encoding failure ignored-------------------------------+  |  |  |  |  |   |  |
Encoding failure---------------------------------+  |  |  |  |  |  |  |   |  |
Ignored as Invalidate CRR sent-----------------+  |  |  |  |  |  |  |  |   |  |
Ignored as first FC never sent-------------+  |  |  |  |  |  |  |  |  |   |  |
Checkpoint rate(last 30 min)-------------+  |  |  |  |  |  |  |  |  |  |   |  |
Checkpoint rate(now)-----------------+  |  |  |  |  |  |  |  |  |  |  |   |  |
Encoding attempted---------------+   |  |  |  |  |  |  |  |  |  |  |  |   |  |
Messages sent--------------+     |   |  |  |  |  |  |  |  |  |  |  |  |   |  |
Checkpoint-type            |     |   |  |  |  |  |  |  |  |  |  |  |  |   |  |

     UPDATE_CLPSTATS       0     0   0  0  0  0  0  0  0  0  0  0  3        0  0
     ACS_SESS_INFO      2060  2060   0  0  0  0  0  0  0  0  0  0  0   298130  0
     ACS_CALL_INFO      2060  2060   0  0  0  0  0  0  0  0  0  0  0   306240  0
     PGW_UBR_MBR_INFO   1000  1000   0  0  0  0  0  0  0  0  0  0  0   457000  0
     ACS_FLOW_INFO        30    50   0  0  0  0  0  0  0  0  0  0  0     8490 20
     DEL_ACS_FLOW_INFO    30    50   0  0  0  0  0  0  0  0  0  0  0     6060 20
     Full Checkpoint    1000  1000   0  0  0  0  0  0  0  0  0  0  0  4754470  0
=============================================================================================================
```

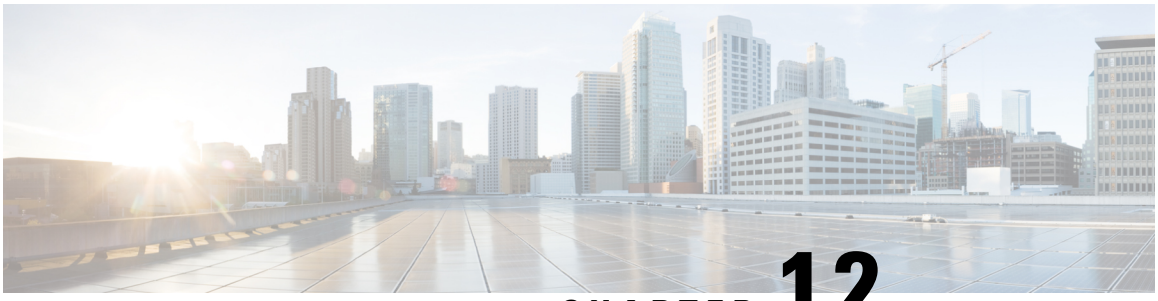
Flow Recovery Bulk Statistics

A new **Flow KPI** schema is added in support of the Flow Recovery feature.

The following bulk statistics are added in the Flow KPI Schema:

- ecs-flow-rule-name – The name of rules eligible for flow checkpointing.
- ecs-num-active-flow – Total number of active flows of the rule.
- ecs-flow-chkpt-sent-sr – Total number of SR flow checkpoint sent for the rule.
- ecs-flow-chkpt-recvd-sr – Total number of SR flow checkpoint received for the rule.
- ecs-flow-chkpt-sent-gr – Total number of GR flow checkpoint sent for the rule.
- ecs-flow-chkpt-recvd-gr – Total number of GR flow checkpoint received for the rule.
- ecs-del-flow-chkpt-sent-sr – Total number of SR delete flow checkpoint sent for the rule.
- ecs-del-flow-chkpt-recvd-sr – Total number of SR delete flow checkpoint received for the rule.
- ecs-del-flow-chkpt-sent-gr – Total number of GR delete flow checkpoint sent for the rule.
- ecs-del-flow-chkpt-recvd-gr – Total number of GR delete flow checkpoint received for the rule.
- ecs-flow-lifetime-bucket1 – Total number of flows of lifetime_bucket1. Lifetime value of bucket1 is configurable.

- ecs-flow-lifetime-bucket2 – Total number of flows of lifetime_bucket2. Lifetime value of bucket2 is configurable.
- ecs-flow-lifetime-bucket3 – Total number of flows of lifetime_bucket3. Lifetime value of bucket3 is configurable.



CHAPTER 12

GTPC Peer Record and Statistic Optimization

This chapter describes the following topics:

Feature Summary and Revision History

Summary Data

Applicable Product(s) or Functional Area	• P-GW • SAEGW • S-GW
Applicable Platform(s)	• ASR 5500 • VPC-SI • VPC-DI
Feature Default	Disabled - Configuration Required
Related Changes in This Release	Not Applicable
Related Documentation	• *Command Line Interface Reference* • *ECS Administration Guide*

Revision History

Revision Details	Release
First introduced.	21.4.2

Feature Description

When the Gateway receives the first GTPC message from a peer, the new peer record entry is added to the Session Manager and Demux. This new peer record entry is also propagated to all Session Managers. This occurs even if a particular GTPC peer does not have any active sessions. This causes accumulation of inactive peer records objects, which results in excess memory usage of the Session Manager and Demux, thereby causing memory overrun of affected proclets. To address this limitation, a new keyword, **peer-salvation** has been added to the existing **gtpc** CLI in the Context Configuration mode.

When enabled, this keyword supports the following behavior:

1. When a peer goes inactive with zero number of active sessions, the timestamp is stored at that peer record object and a peer record is inserted into the inactive peer list.
2. If any new session gets added to inactive peer, the timestamp is reset to zero and the peer record entry is removed from the inactive peer list to avoid salvation of the reactivated peer.
3. A one-hour timeout is set per egtpmgr instance level that gets disabled when the keyword is disabled at the context level.
4. Separate salvation timers run for egtpinmgr and egtpegmgr.
5. By default, (when keyword is not enabled) salvation timer does not run to minimize the memory and CPU impact.

Important

- When the **peer-salvation** keyword is enabled at the context level, but not enabled at egtp-service level, then peer salvation does not occur.
- 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.
- The **min-peers** value should be applied judiciously to ensure that the Session Manager in a fully loaded chassis does not go into warn/over state with many peer records. If the Session Manager goes into a warn/over state, then it is recommended to configure a lesser value for **min-peers** to ensure that the peers are salvaged.
- **min-peers** configuration is not considered during a new peer creation.
- Only peers with zero number of sessions are salvaged for the configured timeout value. Non-zero number of sessions is not salvaged even if there are few.

Demux Session Recover Scenario

When Demux proclet crashes or restarts, all the information related to all the inactive peers is cleared in the proclet and is not added again during the session recovery of Demux. These inactive peer records accumulated on the Demux-serving Session Managers might not get salvaged. The peer salvation functionality reconstructs the inactive peer list a the recovered Demux. The last activity timeout for the inactive peers is set to the timestamp of Demux recovery, thereby, allowing Demux to work even after Demux recovery.

Demux Inter-Chassis Session Recovery Scenario

The **peer-salvation** keyword can be configured on the Active and Standby chassis. When configured, it can even salvage the inactive peers accumulated on the Standby chassis.

Session Manager Session Recovery/ICSR Scenario

Configuring the **peer-salvation** keyword does not impact the Session Manager recovery or ICSR and vice-versa.

Configuring the Peer Salvation Functionality

The following section provides the configuration commands to enable or disable the feature.

gtpc peer-salvation (context configuration mode)

Use this command to enable peer salvation for inactive GTPv2 peers for EGTP services in this context. The **peer-salvation** keyword is introduced in the Context Configuration mode. Minimum peers and timeout values can be provided with this CLI, which is per egtpmgr (separate for egtpinmgr and egtpegmgr) and across all the egtp-services configured in that context.

To configure **peer-salvation** in the Context Configuration mode, enter the following commands:

```
configure
  context context_name
    [ no ] gtpc peer-salvation min-peers value timeout value
    end
```

NOTES:

- **no**: Disables peer salvation at the context level.
- **peer-salvation**: Enables peer salvation for inactive GTPv2 peers for EGTP services in this context.
- **min-peers** *value*: Configures the minimum number of accumulated GTPv2 peers across all EGTP services to start salvaging the inactive peers. The value ranges from 2000 to 12000.
- **timeout** *value*: Configures the peer salvation timeout. The peer that is inactive for salvation time is salvaged, specified in hours. The value ranges from 1 to 48 hours.
- This command is disabled by default.

gtpc peer-salvation (eGTP service configuration mode)

Use this command to enable peer salvation for inactive GTPv2 peers for EGTP services in this context. The **peer-salvation** keyword is added to the existing **gtpc** command in eGTP Service Configuration mode.

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.

To configure **peer-salvation** in the eGTP Service Configuration mode, enter the following commands:

```
configure
  context context_name
    egtp-service egtp_service_name
    [ no ] gtpc peer-salvation
    end
```

NOTES:

- **no**: Disables peer salvation at the context level.
- **peer-salvation**: Enables peer salvation for inactive GTPv2 peers for EGTP services in this context.
- This command is disabled by default.

Monitoring and Troubleshooting

This section provides information regarding show commands and/or their outputs in support of this feature.

Show Commands and/or Outputs

The output of the following CLI command has been enhanced in support of the feature.

show egtp-service all

With 21.4.2 and later releases, the output of this command has been enhanced to include the following new field in support of the Peer Salvation functionality:

GTPC Peer Salvation

show session subsystem debug-info

With 21.4.2 and later releases, the output of this command has been enhanced to include the following new fields in support of the Peer Salvation functionality:

- Peer Salvation Stats
  - No of peer salvation requests received on sessmgr
  - No of peer salvaged on sessmgr

show demux-mgr statistics egtpinmgr all

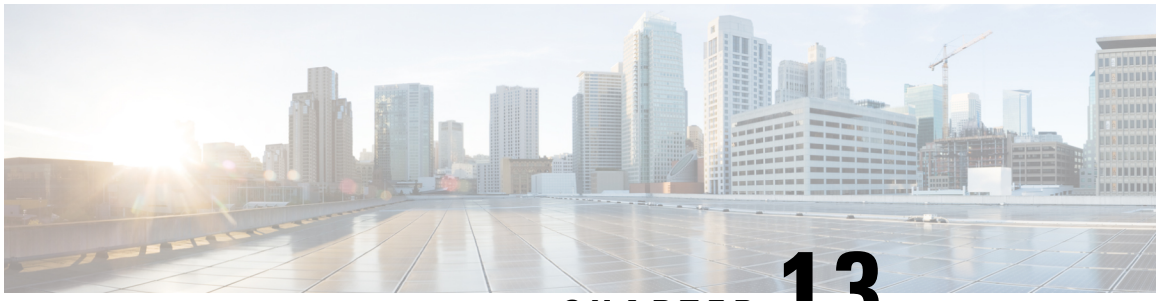
With 21.4.2 and later releases, the output of this command has been enhanced to include the following new fields in support of the Peer Salvation functionality:

- Peer Salvation Stats
  - No of peer salvation requests sent by demux
  - No of peer salvaged on demux

show demux-mgr statistics egtpegmgr all

With 21.4.2 and later releases, the output of this command has been enhanced to include the following new fields in support of the Peer Salvation functionality:

- Peer Salvation Stats
  - No of peer salvation requests sent by demux
  - No of peer salvaged on demux



CHAPTER 13

Handling Flow-Information AVPs

This chapter describes the following topics:

Feature Summary and Revision History

Summary Data

Applicable Product(s) or Functional Area	• GGSN • P-GW • SAEGW
Applicable Platform(s)	ASR 5500
Feature Default	Enabled - Always-on
Related Changes in This Release	Not Applicable
Related Documentation	*ECS Administration Guide*

Revision History

Important Revision history details are not provided for features introduced before releases 21.2 and N5.5.

Revision Details	Release
ECS supports up to 16 Flow-Information AVPs and remaining, if any, are ignored.	21.4
First introduced.	Pre 21.2

Feature Description

PCRF can send any number of Flow-Information AVPs in CC-Answer (CCA) and Re-Auth-Request (RAR) messages. Currently, the ECS supports processing of eight Flow-Information AVPs and remaining, if any, are ignored. With this enhancement, ECS supports up to 16 Flow-Information AVPs and remaining, if any, are ignored.

How It Works

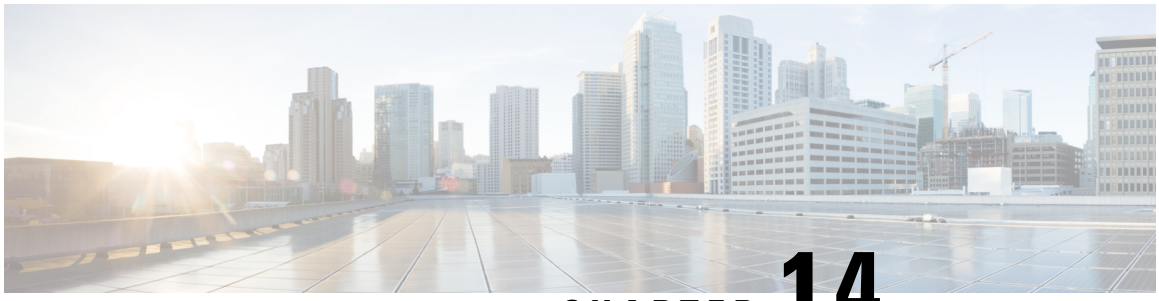
This section provides a brief overview of how this feature works.

- P-GW initiates Create Bearer Request/Update Bearer Request with TFTs for all 16 Flow-Information AVPs.
- P-GW initiates multiple Create Bearer Request/Update Bearer Request if TFTs for all 16 Flow-Information AVPs cannot be accommodated in a single message.
- Rule match for dynamic rule to consider up to 16 Flow-Information AVPs for traffic classification.
- If more than 16 Flow-Information AVPs are received, only 16 AVPs are processed and remaining AVPs are ignored.
- P-GW checkpoints all 16 Flow-Information AVPs received from PCRF in CCA and RAR messages to standby chassis if standby chassis is on same release.
- P-GW recovers all 16 Flow-Information AVPs after Session Manager recovery.
- CRR/checkpoint size does not increase if P-GW receives less than 8 Flow-Information AVPs.
- P-GW checkpoints up to 8 Flow-Information AVPs to standby chassis if standby chassis is on older release, and if it receives more than 8 Flow-Information AVPs from PCRF in CCA and RAR messages.
- ICSR upgrade from N-1 to N, and ICSR downgrade from N to N-1 release works as expected.

Monitoring and Troubleshooting

The output of the following CLI commands has been modified to display 16 Flow-Information AVPs:

- **show active-charging sessions full all**
- **show subscribers pgw-only full all**



CHAPTER 14

HTTP URL Percent Encoding

This chapter describes the HTTP URL Percent Encoding feature in ECS, and provides detailed information on the following topics:

Feature Description

Percent-encoding functionality is used whenever URL redirection happens for an HTTP Request packet and the requested URL (original URL) is embedded in redirected response. The reserved characters present in the requested URL are replaced by % followed by ASCII hexadecimal value. As defined in RFC 3986, a capability is added in the ECS to percent-encode reserved characters when modifying HTTP URLs. The percent (%) character is also percent-encoded as “%25” as this serves as the indicator for percent-encoded octets within a URL.

HTTP URL redirection can happen via Charging Action or DCCA. For redirection via DCCA, percent encoding is needed only when original URL is appended. URL append is supported only for **dcca-custom1** and **dcca-custom24** dictionaries. When original URL is embedded in the redirected URL string for HTTP URL redirection, all reserved characters in the original URL are now percent-encoded when redirection is triggered via charging action or DCCA.

In case of HTTP URL redirection, when original URL is embedded in the redirected URL string:

- All reserved characters in the original URL (per RFC 3986) are percent-encoded in case redirection is triggered via charging-action.
- All reserved characters in the original URL (per RFC 3986) are percent-encoded in case redirection is triggered via DCCA.
- The % character, if present in the original URL is percent-encoded to %25.
- In case the original URL was concatenated with other information, then all reserved characters and % character are percent-encoded if redirection is triggered via charging-action.

The following table lists the reserved characters used for percent encoding and supported as per RFC 3986.

Table 6: Reserved Characters per RFC 3986

Reserved Character	ASCII Value	Percent Encode Format (Hex)	Support for Charging-action based Redirect	Support for DCCA based Redirect
:	58	%3A	Yes	Yes
/	47	%2F	Yes	Yes
?	63	%3F	Yes	Yes
#	35	%23	Yes	Yes
=	61	%3D	Yes	No
&	38	%26	Yes	No
+	43	%2B	Yes	No
@	64	%40	No	No
!	33	%21	No	No
$	36	%24	No	No
'	46	%2E	No	No
(	40	%28	No	No
)	41	%29	No	No
*	42	%2A	No	No
,	44	%2C	No	No
;	59	%3B	No	No
[	91	%5B	No	No
]	93	%5D	No	No

How it Works

Percent Encoding of HTTP URL

Consider the following sample configuration used for URL redirection. The directive **#HTTP.URL#** will embed the original requested URL into the redirected URL string and the embedded URL will be percent-encoded.

```
flow action redirect-url http://www.cisco.com=#HTTP.URL#
```

Where the value of original URL (HTTP.URL) is as follows:

HTTP.URL => mailcisco:/?#=&+@!$'()*,;[]

The percent encoded URL will be:

mailcisco%3A%2F%3F%23%3D%26%2B%40%21%24%2E%28%29%2A%2C%3B%5B%5D

The complete redirected URL in HTTP redirection response will be:

http://www.cisco.com=mailcisco%3A%2F%3F%23%3D%26%2B%40%21%24%2E%28%29%2A%2C%3B%5B%5D

DCCA Dictionaries

For URL redirection via DCCA, percent encoding is required only when original URL is appended. This URL append is supported for **dcca-custom1** and **dcca-custom24** customer-specific dictionaries.

URL Append Message from OCS for dcca-custom1

For **dcca-custom1** dictionary, the URL-Append flag must be set so that the requested URL will be embedded in the redirect response.

```
[M] Final-Unit-Action: REDIRECT (1)
   [M] Redirect-Server:
      [M] Redirect-Address-Type: URL (2)
      [M] Redirect-Server-Address: <redirect-url-address>
   [V] [M] URL-Append: APPEND_URL (1)
```

URL Append Message from OCS for dcca-custom24

For **dcca-custom24** dictionary, although URL-Append flag is not set, the ? character at the end of **redirect-url-address** string indicates that the requested URL will be embedded in the redirect response.

```
[M] Final-Unit-Action: REDIRECT (1)
   [M] Redirect-Server:
      [M] Redirect-Address-Type: URL (2)
      [M] Redirect-Server-Address: <redirect-url-address>?
   [V] [M] URL-Append: DO_NOT_APPEND_URL (0)
```

Standards Compliance

The HTTP URL Encoding feature complies with RFC 3986 — Uniform Resource Identifier (URI): Generic Syntax.

Configuring URL Redirection via Charging Action

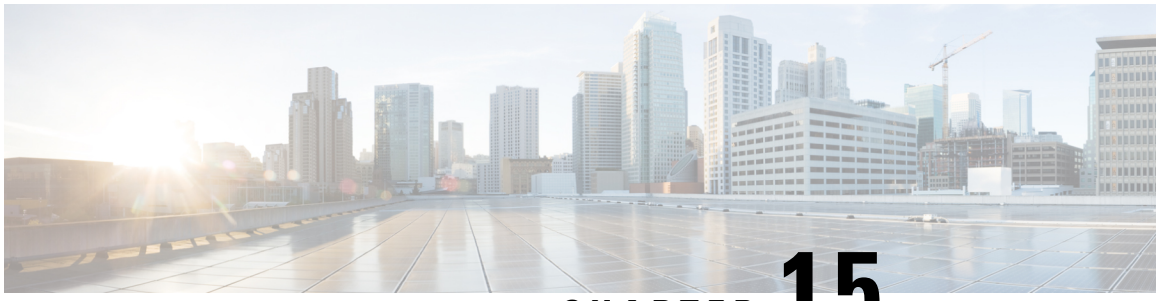
The following configuration command is used to configure URL redirection via charging action.

```
config
  active-charging service service_name
    charging-action charging_action_name
      flow action redirect url url_string
      exit
```

Notes:

url_string specifies the redirect URL and must be an alphanumeric string of 1 through 511 characters. It may include one or more dynamic fields (up to 16 may be specified). Dynamic fields must be enclosed in "#" (hash).

For example: **http://www.cisco.com=#HTTP.URL#**



CHAPTER 15

L7 Dynamic Rule Activation

This chapter describes the L7 Dynamic Rule Activation feature and provides detailed information on the following topics:

Feature Description

Currently gateway supports PCC dynamic rules with L3/L4 filters through the Flow-Description AVP. This feature provides finer control over the filters with L7 support. This feature is implemented in such a way that PCEF/PCRF is able to fully support L7 dynamic rules and thereby enabling dynamic routes to redirect L7 traffic.

Important This feature requires a valid license to be installed prior to configuring this feature. Contact your Cisco account representative for more information on the licensing requirements.

When Out-of-Credit (OOC) trigger is sent from OCS to PCRF, an L7 dynamic rule is sent from PCRF along with a condition and action which allow the subscriber to access specific URLs. The condition is the trigger when to apply the action. For example: If OOC (quota exhaustion condition) is sent from OCS, PCEF should allow (action) all the packets matching that rule (rating-group) to pass through. Once the relocation of credit occurs the gateway reverts back the special treatment for these URLs.

The gateway supports L7 dynamic rule installation through Charging-Rule-Definition AVP. The Charging-Rule-Definition AVP is extended to include these additional AVPs "L7-Application-Description" and "Rule-Condition-Action" to support L7 capabilities.

A new CLI command **policy-control l7-dynamic-rules** is introduced in the ACS Configuration mode to enable L7 capabilities through Charging-Rule-Definition AVP received over Gx interface.

These optional grouped AVPs "L7-Application-Description" and "Rule-Condition-Action" are supported in r8-gx-standard dictionary.

- **L7-Application-Description**: This AVP is part of dynamic rule. This AVP carries L7 information with the L7 dynamic rule. This L7 filter is used by rule matching logic.

- L7-Protocol-Name: This AVP specifies the protocol name for the application.

  This is an enumerated value received from PCRF. In Release 20, only HTTP Protocol is supported.

- L7-Field: This AVP specifies the name of field to be matched from the protocol.

  This is an enumerated value received from PCRF. In Release 20, only URL Field is supported.

- L7-Operator: This AVP specifies the operator to be used for matching the values.

  The following operators are supported:

  - EQUALS (1)
  - STARTS_WITH (2)
  - ENDS_WITH (3)
  - CONTAINS (4)
  - NOT_EQUALS (5)
  - NOT_START_WITH (6)
  - NOT_END_WITH (7)
  - NOT_CONTAINS (8)

- L7-Case-Sensitivity: This AVP mentions if the above L7-Value field has to be compared with or without case-sensitivity.
- L7-Value: This AVP mentions the value that is to be compared with the one received in the user packet. This is a string with length of 256 characters.

- **Rule-Condition-Action**: This AVP specifies the special action to be taken by PCEF when the dynamic rule is matched and conditions are met. This is part of Charging-Rule-Definition AVP and can be received in CCA-I/CCA-U/RAR.
  - Rule-Condition: This AVP mentions the condition with the action that has to be applied for the call. In Release 20, Out-of-Credit is the only condition supported.
  - Rule-Action: This AVP mentions the action to be taken when the above condition occurred for the call. In Release 20, only Allow action is supported.

Relationship to Other Features

L7 Dynamic Rule support is extended to the existing Flow Aware Packet Acceleration (FAPA) and Transactional Rule Matching (TRM) features. The L7 Dynamic Rule Activation feature is independent of these two features.

The flows can be accelerated when the subscriber packets match the L7 dynamic rules. When subscriber quota for a rating group is exhausted and when the OOC condition action is being applied for the rule, accelerated path is not applied to the flow. Once the required quota is available, the accelerated path will resume for that flow.

Limitations

The following are the limitations of this feature:

- A maximum of up to eight L7 application descriptions is supported per one L7 rule.
- When deploying Release 20 software, an L7 dynamic rule is installed correctly with/without flow description and the data flow begins. If a downgrade to any previous software releases is performed, the L7 information checkpoints are not decoded, resulting in a rule without any flow description or L7 information.

How it Works

L7 Static Rule Detection

This section explains how an L7 based ruledef (static rule) can be added to the gateway.

Configuration of the following entities is required to support L7 analysis.

- Routing ruledef: This ruledef captures the L3/L4 filters for the protocol. When packets matching to these filters are received, the packets are passed to the configured analyzer. One or more routing rule lines can be configured in ruledef.
- Charging L7 based ruledef: This ruledef captures the L7 based filters supported by the gateway. This ruledef gets packets only after the routing rule has matched the L3/L4 filters and the application has been identified as matching to the configured value. One or more charging rule lines can be configured in ruledef.
- Charging action: This captures the charging and policy parameters for the charging ruledef. These parameters are used when the charging ruledef is hit.
- Rulebase: Rulebase is a set of routing and charging ruledefs, which will be applied for the subscriber.
  - Route priority line: This configuration links the routing ruledef with the protocol and activates the routing ruledef for the subscriber.
  - Action priority line: This configuration links the charging action with charging ruledef and activates the charging ruledef for the subscriber.

L7 Dynamic Rule Handling

This section explains how the dynamic rule can be extended to support L7 capabilities.

The Flow-Description AVP that is already part of Charging-Rule-Definition AVP is used like a "Routing Rule". The packets matching to the Flow-Information will be sent to the analyzer mentioned in L7-Protocol-Name AVP.

When an L7 dynamic rule is received with Flow-Description AVP, the Flow-Description AVP is used for routing the packets matching to the protocol specified through L7-Application-Description AVP. The gateway internally creates a route using the Flow-Description AVP from the dynamic rule to the protocol mentioned in the L7-Protocol-Name AVP. Hence, all flows matching the specified criteria are sent for protocol analysis.

When Flow-Description AVP is not received with the dynamic rule, default routes are used to enable the corresponding protocol routing. The well-known port numbers are used to enable the protocol analyzer. Since only HTTP protocol is supported, port 80 is used for enabling the protocol.

For rule matching, both the criteria associated with Flow-Description and L7-Application-Description AVPs are used. If Flow-Description AVP is not received, only the criteria associated with L7-Application-Description AVP is used.

For dynamic rule modification, the L7 dynamic rule is installed again with a new set of values for L7-Application-Description AVP. The old values are overridden with the new values and added to the dynamic rule. For removal of L7 dynamic rules, Charging-Rule-Remove AVP is used.

The following are some additional points related to handling of L7 dynamic PCC rules from PCRF.

- L7 dynamic rule binding is similar to the normal L3 dynamic rule.
- If L7 rule does not contain TFT filter then rule will be bound to the bearer matching QoS.
- Session Recovery (SR)/Inter-Chassis Session Recover (ICSR) will be supported for L7 dynamic rules.

In releases prior to 20, when invalid values are sent for Rule-Condition and Rule-Action AVPs from PCRF for a dynamic rule, gateway accepts and installs the dynamic rule. In 20 and later releases, the gateway rejects the dynamic rule with invalid condition-action, and reports the failure with the cause "GW/PCEF_MALFUNCTION (4) ". The same behavior is observed even when the AVP fields are empty.

Configuring L7 Dynamic Rule Activation Feature

This section describes how to configure the activation of L7 dynamic PCC rules from PCRF to support L7 capabilities.

Important This feature requires a valid license to be installed prior to configuring this feature. Contact your Cisco account representative for more information on the licensing requirements.

To enable the L7 capabilities through Charging-Rule-Definition AVP received over Gx interface, use the following configuration:

```
configure
require active-charging
   active-charging service service_name
      policy-control l7-dynamic-rules
      end
```

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.

- The **policy-control l7-dynamic-rules** CLI command is license dependant.
- **policy-control l7-dynamic-rules**: Enables the L7 capabilities through Charging-Rule-Definition AVP received over G.x interface.
- **no policy-control l7-dynamic-rules**: Disables the L7 capabilities through Charging-Rule-Definition AVP. By default, this functionality is disabled.

Verifying the L7 Dynamic Rule Activation Feature Configuration

To verify your configuration, in the Exec mode, enter the following command:

```
show configuration
```

The output displays a concise list of settings that you have configured for the context. From this output, you can confirm if the feature is enabled.

Monitoring and Troubleshooting the L7 Dynamic Rule Activation Feature

This section provides information regarding show commands and/or their outputs in support of the L7 Dynamic Rule Activation feature.

show active-charging sessions full all

The output of this show command has been enhanced to display the L7 filters, condition and action received along with the L7 dynamic rule when the rule is installed.

- Total L7 Dynamic Rules
- Dynamic Charging Rule Definition(s) Configured:
  - L7-Filter
  - Protocol
  - Field
  - Operator
  - Value
  - Case-Sensitive
  - Condition-Action
  - Condition
  - Action

show active-charging service statistics

The following statistics are added to the output of **show active-charging service statistics** command in support of the L7 Dynamic Rule Activation feature.

- Dynamic Rule Statistics:
  - L7 Rules Received: Displays the total number of L7 dynamic rules that are received from PCRF.
  - L7 Install Succeeded: Displays the total number of L7 dynamic rules that are successfully installed.
  - L7 Install Failed: Displays the total number of L7 dynamic rules that failed to install due to invalid L7 dynamic rules, etc.

- Install Failure Reason:
  - L7 Rule Invalid: Displays the total number of L7 dynamic rules that failed to install due to invalid L7 dynamic rule.
  - L7 Protocol Invalid: Displays the total number of L7 dynamic rules that failed to install due to invalid L7 protocol.
  - L7 Field Invalid: Displays the total number of L7 dynamic rules that failed to install due to invalid L7 field.
  - L7 Operator Invalid: Displays the total number of L7 dynamic rules that failed to install due to invalid L7 operator.
  - L7 Value Invalid: Displays the total number of L7 dynamic rules that failed to install due to invalid L7 value.
  - L7 Case-Sens Invalid: Displays the total number of L7 dynamic rules that failed to install due to invalid case-sensitive value.

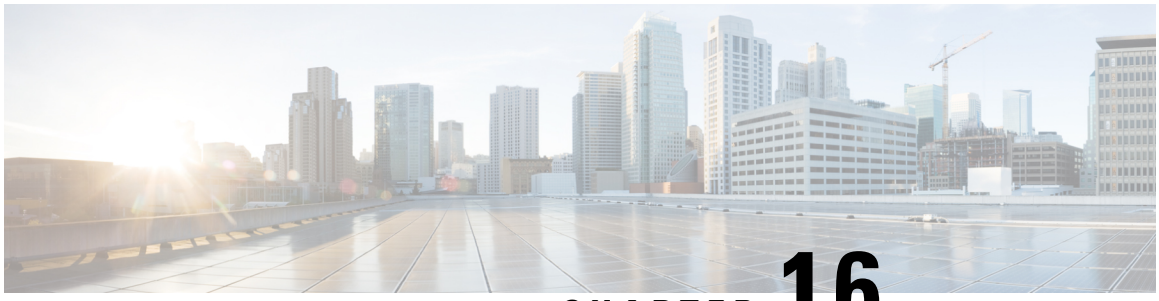
show active-charging rulebase statistics name

The output of this show command has been enhanced to display the total number of packets that are deemed candidates for condition action OOC (Out of Credit).

- Condition Action Statistics:
  - Out of Credit allow actions received: Total number of times the "out of credit allow" actions have been received.
  - Action applied to packets: Total number of packets to which the "out of credit allow" actions are applied.
  - Action applied to bytes: Total number of bytes to which the "out of credit allow" actions are applied.

Monitor Protocol

When using the **monitor protocol** command in the Exec mode, enable option 75 - 3 to check whether or not the L7 dynamic rule is installed successfully.



CHAPTER 16

Location Based QoS Override

This feature enables the gateway to override the QoS values based on subscriber location and also to provide unlimited bandwidth to subscribers.

The following sections provide more detailed information:

Feature Description

With the previous implementation, subscriber bandwidth is limited based on QoS provided by PCRF in order to comply with 3GPP standards. In this release, subscriber is provided with unlimited bandwidth by allowing QoS override based on LAC and/or TAC (individual or range) configured in a local-policy (LP) rule on the gateway. If the subscriber is in the LAC or TAC region and hits the LP rule, the gateway ignores the QoS limits imposed by PCRF and allows the subscriber to have unlimited bandwidth.

Important This feature requires the license to configure local-policy. For more information on the licensing requirements, contact Cisco account representative.

Configuration changes are performed at both ECS and local-policy to achieve this functionality.

For this feature to work, the following operations must be performed in the order as specified below:

- When a subscriber is in the configured RAI or TAI range, the local-policy identifies the associated local-policy rule and sends the rule to ECS for activation.
- ECS provides a configuration in the service scheme, which matches the active lp-rules trigger-condition and associates the trigger-action to be taken when a local-policy rule activation is received. ECS then performs throttle suppression to provide unlimited bandwidth based on the subscriber location.

Important The service-scheme is associated with the subscriber based on the conditions configured under subscriber-class at call setup time.

The location based throttle suppress will override the existing functionality of time based throttle suppression. Irrespective of the configured time delay, if the subscriber is in a particular location for which throttle suppression has been configured then it will hold effect.

Whenever there is new-call or location-change event, the rulebase is checked for location and the LP rules to add and delete are passed to ECS module.

By default, the Change Reporting Action (CRA) notification is sent to MME when ULI-Change, TAI-Change and/or ECGI-Change are installed as part of actiondef configuration. New CLI configuration is provided to control the CRA notification towards MME.

The QoS Override feature can be enabled for subscribers classified based on APN, virtual-APN, rulebase or a combination of these.

This feature works in local-policy fallback, dual-policy mode, and dual-fallback modes. In local-policy fallback mode, irrespective of where the event triggers are being registered, all the event triggers will be reported to local-policy and the corresponding actions will be taken. Whereas in the dual-policy and dual-fallback modes, the event triggers are sent to both local-policy and PCRF depending on where the triggers are registered. Local-policy module handles all the location related events and PCRF handles all other event triggers.

Relationships to Other Features

It is required to have both local-policy and service scheme framework configured to enable this feature for a subscriber. For redundancy support, the corresponding ICSR configuration must also be present.

The service-scheme framework helps in overriding feature behavior specific to a subscriber or a set of subscribers. The user can update the policies specific to subscribers based on preconfigured events. For more information on the service-scheme framework, see the *ECS Administration Guide*.

How it Works

This section describes how the QoS override is performed based on the location of subscriber.

The following example shows the TAC configuration and how the throttle suppression is applied considering the TAC values as 100 -199 for throttle suppress. The configuration for LAC will be similar to the TAC.

Sample local-policy Configuration:

```
configure
  context context_name
    apn apn_name
    ims-auth-service service_name
    end
configure
  context context_name
    ims-auth-service service_name
     policy-control
         associate failure-handling-template template_name
         associate local-policy-service service_name dual-mode
end
configure
  failure-handling-template template_name
```

```
   msg-type any failure-type any action continue local-fallback
end
configure
  local-policy-service service_name
     suppress-cra event-triggers uli-change
    eventbase eventbase_name
      rule priority 1 event new-call ruledef allcalls actiondef
activate_triggers continue
      rule priority 2 event new-call ruledef tai-group actiondef
activate_lp_action
      rule priority 3 event location-change ruledef tai-group actiondef
activate_lp_action
    ruledef allcalls
      condition priority 1 imsi match *
    ruledef tai_group
      condition priority 1 tai mcc 232 mnc 344 tac ge 100
      condition priority 2 tai mcc 232 mnc 344 tac lt 200
    actiondef activate_triggers
      action priority 1 event-triggers uli-change
    actiondef activate_lp_action
      action priority 1 activate-lp-rule name tac_list_1
end
```

Sample Configuration at ECS:

```
configure
   active-charging service s1
       trigger-action ta1
          throttle-suppress
          exit
       trigger-condition tc1
         local-policy-rule = tac_list_1
       exit
       trigger-condition tc2
         any-match = TRUE
       exit
       service-scheme s1
         trigger sess-setup
         priority 1 trigger-condition tc2 trigger-action ta1
         trigger loc-update
         priority 1 trigger-condition tc1 trigger-action ta1
         exit
       subs-class sc1
         rulebase = rb1
       exit
       subscriber-base sb1
         priority 1 subs-class sc1 bind service-scheme ss1
       exit
```

Local-policy provides ECS, the list of rules to activate and the list of rules to delete. In case, the rule to be activated is already installed, ECS ignores this rule. Similarly if the rule to be deleted was not installed, ECS ignores this rule as well. The trigger action will be applied only to a subset of traffic that matches the criteria

defined under trigger condition. If trigger-condition is any-match, then trigger action will be applied to all the flows created after event activation.

Important The "lp-activate-rule" action must be configured as part of "new-call" and "location-change" events.

Based on the subscriber location, the local-policy reporting and the trigger actions vary as provided below.

1. When subscriber starts a session in a zone with tac = 100, local-policy reports a lp-rule-name-install event as it matches the rule tac_list_1.
2. ECS matches the event against the entries for trigger type “loc-update” and perform the throttle-suppress action.
3. If the subscriber moves out to a different zone, LP reports an lp-rule-name-remove for tac-list-1 so that ECS can turn off throttle-suppress.

Limitations

This section identifies the known limitations of this feature.

- Throttling is not supported for uplink packets in case of APN-AMBR. When applying throttling to such packets, it will have no effect.
- For a given ruledef all the conditions should match. Due to this, the current implementation on local-policy has the following restrictions:
  - For every new set of MCC and MNC, a new ruledef should be configured. Also, for every ecgi/3guli/tai range of the given MCC/MNC, a new ruledef should be configured.
  - For each ruledef, new-call and location-change events should be configured additionally in “eventbase” configuration.
  - If a disjoint set of TAC or LAC should be configured, note that up to 32 such values can be configured in a ruledef.
  - Local-Policy supports up to 7 lp-rules to be activated for a given session.
  - Based on location received in CPC or UPC, necessary action is taken in local-policy and the LP rule is activated. If the location is received without that ULI type in the next message (update PDP context), then this will be considered as a location change with ULI value as 0 and the rule will be deactivated even if the location of the UE is not changed.
  - The location change is not identified in the response of network requested update PDP context message. Also if the location change is notified along with call termination, then this is not identified as a location-change.
  - Upon receiving SGSN change and if SGSN supports CRA handling, the CRA with value 0 is reported even though the CRA reporting is suppressed by LP/PCRF.

Configuring Location Based QoS Override

The following sections provide the configuration commands to enable location based QoS Override functionality.

This functionality is achieved through the CLI configurations provided at both local-policy and ECS.

Local-Policy Configurations

The following sections provide the configuration commands that should be enabled within local-policy for the feature to work.

Activating Local-Policy Rule

Use the following configuration to activate the local-policy rule within service scheme based on the configured RAI or TAI range of subscribers.

```
configure
  local-policy-service local_policy_service_name
    actiondef actiondef_name
      action priority priority activate-lp-rule name lprule_name
    end
```

Notes:

- **activate-lp-rule** *lprule_name*: This keyword activates a local-policy rule within service scheme when a subscriber is in the configured RAI or TAI range. *lprule_name* must be an existing local-policy rule within the service scheme expressed as an alphanumeric string of 1 through 63 characters.

  When the subscriber moves out of the configured RAI or TAI range, the local-policy rule is deactivated.

- Local-Policy can support up to 7 lp-rules to be activated for a given session.

Controlling CRA Events

Use the following configuration to suppress the CRA for event triggers enabled in local policy configurations.

```
configure
  local-policy-service local_policy_service_name
    suppress-cra event-triggers { ecgi-change | tai-change | uli-change
 } +
    end
```

Notes:

- **suppress-cra event-triggers { ecgi-change | tai-change | uli-change }**: This keyword restricts sending of CRA towards MME depending on the ECGI-Change, TAI-Change and ULI-Change event triggers configured in local-policy service.

- Use the **no suppress-cra** command to configure the default behavior. By default, the CRA notification is sent to MME if one or a combination of these event triggers is installed.

Configuring Location Change Event Triggers

Use the following configuration to install ECGI Change, TAI-Change and/or ULI-Change event triggers from local policy.

```
configure
   local-policy-service service_name
      actiondef actiondef_name
       action priority priority event-triggers { ecgi-change | tai-change
 | uli-change }
     exit
    eventbase default
         rule priority priority event new-call ruledef ruledef_name actiondef
 actiondef_name [ continue ]
         end
```

Notes:

- **priority** *priority*: Specifies a priority for the specified action. **priority** must be unique and an integer from 1 to 2048.
- **event-triggers { ecgi-change | tai-change | uli-change }**: This keyword specifies to install ECGI-Change, TAI-Change and/or ULI-Change event triggers. If enabled, the respective event triggers are installed from local policy.
- This CLI command is configured in local-policy if operator wants to enable the respective change notification in MME by sending a CRA value.

Applying Rules for TAI-Change Event

Use the following configuration to enable TAI-Change detection and take specific action for TAI-CHANGE event reported by MME.

```
configure
   local-policy-service service_name
      eventbase eventbase_name
       rule priority priority event tai-change ruledef ruledef_name actiondef
actiondef_name [ continue ]
    end
```

Notes:

- **priority** *priority*: Specifies a priority for the specified rule. *priority* must be unique and an integer from 1 to 2048.
- **ruledef** *ruledef_name*: Associates the rule with a specific ruledef. *ruledef_name* must be an existing ruledef within this local QoS policy service.
- **actiondef** *actiondef_name*: Associates the rule with a specific actiondef. *actiondef_name* must be an existing actiondef within this local QoS policy service expressed as an alphanumeric string of 1 through 63 characters.
- **tai-change**: Enables a new event to detect TAI-Change and applies specific action for the TAI-Change event as defined in actiondef configuration.
- **continue**: Subsequent rules are also matched; otherwise, rule evaluation is terminated on first match.

Enforcing LP Rule based on Event Parameter Values

Use the following configuration to apply rules based on the values of ECGI, 3G-ULI, and TAI received in event notification by MME.

```
configure
  local-policy-service service_name
    ruledef ruledef_name
     condition priority priority ecgi mcc mcc_num mnc mnc_num eci { eq | ge
| gt | le | lt | match | ne | nomatch } regex | string_value | int_value | set
 }
       condition priority priority tai mcc mcc_num mnc mnc_num tac { eq | ge
 | gt | le | lt | match | ne | nomatch } regex | string_value | int_value |
set }
       condition priority priority 3g-uli mcc mcc_num mnc mnc_num lac { eq |
 ge | gt | le | lt | match | ne | nomatch } regex | string_value | int_value
 | set }
   exit
```

Notes:

- **priority** *priority*: Specifies a priority for the specified condition. *priority* must be unique and an integer from 1 to 2048.
- **ecgi mcc** *mcc_num* **mnc** *mnc_num* **eci**: Configures ECGI with values for MCC, MNC and ECI.
  - *mcc_num*: MCC is a three digit number from 001 to 999. It is a string of size 3 to 3.
  - *mnc_num*: MNC is two/three digit number from 01 to 999. It is a string of size 2 to 3.
  - **eci**: ECI is a hexadecimal number from 0x1 to 0xfffffff. It is a string of size 1 to 7.
- **tai mcc** *mcc_num* **mnc** *mnc_num* **tac**: Configures TAI with values for MCC, MNC and TAC.
  - *mcc_num*: MCC is a three digit number from 001 to 999. It is a string of size 3 to 3.
  - *mnc_num*: MNC is two/three digit number from 01 to 999. It is a string of size 2 to 3.
  - **tac**: TAC is a 4 byte field. It is a string of 4 hexadecimal values from 0x1 to 0xffff.
- **3g-uli mcc** *mcc_num* **mnc** *mnc_num* **lac**: Configures 3G-ULI parameter with values for MCC, MNC and LAC.
  - *mcc_num*: MCC is a three digit number from 001 to 999. It is a string of size 3 to 3.
  - *mnc_num*: MNC is two/three digit number from 01 to 999. It is a string of size 2 to 3.
  - **lac**: LAC is a 4 byte field. It is a string of 4 hexadecimal values from 0x1 to 0xffff.
- This CLI command is configured in local-policy if operator wants to take specific action based on certain event parameter value received in Change event notification by MME.

ECS Configurations

The following section provides the configuration commands that should be enabled within ECS for the feature to work.

Enabling Location Based QoS Override

Use the following configuration to enable QoS override based on subscriber location.

```
configure
   active-charging service service_name
      trigger-action trigaction_name
         [ no ] throttle-suppress
      exit
```

Notes:

- **throttle-suppress**: This keyword allows operators to suppress the throttling when the subscriber is in a particular LAC or TAC location.
- Use the **no throttle-suppress** CLI command to disable this feature for the subscriber.

Configuring Local-Policy Rule within ECS

Use the following configuration to specify the local-policy rule within ECS for enabling trigger condition.

```
configure
   active-charging service service_name
      trigger-condition trigcond_name
         [ no ] local-policy-rule = lprule_name
      exit
```

Notes:

- **local-policy-rule**: This keyword allows operators to suppress the throttling when the subscriber is in a particular LAC or TAC location and hits the specified local-policy rule. The local-policy-rule contains either a list, range, or index of LAC and/or TAC entries.
- *lprule_name*: Specifies the local-policy rule name. *lprule_name* must be an existing local-policy rule within the service scheme expressed as an alphanumeric string of 1 through 63 characters.
- Use the **no local-policy-rule** CLI command to disable this feature for the subscriber.

Verifying the Location Based QoS Override Configuration

Use the following command to verify the configuration status of this feature.

```
show configuration
```

This command displays all the configurations that are enabled within the chassis for the subscriber. This display can be used to verify if the Location based QoS Override feature is enabled or disabled.

This is an example configuration to enable this feature for a subscriber on a particular rulebase rb1, for a particular local-policy-rule for zone A.

```
configure
active-charging service s1
   trigger-action ta1
       throttle-suppress
   exit
   trigger-condition tc1
     local-policy-rule = zone_A
```

```
exit
service-scheme ss1
  trigger loc-update
    priority 1 trigger-condition tc1 trigger-action ta1
  exit
subs-class sc1
  rulebase = rb1
exit
subscriber-base sb1
  priority 1 subs-class sc1 bind service-scheme ss1
exit
```

Monitoring and Troubleshooting the Location Based QoS Override

This section provides information regarding show commands and/or their outputs in support of this feature.

Use the following CLI commands and collect the output to troubleshoot if any issue is encountered with this feature.

```
logging filter active facility local-policy level debug
show local-policy statistics all
show active-charging sessions full all
show ims-authorization sessions full all
logging filter active facility ims-auth level debug
```

show active-charging subscribers full all

The following field is newly added to the output of this show command in support of this feature.

- Local-policy RAI/TAI Rules Active List – Displays the list of local-policy rules for RAI/TAI that are currently activated for the subscriber.

show active-charging trigger-action all

The following field indicates whether suppress throttling is enabled or disabled.

- Throttle Suppress

show active-charging trigger-condition all

The following field displays the name of the configured local-policy (LP) rule.

- Local-policy Rule Name

show ims-authorization policy-control statistics

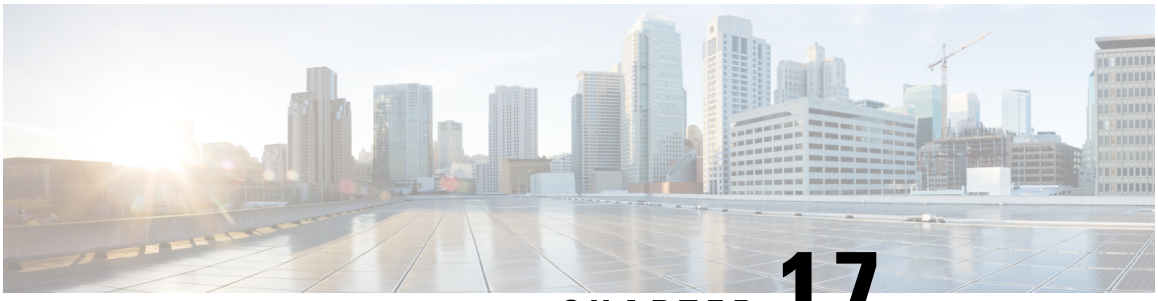
The following field is newly added to the output of this show command in support of this feature.

- Session Recovery Failure
  - Activate-LP-Rule – This field indicates the number of times lp-activate-rules session recovery or ICSR recovery failed.

show local-policy statistics all

The following fields are newly added to the output of this show command in support of this feature.

- Event Statistics
  - 3G-ULI Change – Displays the number of 3G-ULI-CHANGE event triggers that has been received by Local-Policy
  - TAI Change – Displays the number of TAI-CHANGE event triggers that has been received by Local-Policy
- Action Statistics
  - Activate LP Rule – The total number of times the lp-activate-rule action is triggered by local-policy module.
  - Activate LP Rule Failure – The total number of times the lp-activate-rule action fails.
  - Activate LP Rule Success – The total number of times the lp-activate-rule action succeeds.
- Variable Matching Statistics
  - 3G-ULI – Displays the number of times the 3G-ULI value is matched and the specific action is applied based on the event.
  - TAI – Displays the number of times the TAI is matched and the specific action is applied based on the event.



CHAPTER 17

OpenDNS Feature

This chapter describes the following topics:

Feature Summary and Revision History

Summary Data

Applicable Product(s) or Functional Area	• P-GW • SAEGW
Applicable Platform(s)	All
Feature Default	Disabled - Configuration Required
Related Changes in This Release	Not Applicable
Related Documentation	• *Command Line Interface Reference* • *P-GW Administration Guide* • *SAEGW Administration Guide*

Revision History

Revision Details	Release
First introduced.	21.6

Feature Description

Important This is a licensed controlled feature. Contact your Cisco account representative for detailed information on specific licensing requirements.

The OpenDNS feature provides DNS-based security policies to secure the subscriber traffic based on the policy associated with it.

StarOS already supports readdressing of DNS traffic to the specific DNS server. Configuration for readdressing of DNS traffic is available in the charging action of the ECS service. OpenDNS functionality can be invoked on a per-subscriber basis by associating such charging action to predefined rules. Hence, with this feature, by having the PCRF control activation and deactivation of such predefined rules, the readdressing of DNS traffic is made subscriber specific.

New CLI commands have been added to the ACS configuration to support configuration of EDNS format containing fields for the DNS header enrichment:

- MSISDN
- PGW-IP-Address
- APN Name
- IMSI
- Device-id

New CLI command has been added for associating the Device-id's with the security profiles to be applied.

Limitations: Following are the limitations of this feature:

- Registration for Device-ids not supported currently. These are retrieved offline and configured against the respective security profiles.
- Integrity of Device-ids is not validated on the SAEGW.

Configuring Commands for Enabling OpenDNS Feature

This section covers configuration commands used in this feature. Any change in the relevant configuration or activation or deactivation of an associated rule is applicable on the subsequent DNS requests.

Configuring EDNS Mode

The EDNS Mode has been added in the Active Charging Service Configuration Mode to configure EDNS format and fields. You this configuration when you want to convert the DNS traffic to an EDNS request.

This command allows you to enable or disable EDNS Configuration Mode.

```
configure
```

```
active-charging service service_name
  edns
  [ no ] edns
  exit
```

Entering this command sequence results at the following prompt:

```
[local]host_name(config-acs-edns)#
```

Configuring Commands in EDNS Mode

```
configure
  active-charging service service_name
    edns
    [ no ] { fields | format } name | security-profile name device-id
device-id
    exit
```

NOTES:

- **edns**: Enables EDNS format configuration mode.
- **fields**: Defines EDNS fields tag value.
- **format**: Enables EDNS format configuration.
- **name**: Defines the name of EDNS field or EDNS format or security profile.
- **security-profile**: Associates security profile to Device-id.
- **device-id**: Defines the Device-id to map to a EDNS profile.

Configuring the EDNS Fields Mode

This command allows you to enable or disable EDNS Fields Configuration Mode.

```
configure
  active-charging service service_name
    edns
      fields fields_name
      [ no ] fields fields_name
      exit
```

Entering this command sequence results at the following prompt:

```
[local]host_name(config-acs-edns-fields)#
```

NOTES:

- **fields**: Defines EDNS fields tag value.

Configuring Commands in EDNS Fields Mode

```
configure
  active-charging service service_name
```

```
    edns
      fields fields_name
        [ no ] tag { val { imsi | msisdn | pgw-address | apn-name } {
encrypt } } | default device-id }
        exit
```

NOTES:

- **fields**: Inserts EDSN field.
- *fields_name*: Defines the fields name.
- **tag**: Defines the tag value for EDNS fields.
- **val**: Defines the tag value for EDNS fields. This is an integer value between 1 and 65535. Tag value is of 2 bytes.
- **imsi**: Defines the IMSI of the subscriber.
- **msisdn**: Defines the MSISDN of the subscriber.
- **default**: Defines the standard opt-code value.
- **apn-name**: Defines the access point name of the subscriber to which it is connected.
- **device-id**: Defines device-id learned during registration.
- **encrypt**: Encrypts the subscriber traffic. This option is available for IMSI and MSISDN only.

Important If encoding of any of the fields fails, EDNS insert does not happen.

Configuring the EDNS Format Mode

This command allows you to enable or disable EDNS Format Configuration Mode.

```
configure
  active-charging service service_name
    edns
      format format_name
      [ no ] format format_name
      exit
```

NOTES:

- **format**: Enables EDNS format configuration.
- *format_name*: Defines the name of EDNS field or EDNS format.

Configuring Commands in the EDNS Format Mode

```
configure
  active-charging service service_name
```

```
edns
   format format_name
      fields fields_name encode
      [ no ] fields name
      exit
```

NOTES:

- **format**: Associates fields with format.
- *format_name*: Defines the format name.
- **fields**: Inserts the EDNS field.
- *fields_name*: Defines the fields name.
- **encode**: Defines fields to be used for encoding EDNS message.

Configuring Security Profile

Use this CLI command to configure the security profile in EDNS to add mapping with the Device-id.

```
configure
   active-charging service service_name
      edns
         [ no ] security-profile security_profile_name
         exit
```

NOTES:

- **security-profile**: Defines the security profile configuration in the EDNS to add mapping with the Device-id.
- *security_profile_name*: Defines the name of the security profile. This is a string of size 1 to 50.

Associating Charging Action to EDNS Format and Tag to Identify the Device-ID

This CLI command associates the Device-ID's with the security profiles to be applied. If any of the associated formats is not configured or the configured field value is not available for encoding, then the DNS request is sent unchanged and no EDNS translation is performed.

```
configure
   active-charging service service_name
      charging-action charging_action_name
      [ no ] edns format edns_format_name { security-profile profile_name {
encryption rc4md5 encrypted key key_string } }
      exit
```

NOTES:

- **edns format**: Defines the EDNS format.
- *edns_format_name*: Defines the EDNS format name. This is a string of size 1 to 63.

- **security-profile**: Associates the EDNS security profile to the charging action.
- *security_profile_name*: Defines the name of the EDNS security profile. This is a string of size 1 to 50.
- **encryption**: Encrypts the EDNS header fields.
- **encrypted-key**: Designates use of encryption.
- *key*: Defines key used to encrypt EDNS header fields. This is string of size 1 to 255.
- **rc4md5**: Defines the encryption type. This is hardcoded value.

Important Since any other encryption type is not supported currently, the encryption type rc4md5 is hardcoded.

Sample Configuration

This section displays the sample configuration.

```
config
  active-charging service acs
    edns
      security-profile profile_high device-id 1234567890abcdef
       format format_xyz
        fields field_xyz encode
    exit
  fields field_xyz
        tag imsi 10 encrypt
        tag msisdn 20
        tag pgw_address 30
        tag apn-name 40
        tag default device-id
      exit
     exit
  exit
  charging-action action
     edns format format_xyz security-profile profile_high
  exit
```

Show Commands and Outputs

This section provides information regarding show commands and their outputs in support of the feature.

show active-charging analyzer statistics name dns

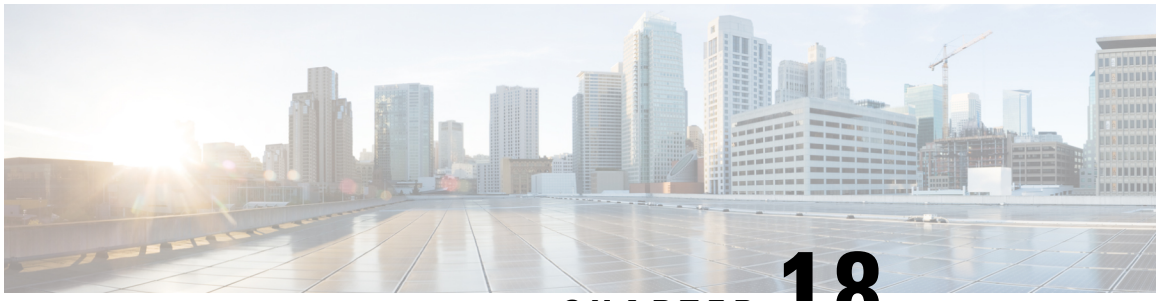
The following new fields are added to the show command to indicate the EDNS encoding status:

- EDNS over UDP:
  - Authorization with S6b: HSS-EGTP-S5S8 GN-GP-Disabled
  - Authorization with S6b: HSS-EGTP-S5S8 GN-GP-Enabled

show active-charging charging-action name action

The following new fields are added to the show command to indicate the EDNS Information:

- EDNS Info:
  - Format Name: Displays the format name of the EDNS format
  - Encryption Type: Displays the encryption type of EDNS header field
  - Encryption Key: Displays the encryption key of the EDNS header fields.
  - Security Profile: Displays the security profile of the associated EDNS security-profile to charging action



CHAPTER 18

Override Control

This chapter describes the Override Control feature and provides detailed information on the following topics:

Feature Summary and Revision History

Summary Data

Applicable Product(s) or Functional Area	• ECS • P-GW
Applicable Platform(s)	ASR 5500
Feature Default	Disabled - Configuration Required
Related Changes in This Release	Not Applicable
Related Documentation	*ECS Administration Guide*

Revision History

Important Revision history details are not provided for features introduced before releases 21.2 and N5.5.

Revision Details	Release
In this release, support is added for Execution-Time AVP in Override-Control AVP to allow the overridden parameters to be specified for a rule (static or predefined), for one or all charging actions (using a wildcard), with the ability to exclude certain rules.	21.3

Revision Details	Release
First introduced.	Pre 21.2

Feature Description

Override Control (OC) feature allows the customer to dynamically modify the parameters of static or predefined rules with parameters sent by PCRF over the Gx interface.

Important Override Control is a license-controlled feature. A valid feature license must be installed prior to configuring this feature. Contact your Cisco account representative for more information.

The Inheritance feature does not support overwriting parameters at rule/charging action level, and exclusion of more than one rule. In order to provide this flexibility and also have a generic capability on chassis, Override Control feature is introduced. This feature will define a set of custom AVPs that will enable the PCRF to override charging and policy parameters for all rules (wildcard) or a specified set of rules or charging actions.

The override values should be sent by PCRF over Gx using the custom AVPs. Override Control provides this capability while addressing the limitations with Inheritance feature like rule level control, charging action level control, exclusion of more than one rule, different override values to be specified for a subscriber, etc. So, the Override Control feature will replace the Inheritance feature.

Override Control feature can be configured at the rulebase level. The Diameter capability exchange message should indicate support for Override control feature when the **override-control** CLI command is configured in the rulebase configuration mode.

Important Both Inheritance and the Override Control features are supported in this release. Note that these two features should not be enabled simultaneously. If these features are enabled by mistake, only Override Control is applied.

In 19 and later releases, support to override Group-of-Ruledefs is provided for the Override Control feature. Override sent for a group-of-ruledefs will apply to all the ruledefs defined in a group. The same Override-Rule-Name AVP is used to send Ruledef or Group-of-Ruledef interchangeably. The two AVPs — Override-Rule-Name and Override-Charging-Action-Exclude-Rule, support either a Ruledef name or a Group-of-Ruledefs name.

The Gx interface is updated to include custom AVPs for the PCRF to send override values to P-GW. These override values may be sent for all rules (wildcard) or for specific rule(s) or for charging action(s). In case the override values are sent for a charging action, a rule or some of the rules may be excluded from using the override values by sending the rules names in the Gx message. The override values will be check pointed and recovered in case of either standalone recovery or ICSR.

This Override Control feature is expected to maintain existing active calls using Inheritance post upgrade. Inheritance feature and Override Control should not be enabled simultaneously. It is necessary that Inheritance feature be turned off once Override Control feature is enabled. Override Control once enabled will apply only to new calls and does not affect the existing calls.

When multiple overrides are received from PCRF, the following is the priority in which they are applied:

1. Rule level override control
2. Charging action level override control
3. Wildcard level override control

When installing a predef rule, if override control is received for that predef rule and QCI/ARP is overridden, then the new overridden QCI/ARP values are used for bearer binding of the predef rule. If the QCI/ARP is not overridden, then the values configured in charging action is used. The override charging and policy parameters received from PCRF will continue to apply for the entire duration of the call. These values may be modified by PCRF by sending the modified values with the same override control criteria (Rule name(s), Charging Action Name(s) and Exclude Rule(s)). Any change in the Override Control criteria will be interrupted as a new OC. There can only be one wildcard OC installed for a subscriber.

Important If two or more different rules with different rule definitions are associated with the same charging action, the operator must ensure that override control sent from PCRF does not bind these rules to different bearers by overriding the QCI/ARP combination for selective rules.

In 20 and later releases, the status of Override Control can be checked based on whether OC is enabled or disabled for the call. This is useful to trace calls for which Override Control is not installed and can also be used for debugging purposes. The OC status will be displayed based on the following conditions:

- If for any call, override control is installed for a static rule, status will be displayed as ON.
- If for any call, override control is installed for any pre-defined rule in rulebase and even if these particular pre-defined rule(s) are not yet activated through PCRF, status will be displayed as ON.
- If inheritance is enabled for calls, status will be displayed as OFF.

Per Subscriber Traffic Steering

In this release, an addional capability is added to the Override Control feature to dynamically route/mark traffic on a per subscriber basis. Override Control functionality will now support the policy parameters in charging action for post processing rules — Nexthop Address (IPv4 address only) and TOS. Diameter AVPs are also added in support of this feature.

The following are limitations for this feature:

- Override Control supports only IPv4 Nexthop Address in this release.
- If invalid value of Nexthop Address or TOS is received, then the complete Override Control message will be discarded.
- Currently there is no way to withdraw an already applied Override Control message. This can only be modified.
- This feature does not support Group-of-Ruledefs in Override Control.
- This feature enables customers to steer specific subscribers to different services using a unique Nexthop address. This could alternatively be accomplished using a unique VLAN ID as well. Due to the current implementation, this feature does not provide support of VLAN ID on Override Control eventhough VLAN ID can be received in OC message, and the packet can still be routed to default VLAN.

For information on how to configure the Override Control feature, refer to the section *Configuring Override Control Feature* in the *Enhanced Charging Service Configuration* chapter of this guide.

Override Control Name for OC Identification

For identification of OC, PCRF uses one or a combination of the following key parameters:

- Rule names
- Charging-action names
- Exclude-rule names

There is no unique identifier available for identification of OC for a particular subscriber session. In release 20, a new Diameter AVP "Override-Control-Name" is defined in the Override-Control grouped AVP. The OC name specified in the AVP is used as the unique key to identify OC for any further updates like OC modification or deletion.

To meet the new AVP requirement, "**with-oc-name**" keyword has been added to the existing **override-control** CLI command under rulebase configuration. If the **override-control with-oc-name** CLI is configured in rulebase, only OCs with Override-Control-Name AVP are supported and the OCs without name are rejected.

If Override-Control-Name AVP is received when the **override-control** CLI command is configured i.e. OC install is supported without OC name, appropriate error is reported in error logs, OC is dropped and OC failure statistics is incremented. Similarly if **override-control with-oc-name** CLI is configured and OC is received without the name AVP, appropriate error is reported, OC is dropped and OC failure statistics is incremented. On receiving an OC without name, installed OC list (without name) is searched for secondary identification criteria. If no OC with same rule/charging-action/exclude rule list is found, it is installed as a different OC.

Also, for OCs with the name, operator can add rule/charging-action/exclude rule to the existing OC in the same category. That means, the rules can be added to a rule level OC, CA names can be added to a CA level OC, and exclude rules can be added to a wildcard or CA level OC.

OCs received with Override-Control-Name AVP are uniquely identified by the OC name. When the Override-Control-Name AVP is not present in Override-Control AVP, the OCs are identified based on the secondary identification criteria, i.e., the list of rule names, charging-action names, and exclude-rule names as these were the criteria before this feature change.

During rulebase change, the feature to support OC name will be controlled based on the configuration of new rulebase. After rulebase change OC will be accepted as per the CLI configured in new rulebase. This is the only scenario where for a single call session, OC can be installed with both OC name and without OC name.

When software upgrade is done on a standby setup where same rulebase is configured with the CLI **override-control with-oc-name**, then no calls are dropped and OC installation status will remain the same as before upgrade. If new call is established after upgrade and OC is installed with OC-name then this will be applied on new call.

During the downgrade, OC-name will be dropped and OCs will be recreated assuming Rule/CA/Exclude rule name list as the primary key for unique identification.

Wildcard/CA Level Overrides with Exclude Rule List

The current design allows adding Exclude Rules to a wildcard or CA level OC. In releases prior to 20, whenever a wildcard level override or CA level override is sent after another wildcard or CA override, the two wildcards or CAs were not merged. The gateway considers only the latest override at wildcard level or CA level to be applied for OC. In release 20 and beyond, the two wildcards or the two CAs received will be merged including the Exclude Rule lists defined within. Through this change, all the rules in the Exclude Rule lists under both overrides will be excluded from being applied in OC.

Disabling Override Control

The current implementation of OC does not allow disabling an already enabled/configured OC or reverting few parameters of a previously installed OC. It can only be modified but cannot be completely disabled. In release 20, operator is provided with the flexibility to disable an installed OC or overridden parameters associated with the installed OC.

Disabling OC is supported through the use of a new Diameter AVP "Disable-Override-Control". This AVP indicates that override parameters for the subscriber will be removed completely or per parameter basis. This is a grouped AVP containing the following sub-attributes:

- Override-Control-Name: Specifies the name of the Override-Control. This AVP may be included more than once if multiple overrides need to be disabled.
- Disable-Override-Control-Parameter: Specifies the Override Control parameter to be disabled. This AVP may be included more than once if multiple parameters need to be disabled.

Important Enable/Disable OC is a license-controlled feature. Contact your Cisco account representative for more information.

This feature allows disabling of installed overrides at three different levels:

- Wildcard Disable: If Disable-Override-Control AVP is received without any OC name or parameters, all installed OCs (with or without name) will be removed.
- OC level Disable: This level will be supported only for calls where overrides with name have been installed.
  - If OC names are specified in Disable-Override-Control AVP with no parameter specified, all installed OCs with given OC name will be deleted.
  - If Disable OC is received for a call where OC name support is not enabled through the **override-control with-oc-name** CLI command, and the grouped AVP for disable OC contains OC name, proper error log will be generated.
- Parameter level Disable: If parameters are specified in Disable-Override-Control AVP, these parameters will be reverted to static values for all the OCs if no OC names are present or for the specified OCs.
  - If OC names are present, overridden values for the given parameters will be removed from the given list of OCs.
  - If no name is present, overridden values for the given parameters will be removed from all the installed OCs.

Limitations:

Checkpointing of Disable OC to standby chassis does not occur. So, if session manager goes down before the Disable OC request has been processed, OC will not be disabled.

Wildcard Support for Override-Control AVP

This feature allows the operator to send partial rule-names, partial charging-action names and partial exclude-rule names through Override-Rule-Name, Override-Charging-Action-Name and Override-Charging-Action-Exclude-Rule-Name AVPs respectively. The rules, charging-actions or exclude-rule names matching the partial string can be overridden or excluded accordingly.

With the previous implementation of Override-Control feature, wildcard is not supported in these existing AVPs. If PCRF want to exclude 100 rules, then including 100 Override-Charging-Action-Exclude-Rule AVPs in CCA message will impact the efficiency and capacity. To address this problem, the Override-Control AVP wildcard support feature is introduced.

The delimiter "<*>" is used to send the partial names over Gx interface. The following string patterns can be added to the Override-Rule-Name, Override-Charging-Action-Name and Override-Charging-Action-Exclude-Rule-Name AVPs:

- Pattern<*> means, the names start with partial string and end with anything
- <*> Pattern means, the names start with anything and end with partial string
- <*> Pattern<*> means, the names contain partial string

For example, Charging-Action-Name: <*>SPDATA01

Important This feature is backward compatible with releases prior to 21 only when the delimiter "<*>" is not used. If this delimiter is used, it will be considered as the full rule name/charging action name/exclude rule name as the support for partial string does not exist.

Support for Execution-Time AVP

In 21.3 and later releases, the support for Execution-Time AVP allows the overridden parameters to be specified for a rule (static or predefined), for one or all charging actions (using a wildcard), with the ability to exclude certain rules. These overrides are sent by the PCRF using the AVP construct in a CCA or RAR message.

As part of this feature, the P-GW supports the following two new AVPs within the proprietary Override-Control grouped AVP:

- Execution-Time (AVP code 132025): This AVP is of type Time. It indicates the Unix Epoch at which the provided Override-Control instance takes effect.
- Override-Control-Pending-Queue-Action (AVP code 132078): This AVP of type ENUM with allowed values FLUSH (0) and RETAIN (1). It indicates the action the gateway takes on the Pending-OC-Queue.

Both the AVPs are included in the Override-Charging-Action-Parameters grouped AVP.

How It Works

The Execution-Time AVP is an optional AVP in the Override-Control grouped AVP, sent in RAR/CCA message by the PCRF. It is valid for Rule-level, Charging-action level, and Wildcard-level OC.

Whenever Override-Control AVP is sent in RAR/CCA message from PCRF, and:

1. it does not contain the Execution-Time AVP, then the existing OC application procedure is adopted for backward compatibility. In other words, the OC parameter is applied immediately by the PCEF.
2. it contains Execution-Time AVP which is in the “Past”, then it is treated as if no Execution-Time AVP was sent and the OC parameter is applied immediately by the PCEF.
3. it contains Execution-Time AVP which is in “Future”, then the PCEF marks the OC as “Pending” and installs the OC when the Execution-Time is reached.

Session Recovery and ICSR

The Pending OCs are recovered on intra/inter chassis session recovery. The PCEF checkpoints Execution-Time using the existing framework that is used for check-pointing of other OC parameters. When the Session Manager restarts or ICSR switchover occurs, timer is started for each of the recovered OC with Execution-Time AVP. The value of this new timer is equal to the time left for the OC to be activated or applied on the call.

If the timer expires during the recovery, then the OC is applied immediately after the recovery.

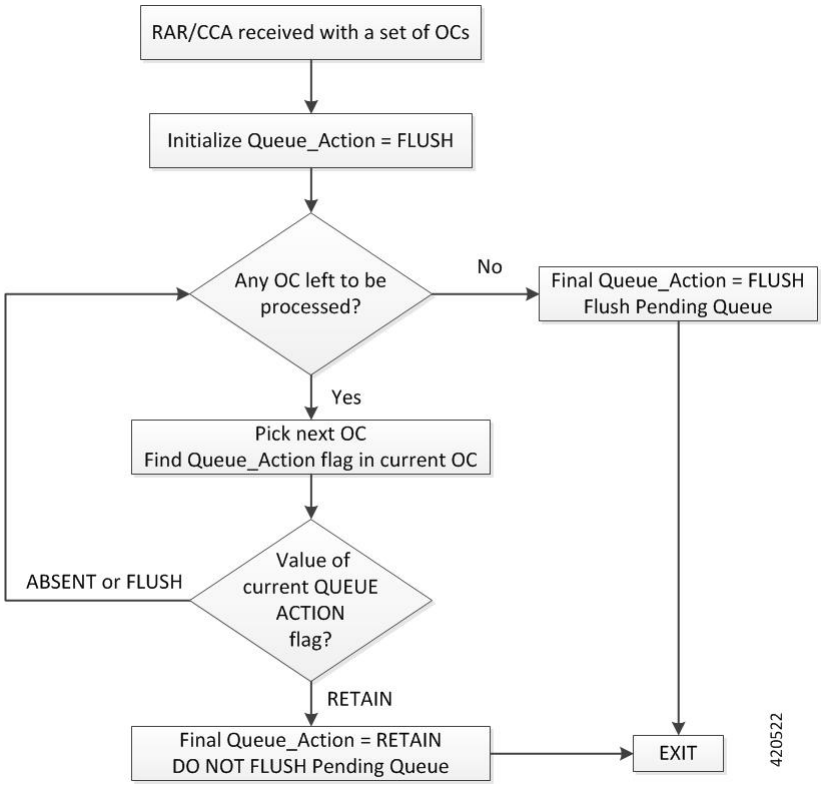
During downgrade from N to N-1 build, the pending OCs on currently active N chassis are not check-pointed to the standby N-1 chassis.

Flushing the Pending OCs

The PCEF decides whether to flush or retain already pending OCs for a subscriber, based on the presence or absence of the Override-Control-Pending-Queue-Action AVP.

On receiving a new OC (with or without Execution-Time AVP) in a new RAR/CCA message, the PCEF flushes all the previous pending OCs for that subscriber. The PCEF either buffers or applies the newly received OC, based on the presence or absence of the Execution-Time AVP respectively. This behavior is the default behavior.

When the requirement is that the Pending OCs should not be flushed, the new AVP, Override-Control-Pending-Queue-Action, is sent within at least one of the Override-Control AVPs in the RAR/CCA message, as shown in the following figure.

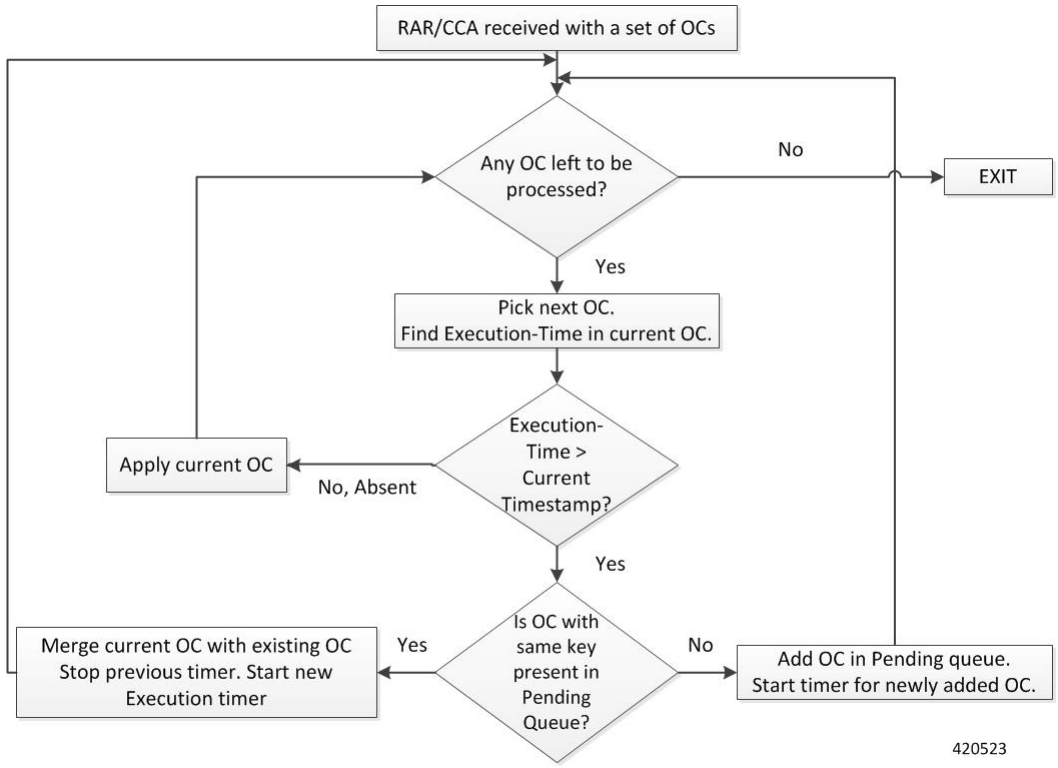


Adding OC in the Pending Queue

The PCEF processes all the instances of Override-Control grouped AVPs received from PCRF in RAR/CCA message. If an Override-Control AVP has an Execution-Time AVP with “Future” time stamp, the PCEF marks it “Pending” for that subscriber. If an OC is received with Execution-Time AVP and there is already an OC pending with same OC identifier, then the newly received OC is merged with the pending OC. The Execution-Time of the merged OC is that of the newly received OC.

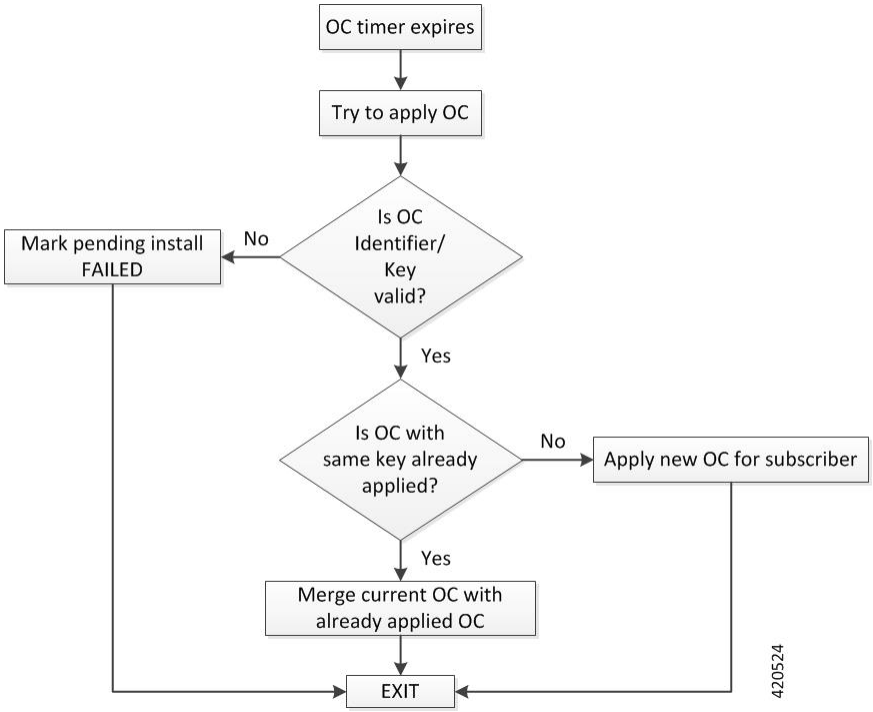
The Override-Control grouped AVP received without the Execution-Time AVP, or with Execution-Time already in “Past”, is not added in the list; in other words, it is applied immediately.

The following figure describes the adding of OC in Pending Queue.



Installing the Pending OC

The PCEF also stores the Execution-Time of all such Pending OCs so that, when the Execution-Time of an OC is reached, the OC is applied to the subscriber as described in the following figure.



Limitations

Following are the known limitations and restrictions of the Support for Execution-Time AVP feature:

- The Execution-Time AVP is supported only for OC without name.
- The maximum time for which an OC can be buffered is 44 days.
- There is no hard limit on number of OCs that can be buffered for a subscriber. However, the decision to buffer an OC depends on the availability of the system resources.
- The OCs with same identifier and different future Execution-Time is merged in the order in which PCEF processes them, and may not be same as the order of occurrence in RAR/CCA. The resultant OC has the Execution-Time of the OC which gets processed at the end.

Configuring Override Control

This section describes how to configure the Override Control feature to override charging and policy parameters for all rules (wildcard) or a specified set of rules or charging actions.

Important Override Control is a license-controlled feature. A valid feature license must be installed prior to configuring this feature. Contact your Cisco account representative for more information.

Use the following configuration to configure the Override Control feature at rulebase level:

Important In this release, both Inheritance and the Override Control features are supported. Note that these two features should not be enabled simultaneously. If by mistake, these features are enabled, only Override Control is applied.

```
configure
  active-charging service  service_name
    rulebase  rulebase_name
      [  default | no ] override-control [ with-oc-name ]
      end
```

Notes:

- The **override-control** CLI command will be visible only when the license to configure the Override Control feature is installed.
- By default, this feature is disabled. If this command is configured, the Override Control feature will be enabled. When enabled, it is necessary to turn off the Inheritance feature.
- The **with-oc-name** optional keyword specifies to use OC-name as the unique key to identify an OC for the session. If **with-oc-name** option is not configured in rulebase, OC will be identified using the Rule/CA and exclude rule as keys. This is the default behavior.

For more information on this command, see the *Command Line Interface Reference*.

Verifying the Override Control Configuration

To verify your configuration, in the Exec Mode, enter the following command:

show active-charging subscribers callid *<callid>* **override-control**

To verify the Override Control status, in the Exec Mode, enter the following command. The status displays ON when OC is installed and OFF when OC is removed.

show active-charging sessions all

Monitoring and Troubleshooting the Override Control feature

This section provides information on the show commands available to support this feature.

show active-charging rulebase statistics name <rulebase_name>

The output of this CLI command has been modified to show information related to pending OCs at rulebase-level. Following is a partial sample output:

```
show active-charging rulebase statistics name cisco

    Override Control Statistics:
       Total number of Installs Received:                6
       Total number of Installs Succeeded:               1
       Total number of Installs Failed:                  1
       Install Pending:
         Total:                                          4
         Merged:                                         1
         Flushed:                                        1
         Failed:                                         1
       Total number of Disables Received:                0
       Total number of Disables Succeeded:               0
       Total number of Disables Failed:                  0
       Total number of Subscribers:                      1
```

show active-charging service all

The following fields are newly added to the output of this show command:

- Override Control
  - Supported parameters
    - Charging Parameters
    - Policy Parameters

show active-charging sessions full all

The output of this show command is changed to indicate how many Overrides were received and how many are currently active for the subscriber. The following fields are new in this release:

- Override Control

- Installs Received
- Installs Succeeded
- Installs Failed

- Total Override Control

As part of Support for Execution-Time AVP feature, the output of this CLI command has been further modified to show information related to pending OCs at subscriber-level. Following is a partial sample output:

```
show active-charging sessions full all
.
.
.
Override Control:
    Installs Received:              2
    Installs Succeeded:             1  Installs Failed:             0
    Install Pending:
      Total  :                      2
      Merged :                      0
      Flushed:                      0
      Failed :                      0
    Disables Received:              0
    Disables Succeeded:             0  Disables Failed:             0

No Charging ruledef(s) match the specified criteria
No Firewall ruledef(s) match the specified criteria

  Post-processing Rulestats : No Post-processing ruledef(s) match the specified criteria
Dynamic Charging Rule Name Statistics: n/a
Total Dynamic Rules:             0
Total L7 Dynamic Rules:          0
Total Predefined Rules:          0
Total ADC Rules:                 0
Total Firewall Predefined Rules: 0
Total Override Control:          0
Total Override Control Pending:  3
```

show active-charging subscribers callid <callid> override-control

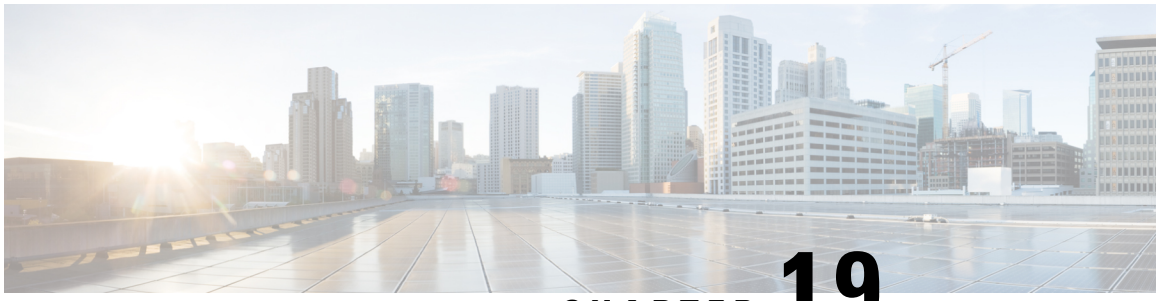
This command is added to display the override being applied for the subscriber.

show active-charging subscribers callid <call_id> override-control pending

As part of the Support for Execution-Time AVP feature, this new show command has been introduced to check the pending OCs at subscriber-level. Following is a sample output:

```
show active-charging subscribers callid 00004e21 override-control pending
CALLID: 00004e21
Override Control :
  Rule Name :
                     qci2
  Charging Parameters:
    Rating Group     : 100
    Offline Enabled  : TRUE
  Execution-Time     : <Day Month DD HH:MM:SS GMT YYYY>
Override Control :
  Rule Name :
                     qci1
```

```
Charging Parameters:
  Rating Group     : 100
  Offline Enabled  : TRUE
Policy Parameters:
  QCI              : 4
  ARP Byte         : 81
  MBR UL           : 25000
  MBR DL           : 13000
  TOS UL           : af23 (22)
  TOS DL           : af23 (22)
  NEXTHOP ADDR     : 10.20.10.10
  CF State       : 0
Execution-Time     : <Day Month DD HH:MM:SS GMT YYYY>
```



CHAPTER 19

Override Control Enhancement

This chapter describes the following topics:

Feature Summary and Revision History

Summary Data

Applicable Product(s) or Functional Area	• P-GW • S-GW • SAEGW
Applicable Platform(s)	• ASR 5500 • VPC - DI • VPC - SI
Feature Default	Disabled - Configuration Required
Related Changes in This Release	Not applicable
Related Documentation	• *ECS Administration Guide* • *Stats and Counters Reference Guide*

Revision History

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

Revision Details	Release
The Override Control (OC) feature, introduced in an earlier release, allowed you to dynamically modify the parameters of static or predefined rules with parameters sent by the PCRF over the Gx interface. This feature allows you to merge rule level or charging action level override control with Wildcard OC.	21.8
First introduced.	Pre 21.2

Feature Changes

The Override Control (OC) feature, introduced in an earlier release, allowed you to dynamically modify the parameters of static or predefined rules with parameters sent by the PCRF over the Gx interface. OC allows you to specify the overridden parameters with the ability to exclude certain rules.

The PCRF sends these overrides as Override-Control grouped AVP in a CCA or RAR message.

Currently, if a rule or charging-action level and Wildcard level OC parameters are received in a single message, then the rule level or charging-action level OC parameters are applied without merging any of the parameters with the Wildcard OC parameters.

A new Diameter AVP "Override-Control-Merge-Wildcard" is added to the grouped AVP "Override-Charging-Action-Parameters" and included in the 'dpca-custom8' dictionary. This AVP is required to indicate that an OC needs to be merged with a Wildcard OC.

On receiving this new AVP, the gateway merges the parameters of received OC with the Wildcard OC. The merged OC is applied to the rules matching rule-name/ca-name criteria of the received OC. If the Wildcard OC is not present, then the received OC is applied as it is.

Important While applying OC on the rules of a rulebase, if the rule is present in the Exclude-Rule list of Wildcard OC then for such rule, unmerged or original rule level or charging-action level OC is applied.

Monitoring and Troubleshooting

This section provides information on how to monitor and troubleshoot the Override Control Enhancement feature.

Show Commands and Outputs

This section provides information on show commands and their corresponding outputs for the Override Control Enhancement feature.

show active charging sessions full all

The following new fields are added to the output of this command:

- Merge with Wildcard
  - Received
  - Succeeded
  - Failed

show active-charging subscribers callid *callid_name* override-control

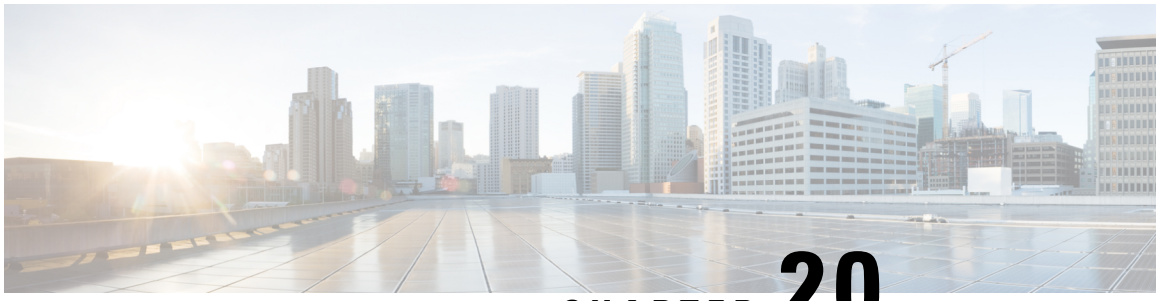
The following new fields are added to the output of this command:

Merge with Wildcard: TRUE/FALSE

show active-charging rulebase statistics name *statistics_name*

The following new fields are added to the output of this command:

- Merge with Wildcard
  - Received
  - Succeeded
  - Failed



CHAPTER 20

Override Control Support for Group-of-Ruledef

This chapter describes the following topics:

Feature Summary and Revision History

Summary Data

Applicable Product(s) or Functional Area	• GGSN • P-GW
Applicable Platform(s)	• ASR 5500 • VPC - Di • VPC - Si
Feature Default	Disabled - Configuration Required
Related Changes in This Release	Not Applicable
Related Documentation	• *Command Line Interface Reference* • *GGSN Administration Guide* • *P-GW Administration Guide*

Revision History

Important Revision history details are not provided for features introduced before releases 21.2 and N5.5.

Revision Details	Release
With this feature enhancement, to apply override control appropriately, override control subscriber map is identified by ruledef and group of ruledefs both. A new CLI command **align-with-gor** is added in rulebase for the same purpose.	21.3
First introduced.	Pre 21.2

Feature Changes

With this feature enhancement, to apply override control appropriately, ruledef, and group of ruledefs both identifies the override control subscriber map. A new CLI command **align-with-gor** has been added in rulebase for the same purpose.

Old Behavior:

1. Earlier, names of ruledefs present in the group-of-ruledefs were not supposed to be same as that of the independent ruledefs. Also, same ruledefs were not supposed to be part of two different group-of-ruledefs.
2. Exclusion of individual ruledef from override-control parameters was applied to the same ruledef under group-of-ruledefs.

New Behavior:

1. Now, overlapping of ruledef names across the group-of-ruledefs and standalone ruledefs is allowed.
2. Exclude-rule received in charging-action or wildcard level override control is applied only to the standalone ruledef or group-of-ruledefs. It is not applied to the rule present inside the GOR.

Configuring align-with-gor Override Control

To enable this feature enhancement, you must configure the **align-with-gor** along with the **override-control** CLI command in the rulebase. This CLI keyword when enabled, populates the override control subscriber map with the group information. If group-id is present, it is associated with the ruledef-id.

Once enabled, commit the feature by executing `update active-charging override-control rulebase-config`.

```
configure
    active-charging service <service_name>
       rulebase <rulebase_name>
         override-control [ align-with-gor | with-oc-name [ align-with-gor
 ] ]
           [ default | no ] override-control [ align-with-gor ]
           end
```

Notes:

- **default:** Configures this command with its default setting. By default, this feature is disabled.
- **no:** If previously enabled, disables override control in the current rulebase.
- **align-with-gor:** Resolves ambiguity when same ruledefs are defined in multiple Groups of Ruledefs.
- **with-oc-name:** Uses the override control name as unique key to identify override control for a session.

Upgrading and Downgrading Information

This section covers the upgrade and downgrade procedures.

Consider the following two configurations:

Configuration A:

GoR G1: R1, R2

GoR G2: R1', R2'

Configuration B:

GoR G1: R1, R2

GoR G2: R1, R2

Currently, configuration A is used to make sure that correct "Charging Action' is applied with application of the override control.

Upgrade Procedure:

Consider two chassis(Active Chassis and Standby Chassis) having StarOS 21.3 installed with configuration A. To upgrade the configuration from 21.3 to 21.4, perform the following steps:

1. Upgrade Standby to StarOS 21.4.
2. Perform ICSR switch over.
3. Upgrade the new Standby to StarOs 21.4.
4. Perform ICSR switch over.
5. Enable the feature CLI "override-control align-with-gor" in the rulebase.
6. Remove the 21.3 workaround, apply new configuration, where same ruledef can be part of multiple GOR, that is, Configuration B on Active and Standby chassis. For updating or deleting chassis ruledef, existing MOP procedure must be used. Note: If the override control was received for R1' and R2', it would not be applied as we remove R1' and R2'.
7. Commit the feature by executing the CLI command, "update active-charging override-control rulebase-config."
8. Wait for 20 minutes to ensure smooth transition.

Downgrade Procedure:

Consider two chassis(Active Chassis and Standby Chassis) having StarOS 21.4 installed with configuration B (after optimized configuration). To downgrade the configuration from 21.4 to 21.3, perform the following steps:

1. Change configuration on both the chassis to configuration A, where same ruledef are not part of the multiple GOR. For updating or deleting ruledef, existing MOP procedure must be used.
2. Disable the feature by executing the CLI command, "no override-control align-with-gor" in the rulebase.
3. Commit the new configuration by executing the CLI command "update active-charging override-control rulebase-config."
4. Wait for 20 minutes to ensure smooth transition.
5. Downgrade StarOS release 21.4 on Standby chassis to StarOS release 21.3.
6. Perform ICSR switch over.
7. Downgrade StarOS release 21.4 on new standby to StarOS release 21.3.
8. Perform ICSR switch over.

Feature turn on and turn off Procedure

This section covers steps to turn the feature ON and OFF.

Feature is ON and must be turned OFF.

Consider a scenario where the chassis has configuration B(after optimized configuration) and the feature CLI "override-control align-with-gor" is configured in the rulebase. When the feature is ON, to turn it OFF, perform the following steps:

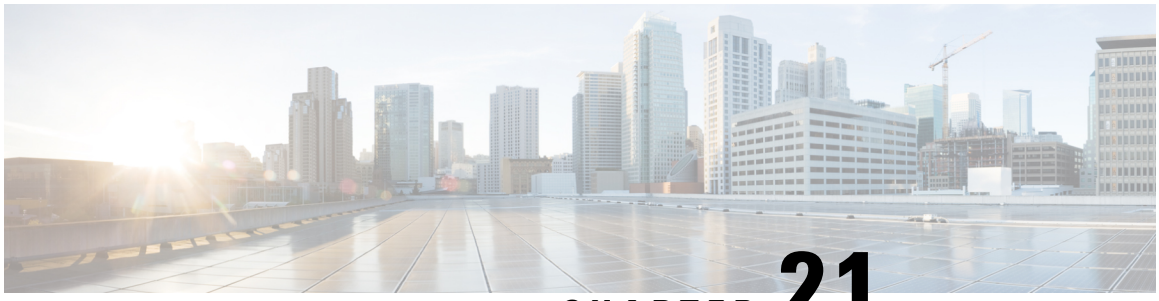
1. Change the configuration to Config A.
2. Verify if the configuration is aligned with the following:

   Ruledef R1 present in the rulebase is not present in any group-of-ruledefs, where the group-of-ruledefs must be disjoint.
3. Disable the feature using the CLI command, "no override-control align-with-gor" in rulebase.
4. Commit the new configuration by executing the CLI command "update active-charging override-control rulebase-config.
5. Wait for 20 minutes to ensure smooth transition.

Feature is OFF and must be turned ON.

Consider a scenario where the chassis has configuration A and the feature CLI "override-control align-with-gor" is configured in the rulebase. When the feature is OFF, to turn it ON, perform the following steps:

1. Remove the StarOS 21.3 workaround, wherein the new configuration, where the same ruledef can be part of multiple GOR is applied to both active and standby chassis. For updating or deleting ruledef, existing MOP procedure must be used. Note: If the override control was received for R1' and R2', it would not be applied as we remove R1' and R2'.
2. Commit te feature by executing the CLI command "update active-charging override-control rulebase-config."
3. Wait for 20 minutes to ensure smooth transition.



CHAPTER 21

Response-based Charging

This chapter describes the response-based Charging feature and provides detailed information on the following topics:

Feature Description

The Response-based Charging feature is introduced to classify data volumes for HTTP request with the content-id and service-id of the HTTP response. Charging of HTTP request packets will be deferred until the HTTP response packet arrives. Once the response is received, the request packet will be charged according to the service-id and content-id of the response.

- Response-based charging is supported only for the HTTP protocol. When a HTTP method is configured, it will be applicable to all HTTP transactions of that method type for the subscriber. There is no capability to selectively enable response-based charging for some HTTP transactions of a particular HTTP method type and not to other transactions of the same method for a subscriber.
- Response-based charging supports pipelined HTTP requests (both concatenated and non-concatenated). This will also support persistent HTTP connections. In case of pipelined HTTP requests of different HTTP methods, this feature will be applied only to those HTTP methods for which it is configured.
- Response-based Charging can be enabled for subscribers classified according to the APN, Virtual-APN, Rulebase, or a combination of these.
- A configurable option is provided to limit the behavior to HTTP methods configured for response-based charging. The **charge-request-to-response** CLI command is added in the ACS Trigger Action configuration mode to delay charging for a subscriber flow.

The Response-based Charging feature supports both offline and online charging modes.

- **Offline charging**: Response-based charging of HTTP request packet will be reflected in the offline charging records such as EDR, EGCDR, UDR, and RF. According to the billing method configured in the billing policy, the HTTP request bytes will be accounted for against the rating group/content-id of the HTTP response and will be reflected in the data record generated.

- **Online charging**: Online charging using Gy interface and Volume Reporting based on monitoring key via Gx interface will be supported. Response-based charging of HTTP request packet will be reflected in these charging methods. According to the billing method configured in the billing policy, the HTTP request bytes will be accounted for against the rating group/content-id of the HTTP response and will be reflected in the data volume reported.

The service-scheme framework configuration is required to configure and enable the Response-based Charging feature for a subscriber. This framework is introduced to disassociate the dependency with rulebase/PCRF and update the policies specific to subscribers based on pre-configured events. For more information on the service-scheme framework, see the *ECS Administration Guide*.

Relationships to Other Features

This section describes the interoperability of the Response-based Charging feature with other ECS features.

- Delay charging of control packets:
  - Charge-to-application option: This option to delay charging of control packets delays rulematching and charging of all TCP control packets or subsets of it, depending on the configuration. These packets are rulematched and charged according to the first application packet of the HTTP flow. This feature and all its variants will continue to work as is when response-based charging is not configured. When response-based charging is configured, these packets will be rulematched according to the first HTTP request packet but will be charged according to the HTTP response.
  - Charge-separate-from-application option: This option to delay charging of control packets delays rule matching and charging of TCP control packets and charges them to an L7/L4/L3 rule at the end of the flow. This feature will continue to work as is when response-based charging is configured or not.
- Websocket: This feature involves charging subsequent packets of the flow after HTTP GET request as per the HTTP request, if the HTTP flow is upgraded to be a websocket flow. When Response-based charging is configured for HTTP GET method, the HTTP GET request and all subsequent packets are charged according to the HTTP response, if the flow is upgraded to a websocket flow.
- Transactional Rule Matching (TRM): When TRM is enabled for a subscriber, TRM gets engaged on the HTTP request packet and rule match for subsequent packets is bypassed. These packets are charged according to the content-id and service-id of the HTTP request. Since rule match is bypassed for HTTP response, response-based charging will not be supported with TRM.
- Response-based TRM: When Response-based TRM is enabled for a subscriber, TRM gets engaged on the HTTP response packet and rule match for subsequent packets is bypassed. Response-based charging will be supported with Response-based TRM.
- HTTPS, RTSP, RTCP and WSP analyzers: Response-based charging is not supported for these analyzers. This feature is supported only for the HTTP protocol.

Limitations for Response-based Charging

The HTTP request packet will be sent towards the server but charging will be deferred. On receiving a response, the following charging methods will be implemented based on the treatment for the response.

- Response packet gets dropped in ECS (drop due to bandwidth limiting, flow actions, etc.):

The request packet will be charged according to the service-id, content-id, and rating-group of the HTTP response. If delay charging of control packets is enabled for the subscriber, the TCP TWH (three-way-handshake) control packets along with HTTP request will be charged according to the service-id, content-id and rating-group of HTTP response.

- Response packet is lost in transit and not received at ECS:

  If the request packet gets retransmitted by the client and a response is received, the original and retransmitted request packets will be charged according to the response. If the response is not received even after retransmissions, then the parent TCP flow will timeout. In this case, the request packets for which charging is still pending will be rule matched again and charged according to the L3/L4 rule. If delay charging of control packets is enabled for the subscriber, the TCP TWH control packets along with HTTP request will be rule matched again and charged according to the L3/L4 rule that it matches.

How It Works

This section describes the Response-based Charging configuration. The Service Scheme framework configuration is required to configure and enable this feature for a subscriber.

Use the sample configuration to enable the response-based Charging feature for a subscriber on a particular rulebase.

```
configure
  active-charging service s1
    trigger-action ta1
      charge-request-to-response http all
    #exit
    trigger-condition tc1
      any-match = TRUE
    #exit
    service-scheme ss1
       trigger sess-setup
         priority 1 trigger-condition tc1  trigger-action ta1
      #exit
    #exit
    subs-class sc1
      rulebase = rb1
    #exit
    subscriber-base sb1
       priority 1 subs-class sc1  bind service-scheme ss1
    #exit
```

Notes:

- If **transactional-rule-matching** is configured in the rulebase and response-based charging is also configured in the trigger-action for a subscriber, then response-based TRM must be configured in the same trigger-action for response-based charging to be functional.
- If **transactional-rule-matching** is not configured in the rulebase and response-based charging is configured in the trigger-action for a subscriber, then it is optional to configure response-based TRM in the same trigger-action. Response-based charging will be functional with or without the configuration of response-based TRM in the trigger-action.

Configuring Response-based Charging

Use the following configuration in the ACS Trigger Action Configuration mode to allow operators to charge the HTTP request packets based on the specified HTTP response received.

```
configure
  active-charging service <service_name>
    trigger-action <trigger_action_name>
      [ no ] charge-request-to-response http { all | connect | delete |
 get | head | options | post | put | trace }
      exit
```

Notes:

- To disable the feature for the subscriber, configure the following command:

  no charge-request-to-response http all

- The response-based charging feature applies to the following HTTP methods:
  - all - Applies to all HTTP methods
  - connect - HTTP Connect method
  - delete - HTTP Delete method
  - get - HTTP Get method
  - head - HTTP Head method
  - options - HTTP Options method
  - post - HTTP Post method
  - put - HTTP Put method
  - trace - HTTP Trace method

Verifying the Response-based Charging Configuration

Enter the following command to check if the response-based charging feature is applied to the different HTTP transactions based on HTTP request methods:

show active-charging analyzer statistics name http

Monitoring and Troubleshooting the Response-based Charging feature

This section provides information on the show commands available to support this feature.

show active-charging analyzer statistics name http

The following fields display the count per HTTP method that has response-based charging applied.

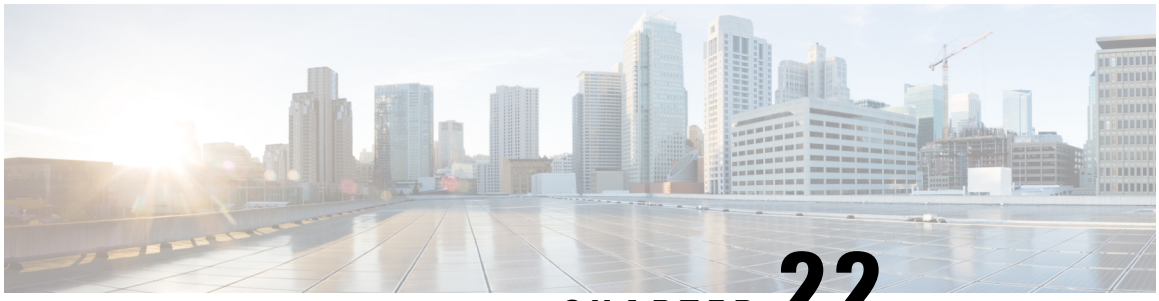
A sample output is shown below.

```
Response Based Charging:
 GET       1    POST    0
CONNECT    0    PUT     0
HEAD       0    OPTION  0
DELETE     0    TRACE   0
Websocket  0
```

show active-charging trigger-action all

The following fields displays the specified HTTP method(s) that has response-based charging applied. This field displays "all" if all HTTP methods are configured and "none" if no HTTP method is configured.

- HTTP Response Based Charging



CHAPTER 22

Response-based TRM

This chapter describes the response-based Transactional Rule Matching (TRM) feature and provides detailed information on the following topics:

Feature Description

The Transactional Rule Matching (TRM) feature is enhanced to support response-based TRM for HTTP protocol. This feature is applicable to all HTTP transactions of a method type for the subscriber, when an HTTP method is configured for response-based TRM. A configurable option is provided to limit the behavior to HTTP methods configured for response-based TRM.

TRM gets engaged on the HTTP request packet and all further packets in the HTTP transaction bypass rule matching. The same rule is matched and charging-action is applied as that of the HTTP request.

With this release, a new variant of TRM is introduced, where engagement of TRM can be delayed till HTTP response is received. When response-based TRM is configured for a subscriber, TRM gets engaged on the first complete HTTP response packet. The same rule and charging action is applied to all further packets in the HTTP transaction.

- Response-based TRM is supported only for the HTTP protocol. There is no capability to selectively enable response-based TRM for some HTTP transactions of a particular HTTP method type and not to other transactions of the same method for a subscriber.
- Response-based TRM will support pipelined HTTP requests (both concatenated and non-concatenated). This will also support persistent HTTP connections. For pipelined HTTP requests of different HTTP methods, this feature will be applied only to those HTTP methods for which it is configured.
- Response-based TRM can be enabled for subscribers classified according to the APN, Virtual-APN, Rulebase, or a combination of these.
- The **transactional-rule-matching response** command is added in the ACS Trigger Action configuration mode to delay engagement of TRM for the flow.
- The **show active-charging analyzer statistics name http** command displays the count if the feature has been applied to the different HTTP transactions based on HTTP request methods.

The service-scheme framework configuration is required to configure and enable the Response-based Charging feature for a subscriber. This framework is introduced to disassociate the dependency with rulebase/PCRF and update the policies specific to subscribers based on pre-configured events. For more information on the service-scheme framework, see the *ECS Administration Guide*.

Overview

The response-based TRM feature enables inspection of the HTTP response packets while still deriving the performance benefit of TRM. Based on the content in the HTTP response, free-rating or charging can be applied to all subsequent packets in the HTTP transaction. TRM is engaged on the response packet and rule-match can be bypassed for all subsequent packets in the HTTP transaction.

Relationships to Other Features

This section describes the interoperability of the Response-based TRM feature with other ECS features.

1. Delay charging of control packets:
   - Charge-to-application option: This option of delay charging of control packets delays rule matching and charging of all TCP control packets or subsets of it, depending on the configuration. This feature will continue to work with response-based TRM in the same way as is with existing TRM.
   - Charge-separate-from-application option: This option to delay charging of control packets delays rule matching and charging of TCP control packets and charges them to an L7/L4/L3 rule at the end of the flow. This feature will continue to work with response-based TRM in the same way as is with existing TRM.
2. Websocket: This feature involves charging subsequent packets of the flow after HTTP GET request as per the HTTP request, if the HTTP flow is upgraded to be a websocket flow. When Response-based TRM is configured for HTTP GET method, the HTTP response packet is rule matched and rule matching is bypassed for subsequent packets till the end of the flow, if the flow is upgraded to a websocket flow. If flow does not get upgraded to websocket flow, then rule match is bypassed for subsequent packets till the end of the GET transaction.
3. HTTPS, RTSP, RTCP, WSP analyzers: Response-based TRM is not supported for these analyzers. This is supported only for the HTTP protocol.
4. Transactional Rule Matching (TRM): The TRM-specific CLI command must be configured at the rulebase level for response-based TRM to be functional. When both TRM and response-based TRM are configured for a subscriber, response-based TRM will take precedence.
5. Response-based Charging: When Response-based charging is enabled for a subscriber, HTTP request packets will be charged according to the HTTP response packet. Response-based charging will be supported with Response-based TRM.

How It Works

This section describes the Response-based TRM configuration. The Service Scheme framework configuration is required to configure and enable this feature for a subscriber.

Use the sample configuration to enable the response-based TRM feature for a subscriber on a particular rulebase.

```
configure
  active-charging service s1
    trigger-action ta1
      transactional-rule-matching response http all
    #exit
    trigger-condition tc1
      any-match = TRUE
    #exit
    service-scheme ss1
      trigger sess-setup
        priority 1 trigger-condition tc1 trigger-action ta1
      #exit
    #exit
    subs-class sc1
      rulebase = rb1
    #exit
    subscriber-base sb1
      priority 1 subs-class sc1 bind service-scheme ss1
    #exit
```

Notes:

- The **transactional-rule-matching** command configured in the rulebase that handles the subscriber for response-based TRM must be functional.
- If the **transactional-rule-matching** command is not configured in the rulebase for a subscriber, response-based TRM will not be applicable to the subscriber.

Configuring Response-based TRM

Use the following configuration in the ACS Trigger Action Configuration mode to delay engagement of TRM for the flow.

```
configure
  active-charging service service_name
    trigger-action trigger_action_name
      [ no ] transactional-rule-matching response http { all | connect
| delete | get | head | options | post | put | trace }
      exit
```

- To disable the feature for the subscriber, configure the following command:

  no transactional-rule-matching response http all

- The response-based TRM feature applies to the following HTTP methods:
  - all - Applies to all HTTP methods
  - connect - HTTP Connect method
  - delete - HTTP Delete method

- get - HTTP Get method
- head - HTTP Head method
- options - HTTP Options method
- post - HTTP Post method
- put - HTTP Put method
- trace - HTTP Trace method

Verifying the Response-based TRM Configuration

Enter the following command to check if the response-based TRM feature is applied to the different HTTP transactions based on HTTP request methods:

show active-charging analyzer statistics name http

Monitoring and Troubleshooting the Response-based TRM feature

This section provides information on the show commands available to support this feature.

show active-charging analyzer statistics name http

The following fields display the count per HTTP method that has response-based TRM applied.

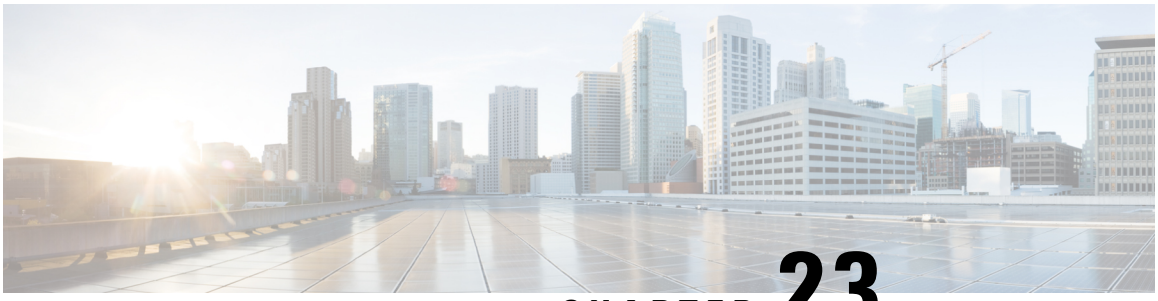
A sample output is shown below.

```
Response Based TRM:
 GET      1    POST   0
CONNECT   0    PUT    0
HEAD      0    OPTION 0
DELETE    0    TRACE  0
Websocket 0
```

show active-charging trigger-action all

The following field displays the specified HTTP method(s) that has response-based TRM applied. This field displays "all" if all HTTP methods are configured and "none" if no HTTP method is configured.

- HTTP Response Based TRM



CHAPTER 23

SNMP Trap Support for Ruledef and Rulebase

Feature Summary and Revision History

Summary Data

Applicable Product(s) and Functional Area	• GGSN • P-GW
Applicable Platform(s)	• ASR 5500 • VPC - DI • VPC - SI
Feature Default	Disabled - Configuration Required
Related Changes in This Release	Not Applicable
Related Documentation	• *Command Line Interface Reference* • *ECS Administration Guide*

Revision History

Revision Details	Release
First introduced.	21.3

Feature Description

During back-to-back ICSR switchovers there are instances where a number of subscribers lose the assigned dynamic charging rules. In such cases traffic starts hitting static catchall rules, which not be charged. To track this loss, an SNMP trap is sent when traffic hit these rules. This timely tracking ensures accurate charging.

To ensure that there is no loss of dynamic charging rules, new CLIs are introduced to configure the threshold value, volume monitoring duration, and enable threshold volume for total-volume.

The threshold can be configured for a combination of rulebase and ruledef or rulebase and group-of-ruledef. This threshold is monitored across all subscribers of the system. The SNMP trap is generated when traffic reaches configured threshold value with severity information.

The SNMP trap contains information about the rulebase-name, ruledef-name or group-of-ruledef name, and volume in bytes.

Configuration and Restrictions

Following are the configuration and restrictions that are applicable to this feature:

- The CLI configurations are applicable only for static ruledefs and static group-of-ruledefs.
- P-GW and GGSN service name is not included in the trap information.
- There is no support for Session Recovery and ICSR. These per rulebase per ruledef data statistics are not check-pointed during session recovery and switchover.
- There is a single trap per rulebase and ruledef configuration per system. The SNMP trap is not per subscriber.
- The Operator can configure a maximum of 30 rulebase and ruledefs/group-of-ruledefs to monitor volume thresholds.
- Ruledef, group-of-ruledef, and Rulebase for which threshold monitoring is enabled should be configured in active-charging service configuration. Else trap is not be generated for these.
- Threshold value for total-volume is configured in bytes. The total-volume data sent towards THE external server for a particular monitoring interval is reported in kilobytes.
- As measured data forwarded to the external server is always in kilobytes, data measured in range of 1 to 999 bytes is rounded off to 1 kilobyte. Actual measured data in such cases can be checked using the **show snmp trap history** command.

Configuring the SNMP Trap support for Ruledef and Rulebase Feature

The following section provides the configuration commands in support of this feature.

threshold total-volume rulebase

The following new CLI command is added to configure or delete the threshold value of the total volume for rulebase and ruledef.

This CLI is disabled by default.

To configure the threshold value:

```
configure
        threshold total-volume rulebase rulebase_name { ruledef ruledef_name
 | group-of-ruledef gor_name } clear
        end
```

To delete the threshold value:

```
configure
        default threshold total-volume rulebase rulebase-name { ruledef
ruledef_name | group-of-ruledef gor_name }
        end
```

Notes:

- **default**: Deletes the specified threshold value.
- **total-volume**: Configures total volume amount threshold.
- **rulebase** *rulebase-name*: Configures rulebase for which threshold is monitored. For rulebase name, enter a string of size 1 to 63.
- **ruledef** *ruledef-name*: Configures ruledef for which threshold is monitored. For ruledef name, enter a string of size 1 to 63.
- **group-of-ruledef**: Configures a group-of-ruledef for which threshold is monitored.
- **threshold value for total-volume**: Enter an integer from 1 to 1000000000.
- **clear**: Configures the alarm clear threshold.

threshold poll total-volume interval

This new CLI command is added to configure the volume monitoring window duration for which to check the threshold.

This CLI is disabled by default.

To configure the threshold poll interval:

```
configure
        threshold poll total interval poll_interval
        end
```

To delete the threshold poll interval:

```
configure
        default threshold poll total interval poll_interval
        end
```

Notes:

- **default**: Configures this command with the default threshold setting. The default value is 15 minutes.
- **total-volume**: Configures total-volume threshold interval.
- **threshold poll total-volume interval** : Enter polling interval in second in the range of 300 to 14400 seconds.
- **default threshold poll total-volume**: Configures the default value. The supported default value supported is 15 minutes.

threshold monitoring total-volume

This new CLI command is added to enable or disable the threshold monitoring for the total volume.

This CLI is disabled by default.

To enable threshold monitoring for the total volume:

```
configure
        threshold monitoring total-volume
        end
```

To disable threshold monitoring for the total volume:

```
configure
        no threshold monitoring total-volume
        end
```

Notes:

- **no**: Disables the total-volume related threshold.
- **threshold monitoring total-volume** : Enables the total-volume related threshold.

Monitoring and Troubleshooting

This section provides information regarding show commands and/or their outputs in support of this feature.

Show Commands

This section lists all the show commands available to check the threshold configured and the trap output..

show snmp trap history

This command displays the following output:

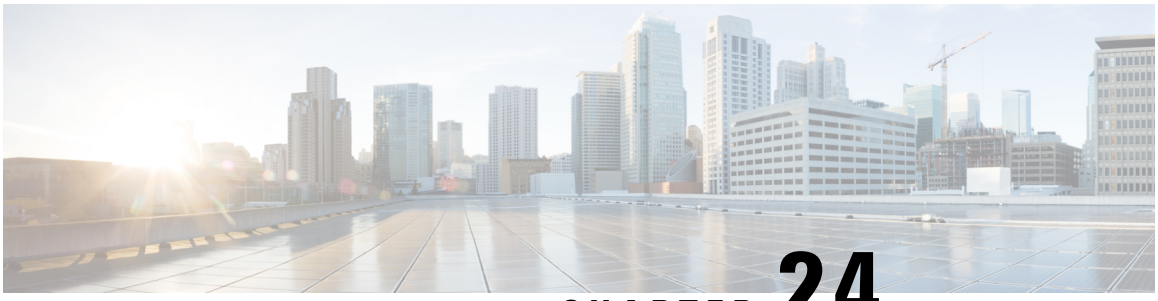
```
Tue Jun 27 00:20:00 2017 Internal trap notification 1363 (TotalVolume) facility sessmgr
rulebase 'plan1' ruledef 'ip-all' threshold 200 measured value 520

Tue Jun 27 03:05:00 2017 Internal trap notification 1364 (TotalVolumeClear) facility sessmgr
 rulebase 'plan1' ruledef 'ip-all' threshold 1 measured value 0

Wed Jun 28 10:30:00 2017 Internal trap notification 1363 (TotalVolume) facility sessmgr
rulebase 'plan1' group-of-ruledef 'GOR' threshold 100 measured value 8053
```

```
Wed Jun 28 10:35:00 2017 Internal trap notification 1364 (TotalVolumeClear) facility sessmgr
 rulebase 'plan1' group-of-ruledef 'GOR' threshold 0 measured value 0
```

show threshold

This command displays the following output:

```
show threshold
Threshold operation model: ALARM
Configured thresholds:
Name:             total-volume
        Config Scope:     RuleBase[plan1]Ruledef[ip-all]
        Threshold:        200
        Clear Threshold:  0
Active thresholds:
Name:             total-volume
        Config Scope:     RuleBase[plan1]Ruledef[ip-all]
        Threshold:        200
        Clear Threshold:  0
        Poll Interval:    300Seconds
        Next Poll Time:   2017-Jun-27+00:15:00
Enabled threshold groups: (name, scope)
        total-volume           SYSTEM
Non-default poll intervals:
        total-volume                300Sec
No outstanding alarm
```



CHAPTER 24

Support for Interim EDRs

Feature Summary and Revision History

Summary Data

Applicable Product(s) or Functional Area	• P-GW • SAE-GW
Applicable Platform(s)	• ASR 5500 • VPC - DI • VPC - SI
Default Setting	Disabled - Configuration Required
Related Changes in This Release	Not Applicable
Related Documentation	• *Command Line Interface Reference* • *ECS Administration Guide* • *Statistics and Counters Reference*

Revision History

Revision Details	Release
First introduced.	21.9.4

Feature Changes

ECS supports generation of Interim EDRs – EDRs that are generated for ongoing flows based on a configurable timer.

Usually, EDRs are generated for flows only when the flow terminates or when the flow reaches the configured flow idle-timeout value. These flows could have time duration that is as long as 48 hours, which makes it difficult to track subscriber activity until an EDR is generated.

Thus, with interim EDRs, ongoing flow activities can be tracked by configuring an interim timeout value for a flow. On expiration of the interim timer, an EDR is generated.

For configuring an interim EDR, a new condition – **interim** is CLI configurable. Based on the configuration, the interim timer is applied to newly-created flows. On expiration of the timer, an interim EDR is generated along with the following reason: **sn-closure-reason (23)**. The information volume available until the expiration of the timer is populated in the EDR along with its respective timestamps.

The Interim EDR functionality is license controlled. 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 the *Software Management Operations* chapter in the *System Administration Guide*.

Command Changes

flow end-conditon

In the *ACS Rulebase Configuration* mode, the **flow end-condition** command supports the **interim** condition. On configuring this condition, the interim timer is initiated for newly-created flows. An interim EDR is generated on expiration of the timer.

Use the following configuration to enable generation of interim EDRs.

```
configure
   require active-charging
   active-charging service service_name
      rulebase rulebase_name
         flow end-condition interim interim_timer_value charging-edr
charging_edr_format_name
         end
```

NOTES:

- **flow**: Specifies a flow related to the user session.
- **end-condition**: Specifies the end condition of the flow related to a user session that triggers EDR generation.
- **interim**: This condition specifies the interim threshold condition of the flow where an EDR is generated based on the configured timer value. The *interim_timer_value* is configured in minutes with a configurable range from 15 to 1440 minutes.

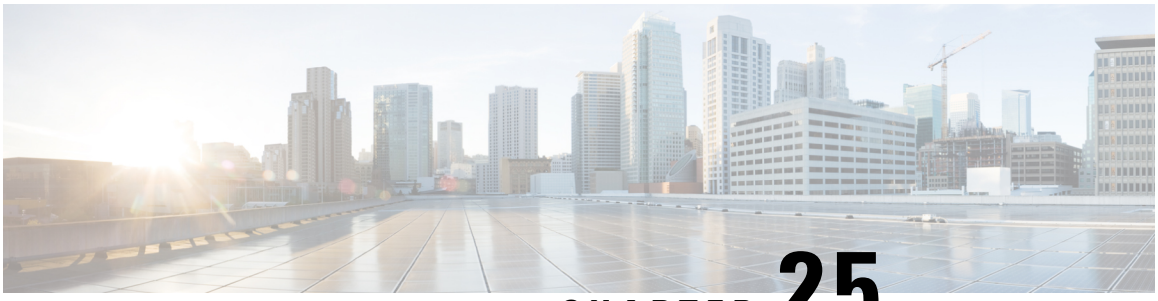
- **charging-edr**: Specifies the charging EDR format that is used to generate the EDRs. *charging_edr_format_name* must be the name of a charging EDR format, and must be an alphanumeric string of 1 through 63 characters.
- The **interim** keyword is only applicable for new flows created and not on existing flows.

Performance Indicator Changes

show active-charging rulebase statistics name *name*

The output of this command includes the following Interim EDR related fields:

- EDRs generated for interim
  - Interval



CHAPTER 25

Tethering Detection

This chapter describes the Tethering Detection feature and provides detailed information on the following topics:

Feature Description

This section provides an overview of the Tethering Detection feature.

Important In this release, the Tethering Detection feature is supported only on the GGSN, HA, and P-GW.

Tethering refers to the use of a mobile smartphone as a USB dongle/modem to provide Internet connectivity to PC devices (laptops, PDAs, tablets, and so on) running on the smartphone's data plan. Typically, for smartphone users, most operators have in place an unlimited data plan, the usage of which is intended to be from the smartphone as a mobile device. However, some subscribers use the low cost / unlimited usage data plan to provide Internet connectivity to their laptops in places where normal Internet connection via broadband/WiFi may be costly, unavailable, or insecure.

The Tethering Detection feature enables detection of subscriber data traffic originating from PC devices tethered to mobile smartphones, and also provides effective reporting to enable service providers take business decisions on how to manage such usage and to bill subscribers accordingly. Tethering Detection is supported for IPv4 (TCP)and IPv6 traffic flows.

The Tethering Detection feature is enabled on a per rulebase basis. The rulebase (billing plan) assigned for APN will contain the tethering detection related configuration. ECS performs tethering detection on a per flow basis for all subscribers (for whom TAC database match succeeded) using an APN in which the feature is enabled. The extent to which the detection mechanism is executed depends on the type of flow. If it is a non-TCP flow, for example UDP or ICMP, then tethering detection is not possible for the same, in releases prior to 18.2.

ECS supports various tethering detection solutions:

- TTL-based tethering detection

- UA-based tethering detection
- OS-based tethering detection

In 18.2 and later releases, the IP-TTL based tethering detection solution is implemented to support tethering for all IP flows - both IPv4 and IPv6. This feature is configurable at the rulebase level and will be done for all flows of all subscribers having IP-TTL configuration within the rulebase.

Important In release 18.2, IPv6 tethering detection is supported with only TTL and UA signatures, and not supported for OS signatures. In 18.4 and later releases, IPv6 OS-based tethering detection is supported.

In 18.4 and later releases, IPv6 Tethering Detection is supported for OS-based signatures. If the signature format used for IPv6 OS based tethering detection needs to be modified, additional data must be collected to identify fields of the new signature. The following new TCP parameters are added to EDRs:

- IPv6 OS signature string
- Sequence number from the TCP SYN packet of a flow
- 8 bytes of control parameters that include data offset, reserved, flags, window size, checksum and urgent pointer from TCP SYN header
- TCP options, if they are present in TCP SYN header

License Requirements

Tethering Detection 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 the *Software Management Operations* chapter in the *System Administration Guide*.

Flow Recovery Support for Tethering Detection

Feature Information

Summary Data

Status	New Functionality
Introduced-In Release	21.2
Modified-In Release(s)	Not Applicable
Applicable Product(s)	GGSN, HA, P-GW
Applicable Platform(s)	ASR 5500
Default Setting	Enabled. Flow recovery configurations are inherited.
Related CDETS ID(s)	CSCvc80454

Related Changes in This Release	Not Applicable
Related Documentation	ECS Administration Guide Statistics and Counters Reference Statistics and Counters Reference - Counter Descriptions

Revision History

Important Revision history details are not provided for features introduced before release 21.2.

Revision Details	Release	Release Date
New in this release.	21.2	April 27, 2017

Feature Changes

This feature introduces recovery support for the flows which have tethering detection enabled for them. For this, the flow recovery uses the service scheme framework to support the recovery. The recovered information helps in post SR and ICSR and continue marking tethered flows.

Previous Behavior: Flow recovery did not recover the tethering specific parameters.

New Behavior: Flow recovery now recovers tethering specific parameters in addition to existing Flow Recovery behavior.

Customer Impact: Post recovery, tethered flows might have differential charging and tethering might have visibility in the EDR.

How it Works

This section describes the working of "Recovery of tethered flag" and "Recovery of parameters for tethering the EDRs".

Recovery of Tethered Flag

This section lists the steps that occur during recovery of tethered flag.

1. The following checkpoint tethering-detection-method-types are used, by which flow is detected as tethered:
   - IP-TTL-tethered
   - OS-tethered-flag/UA-tethered
   - DNS-based-tethered
2. Check point is done as before for all flows that qualify the flow recovery trigger condition. With this enhancement, tethering flags and EDR values are also checkpointed for such flows, additionally.
3. Piggyback above tethering-recovery-info on the existing flow recovery information is done, that would be happening for trigger-type flow-create and the given trigger condition.
4. Check point is triggered when trigger condition is met (and corresponding action is flow recovery) .

5. On the first packet received, the flags are restored after recovery.

Important Post-flow-recovery UA-based-TD is not supported as per flow-recovery-framework limitations.

Recovery of Tethering EDR Parameters

This section lists the steps that occur during recovery of parameters for tethering the EDRs:

1. The following tethering specific EDR parameters are recovered for a tethered flow:
   - IP-TTL value
   - IPv4 OS signature
   - IPv6 OS signature
   - Signature-scan-status
2. Check point is done as before for all flows that qualify the flow recovery trigger condition. With this enhancement, tethering flags and EDR values are also checkpointed for such flows, additionally.
3. Piggyback above tethering-recovery-info on the existing flow recovery information is done, that would be happening for trigger-type flow-create and the given trigger condition.
4. Check point is triggered when trigger condition is met (and corresponding action is flow recovery) .
5. On the first packet received, the flags are restored after recovery.

Important Recovery support for multiple tethering detection EDR OS signatures in the same flow is not supported.

Sample Configuration

This section lists the sample configuration for "Recover IP-TTL based tethering detection" and "EDR Configuration".

Sample Configuration to Recover IP-TTL Based Tethering-Detection

```
configure
 active-charging service s1
  subscriber-base default
    priority 10 subs-class class1 bind service-scheme flow_recovery
    #priority 20 subs-class class2 bind service-scheme scheme2
  #exit

  subs-class class1
    apn = starent.com
    rulebase = plan1
    v-apn != some_virtual_apn
    multi-line-or all-lines
```

```
#exit

  rulebase plan1
    action priority 1 ruledef tethered_flow charging-action standard
    tethering-detection ip-ttl value 62
  #exit

  ruledef tethered_flow
    tethering-detection ip-ttl flow-tethered
    ip any-match = TRUE
  #exit

  service-scheme flow_recovery
    trigger flow-create
    priority 1 trigger-condition flow_rec_cond trigger-action
tether_flow_recov
  #exit

  trigger-condition flow_rec_cond
    rule-name = tethered_flow
    #any-match = TRUE
#exit

  trigger-action tether_flow_recov
    flow-recovery
  #exit
end
```

Sample EDR Configuration

```
conf
act s s1
rulebase plan1
billing-records egcdr
flow end-condition normal-end-signaling edr edr_td
end

conf
act s s1
edr-format edr_td
rule-variable flow ttl priority 1
rule-variable tcp os-signature priority 2
rule-variable tcp v6-os-signature priority 3
end
```

Monitoring and Troubleshooting

This section lists the commands available to monitor the "Flow Recovery Support for Tethering Detection" feature.

Show Commands

This section lists all the show commands available to monitor this feature.

show active-charging tethering-detection statistics

This command has been modified to display the following output:

```
Current Tethered Subscribers:                    0
    Total flows scanned:                         0
    Total Tethered flows detected:               0
    Total Tethered flows recovered:              1
    Total flows bypassed for scanning :          0

Tethering Detection Statistics (os-ua):
   TAC ID lookups:                               0
   TAC ID matches:                               0
   OS signature lookups:                         0
   OS signature matches:                         0
   IPv6 OS signature lookups:                    0
   IPv6 OS signature matches:                    0
   UA signature lookups:                         0
   UA signature matches:                         0
   Total flows scanned:                          0
   Tethered flows detected:                      0
   Non-tethered flows detected:                  0
   Tethered Uplink Packets:                      0
   Tethered Downlink Packets:                    0
   Current tethering-detected indications sent:  0
   Total tethering-detected indications sent:    0

Tethering Detection Statistics (ip-ttl):
   Total flows scanned:                          0
   Tethered flows detected:                      0
   Tethered uplink packets:                      0
   Tethered downlink packets:                    0

Tethering Detection Statistics (DNS Based):
   Total flows scanned:                          0
   Tethered flows detected:                      0
   Tethered uplink packets:                      1
   Tethered downlink packets:                    1

Change Statistics for Multiple SYN in Flow:
   Tethered to Non-Tethered:                     0
   Non-Tethered to Tethered:                     0
   Tethered to Tethered:                         0
   Non-Tethered to Non-Tethered:                 0
```

Bulk Statistics

This section lists all the bulk statistics that have been added, modified, or deprecated to support this feature.

ECS Schema

This section displays the bulk stats that have been added for total tethered recovered flows:

- ecs-td-total-recovered-flows - Total number of recovered tethered flows

IPv6 DNS-based Tethering Detection

The Tethering Detection feature is enhanced to detect DNS-based tethering for IPv6 flows. This feature can be used to detect tethering for users using stealth tethering applications such as FoxFi, EasyTether, and so on.

The DNS-based solution is implemented to address the deficiencies that other incumbent solutions had in detecting IPv6 tethered flows. Tethering detection for IPv6 traffic was achieved using either IPv6 IP-TTL based tethering detection or IPv6 OS-based tethering detection solution.

Mid-flow Tethering Detection

The Tethering Detection feature supports mid-flow tethering detection and EDR/REDR generation on tethering signature change. IP-TTL and OS-based tethering detection will analyze mid-flow SYN to generate IP-TTL/OS signature. When this feature is enabled, OS-based tethering detection keeps the OS-signature value of the last SYN packet and also generates EDR/REDR on tethering signature change. The generated EDR/REDR contains the signature of the previous SYN packet. Once the flow is classified as tethered (from first SYN or mid-flow SYN), that flow will remain tethered throughout the lifetime of the flow.

This feature provides the capability to limit the number of SYN packets that need to be analyzed on the same flow. As EDR/REDR gets generated in mid-flow due to tethering signature change, a new closure reason is introduced to track such scenarios.

The **max-syn-packet-in-flow** command in the ACS Rulebase Configuration mode is configured to limit the maximum number of SYN packets to be analyzed for tethering detection. The **flow end-condition tethering-signature-change** command in the ACS Rulebase Configuration mode is configured to control the behavior of mid-flow EDR generation due to tethering signature change.

Tethering Detection Bypass based on Interface-ID

The Tethering Detection feature is enhanced to support tethering detection bypass based on Interface ID. A 64-bit Interface ID can be configured to bypass tethering detection for flows with this interface ID matching the interface ID of the source IPv6 address.

The **bypass interface-id** keyword is added to the **tethering-detection** command in the ACS Configuration mode to bypass IP-TTL and OS based tethering detection for IPv6 packets.

- By default, tethering detection is not bypassed.
- Only one Interface ID can be configured for tethering detection bypass in the active-charging service.
- Only IPv6 OS-based and IPv6 IP-TTL-based tethering detection is bypassed. IPv4 tethering detection and UA-based tethering detection is not impacted.
- Tethering detection bypass is applicable only to IPv6 flows.
- Tethering detection bypass is applicable only to IPv6 OS database lookup and IPv6 TTL lookup. EDR generation, IPv6 OS signature generation and population of the signature in EDR will remain unaffected.

Tethering Detection Databases

The Tethering Detection database files must be populated and loaded on to the ASR chassis by the administrator. The procedure to load the database is the same for all the different databases.

Before the database(s) can be loaded for the first time, tethering detection must be enabled using the **tethering-database** CLI command in the ACS Configuration Mode.

For all the databases, only a full upgrade of a database file is supported. Incremental upgrade is not supported. If, for any particular database, the upgrade procedure fails, the system will revert back to the previous working version of that database.

In 15.0 and later releases, os-signatures can be collected from TCP SYN even when tethering detection is disabled in the rulebase. The os-signature will be parsed if an EDR/UDR with an os-signature variable is present in a rulebase or charging-action in the rulebase. This can be used to collect os-signatures that can then be used to build an OS database for the tethering detection feature.

In 20.0 and later releases, TAC-db lookup for tethering detection can be enabled or disabled using the **tethering-detection tac-db** CLI command in the ACS Configuration mode. Based on the CLI configuration, TAC-db lookup needs to be performed at session setup.

Loading and Upgrading Tethering Detection Databases

This section provides an overview of loading and upgrading the databases used in tethering detection.

The database files from MUR/MURAL must be copied onto the ASR chassis to the following directory path designated for storing the database files:

/hd-raid/databases/

Any further upgrades to the database files can be done by placing the file named *new-filename* in the designated directory path. ACS auto-detects the presence of files available for upgrade daily. When a new version of a file is found, the upgrade process is triggered. The upgrade can also be forced by running the upgrade command in the CLI. On a successful upgrade this file is renamed to *filename*.

MUR/MURAL Support for Tethering Detection

The ASR chassis works in conjunction with the Mobility Unified Reporting (MUR/MURAL) application to facilitate tethering detection on the chassis.

If MUR/MURAL is not deployed, then the database file must be manually placed on the ASR chassis, in the /hd-raid/databases/ directory, and loaded into configuration using CLI command.

For more information on MUR, refer to the *MUR Online Help System* and the *Mobility Unified Reporting System Installation and Administration Guide*.

For more information on MURAL, refer to the *MURAL Online Help System* and the *Mobility Unified Reporting and Analytics System Installation and Administration Guide*.

Session Recovery Support

The following Session Recovery features are supported:

- Database recovery after SessCtrl getting killed
- Database recovery after one or more SessMgrs getting killed

Note that depending on the size of the database files and the number of SessMgrs operational in the system, it may take sometime (ranging from five seconds to five minutes) for the database to become available in all the SessMgrs post recovery/migration.

How It Works

This section describes the Tethering Detection configuration. The following examples illustrate two different implementations of the Tethering Detection configuration.

- The following type of configuration is suitable where ECS performance is critical and the operator wants to put in a flat charging plan in place for all the tethered traffic. In such a scenario, addition of a single new ruledef to the configuration suffices. Placing this ruledef at the highest priority in the rulebase will ensure all the tethered flows are charged as per the tariff plan for tethered traffic.

```
configure
     active-charging service ecs_service
          tethering-database
          ruledef tethered-traffic
               tethering-detection flow-tethered
               tcp any-match = TRUE
               exit
          ruledef ftp-pkts
               ftp any-match = TRUE
               exit
          ruledef http-pkts
               http any-match = TRUE
               exit
          ruledef tcp-pkts
               tcp any-match = TRUE
               exit
          ruledef ip-pkts
               ip any-match = TRUE
               exit
          ruledef http-port
               tcp either-port = 80
               rule-application routing
               exit
          ruledef ftp-port
               tcp either-port = 21
               rule-application routing
               exit
          charging-action premium
               content-id 1
               retransmissions-counted
               billing-action egcdr
               exit
          charging-action standard
               content-id 2
               retransmissions-counted
               billing-action egcdr
               exit
          rulebase consumer
               tethering-detection
               action priority 10 ruledef tethered-traffic charging-action premium
               action priority 20 ruledef ftp-pkts charging-action standard
               action priority 30 ruledef http-pkts charging-action standard
               action priority 40 ruledef tcp-pkts charging-action standard
               action priority 50 ruledef ip-pkts charging-action standard
               route priority 80 ruledef http-port analyzer http
               exit
          rulebase default
          end
```

- The following type of configuration is suitable when operators want to apply differentiated charging to various flows that are found to be tethered. In this case, traffic that requires different charging action or

content ID when it is tethered will be identified using two ruledefs, one with "flow-is-tethered = TRUE" option and another without this option. This configuration provides finer granularity of control but results in higher performance degradation because the rule matching tree size increases.

```
configure
     active-charging service ecs_service
          tethering-database
          ruledef ftp-pkts
               ftp any-match = TRUE
               exit
          ruledef ftp-pkts-tethered
               ftp any-match = TRUE
               tethering-detection flow-tethered
               exit
          ruledef http-pkts
               http any-match = TRUE
               exit
          ruledef http-pkts-tethered
               http any-match = TRUE
               tethering-detection flow-tethered
               exit
          ruledef tcp-pkts
               tcp any-match = TRUE
               exit
          ruledef tcp-pkts-tethered
               tcp any-match = TRUE
               tethering-detection flow-tethered
               exit
          ruledef ip-pkts
               ip any-match = TRUE
               exit
          ruledef ip-pkts-tethered
               ip any-match = TRUE
               tethering-detection flow-tethered
               exit
          ruledef http-port
               tcp either-port = 80
               rule-application routing
               exit
          ruledef ftp-port
               tcp either-port = 21
               rule-application routing
               exit
          charging-action premium-http
               content-id 10
               retransmissions-counted
               billing-action egcdr
               exit
          charging-action premium-ftp
               content-id 20
               retransmissions-counted
               billing-action egcdr
               exit
          charging-action premium
               content-id 1
               retransmissions-counted
               billing-action egcdr
               exit
          charging-action standard
               content-id 2
               retransmissions-counted
               billing-action egcdr
               exit
          rulebase consumer
```

```
    tethering-detection
   action priority 10 ruledef ftp-pkts-tethered charging-action premium-ftp

    action priority 20 ruledef ftp-pkts charging-action standard
  action priority 30 ruledef http-pkts-tethered charging-action premium-http

    action priority 40 ruledef http-pkts charging-action standard
    action priority 50 ruledef tcp-pkts-tethered charging-action premium
    action priority 60 ruledef tcp-pkts charging-action standard
    action priority 70 ruledef ip-pkts-tethered charging-action premium
    action priority 80 ruledef ip-pkts charging-action standard
    route priority 80 ruledef http-port analyzer http
    exit
rulebase default
    end
```

Configuring Tethering Detection

This section describes how to configure the Tethering Detection feature to detect subscriber flows from PC devices tethered to mobile smartphones.

Important This command is available only if the *Smartphone Tethering Detection* license is enabled. For more information please contact your Cisco account representative.

To enable and configure the Tethering Detection feature, use the following configuration:

```
configure
  active-charging service <ecs_service_name>
    tethering-database [ ipv6-os-signature ipv6_os_signature_db_file_name |
os-signature <os_signature_db_file_name> | tac <tac_db_file_name> | ua-signature
<ua_signature_db_file_name> ] +
    [ no ] tethering-detection { bypass interface-id ifid | dns-based
nat64 ipv6_network_prefix | tac-db }
    ruledef <tethering_detection_ruledef_name>
      tethering-detection [ dns-based | ip-ttl | os-ua ] {
flow-not-tethered | flow-tethered }
      exit
    rulebase <rulebase_name>
     tethering-detection { dns-based | bypass interface-id ifid | ip-ttl
 value ttl_value | max-syn-packet-in-flow max_syn_packets | os-db-only |
os-ua-db | ua-db-only }
      action priority <priority> ruledef <tethering_detection_ruledef_name>
charging-action <charging_action_name>
      ...
      end
```

Notes:

- The default filename for the IPv6 OS Signature database is **v6-os-db**. The default filename can be changed by the user only during boot up. Once the system is up and running, the database file name cannot be modified. This is true for all tethering related database files.

- In release 20.2, the **bypass interface-id** *ifid* keyword is added in the ACS Rulebase Configuration mode to bypass IP-TTL and OS based tethering detection for IPv6 packets. *ifid* specifies the Interface ID from an IPv6 address, that is a 64-bit unsigned integer.

  When configured, all the IPv6 flows having this interface ID in the source IP address will bypass IP-TTL and OS based tethering detection.

- In release 20.2, the **max-syn-packet-in-flow** *max_syn_packets* keyword is added in the ACS Rulebase Configuration mode to determine the number of SYN packets applicable for tethering detection in a flow. *max_syn_packets* must be an integer ranging from 1 to 3.

  Default number of SYN packets is 1. This means that only the first SYN packet in flow will be analyzed for IP-TTL/OS signature generation and tethering detection. All other mid flow SYN packets will be ignored for IP-TTL/OS signature generation and tethering detection.

- In release 21, the **dns-based nat64** *ipv6_network_prefix* keyword is added in the ACS Configuration mode to configure DNS-based lookup for tethering detection. *ipv6_network_prefix* must be an IPv6 colon-separated-hexadecimal notation with subnet mask bit. IPv6 also supports :: notation.

  The configured NAT64 prefixes are used to identify the IPv6 flows that will be considered for DNS-based tethering detection.

- In release 21, the **dns-based** keyword is added in the ACS Rulebase Configuration mode to perform tethering detection based on DNS pattern. When DNS-based tethering detection is configured, this feature is enabled for all subscribers using this billing plan.

- In release 21, the **dns-based** keyword is added in the ACS Ruledef Configuration mode to match traffic identified as tethered or non-tethered by the DNS-based detection solution.

Enabling TAC Database Lookup

To enable TAC-db lookup for tethering detection in the ACS configuration mode, use the following configuration.

```
configure
   active-charging service service_name
      [ no ] tethering-detection { tac-db }
      end
```

Notes:

- **no tethering-detection tac-db**: Skips TAC-db lookup for tethering detection.
- Enabling TAC-db lookup for tethering detection is the default behavior.

Verifying your Configuration

To verify your configuration, in the Exec Mode, enter the following command:

```
show active-charging subscribers full all
```

The **Tethering Detection Enabled** field displays whether tethering detection for a subscriber is enabled or not.

Enabling DNS Caching

Use the following configuration to allow caching from DNS flows when the DNS-based tethering detection is enabled.

```
configure
   active-charging service service_name
      charging-action charging_action_name
         flow tethering-detection dns-based host-table caching
         { default | no } flow tethering-detection
         end
```

Upgrading Tethering Detection Databases

To upgrade the Tethering Detection feature databases, in the Exec mode, use the following CLI command:

```
upgrade tethering-detection database { all | ipv6-os-signature |
os-signature | tac | ua-signature } [ -noconfirm ]
```

Notes:

- To load and upgrade the databases used in detecting tethering, the database files must be copied from MUR/MURAL onto the ASR chassis to the designated directory path for storing the database files:

  `/mnt/hd-raid/data/databases/`

  Any further upgrades to the database files can be done by placing the file named *new-filename* in the designated directory path. ECS auto-detects the presence of files available for upgrade daily. When a new version of a file is found, the upgrade process is triggered. The upgrade can also be forced by running the upgrade command in the CLI. On a successful upgrade this file is renamed to *filename*.

Monitoring and Troubleshooting the Tethering Detection feature

This section provides information regarding show commands and/or their outputs in support of this feature.

Tethering Detection Show Command(s) and/or Outputs

This section provides information regarding show commands and/or their outputs in support of the Tethering Detection feature.

Bulk Statistics

Bulk statistics reporting for the Tethering Detection feature is supported.

The following bulk statistics are available in the ECS schema:

- ecs-td-tac-id-lookups
- ecs-td-tac-id-matches
- ecs-td-os-signature-lookups
- ecs-td-os-signature-matches
- ecs-td-ua-signature-lookups

- ecs-td-ua-signature-matches
- ecs-td-v6-os-signature-lookups
- ecs-td-v6-os-signature-matches
- ecs-td-total-flows-scanned
- ecs-td-tethered-flows-detected
- ecs-td-non-tethered-flows-detected
- ecs-td-osua-total-flows-scanned
- cs-td-osua-tethered-flows-detected
- ecs-td-osua-non-tethered-flows-detected
- ecs-td-ipttl-total-flows-scanned
- ecs-td-ipttl-tethered-flows-detected
- ecs-td-ipttl-non-tethered-flows-detected
- ecs-td-current-tethered-subscribers
- ecs-td-tethered-uplink-packets
- ecs-td-tethered-downlink-packets
- ecs-td-ipttl-tethered-uplink-packets
- ecs-td-ipttl-tethered-downlink-packets
- ecs-td-tether-to-tether-signature-change-in-flow
- ecs-td-tether-to-non-tether-signature-change-in-flow
- ecs-td-non-tether-to-tether-signature-change-in-flow
- ecs-td-non-tether-to-non-tether-signature-change-in-flow

For more information on these bulk statistics, see the *ECS Schema Statistics* chapter of the *Statistics and Counters Reference*.

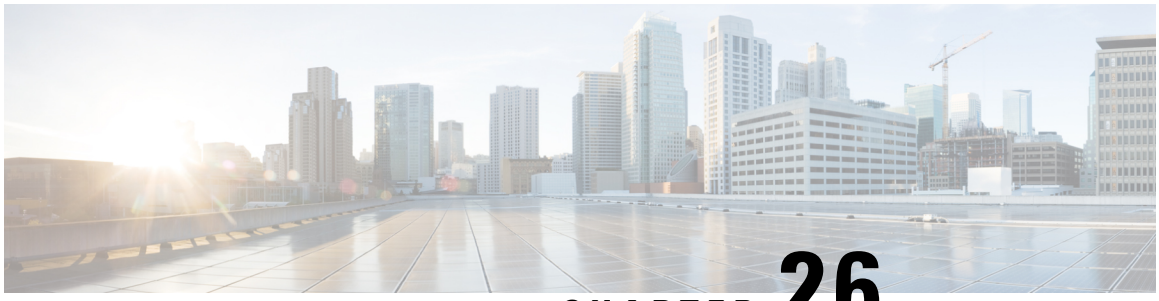
SNMP Traps

SNMP traps indicate the load/upgrade status for the TAC, UA, OS and IPv6 OS signature tethering databases.

The following SNMP traps are added:

- starTetheringOSDatabaseUpgradeFailureStatus - OS database upgrade failure status
- starTetheringOSDatabaseUpgradeSuccessStatus - OS database upgrade success status
- starTetheringTACDatabaseUpgradeFailureStatus - TAC database upgrade failure status
- TetheringTACDatabaseUpgradeSuccessStatus - TAC database upgrade success status
- starTetheringUADatabaseUpgradeFailureStatus - UA database upgrade failure status
- starTetheringUADatabaseUpgradeSuccessStatus - UA database upgrade success status
- starTetheringV6OSDatabaseUpgradeFailureStatus - IPv6 OS database upgrade failure status
- starTetheringV6OSDatabaseUpgradeSuccessStatus - IPv6 OS database upgrade success status

For more information regarding SNMP MIB objects, refer to the *SNMP MIB Reference*.



CHAPTER 26

Transactional Rule Matching

This chapter describes the Transactional Rule Matching (TRM) feature and provides detailed information on the following topics:

Feature Description

The Transactional Rule Matching (TRM) feature enables the Enhanced Charging Service (ECS) to bypass per-packet rule matching on a transaction once the transaction is fully classified. This enables ECS to better utilize CPU resources and accomodate additional throughput for the system, thus improving the overall performance.

A transaction for TRM can be defined as the entire UDP flow, the ACK of the 3-way handshake to the FIN/RST of a TCP flow, or the HTTP request to the next HTTP request, or HTTP request to the FIN/RST for the final request of the flow. The TRM feature can perform rule matching on IP L4 rules (UDP, TCP), HTTP, and HTTPS.

Fastpath

The Fastpath feature can be used to reduce the overall system performance impact as a large amount of data packet is consumed through the ECS data path. The Fastpath feature introduces an alternate ECS data path (Fastpath) with limited supported features. By limiting the supported features, Fastpath eliminates the overhead of packets being subjected to the large number of feature-based conditional checks in ECS.

Fastpath does not replace the existing data path, and works in parallel with the existing ECS data path. The Fastpath feature is part of the Transactional Rule Matching (TRM) feature and requires TRM to be enabled.

Important From 16.0 release, **Transactional Rule Matching** and **Fastpath** functionalities have been merged, and will be governed only by the **transactional-rule-matching** keyword alone. The **fastpath** keyword independently can no longer be used to turn on or turn off this functionality.

Feature Support

The following table provides information on the supported and unsupported features of Fastpath. Features that are listed under the Optimized column in the table below indicate that the features are directly supported by Fastpath. Features that are listed under the Eligible column in the table below indicate that a flow requiring the feature does not prevent it from taking advantage of Fastpath. Features that are listed under the unsupported column in the table below indicate that all packets in the flow that belong to the feature is not supported for Fastpath and will take the existing ECS data path.

Important The TRM feature is supported in SSI platform; earlier it was restricted only to ASR 5500.

Important In 20.0 and later releases, MVG is not supported. For more information, contact your Cisco account representative.

Table 7: Flow-level Support

Feature	Fastpath Optimized	Fastpath Eligible	Unsupported
Url Redirect	—	Yes	—
Charging Bucket Maintenance	—	Yes	—
ITC/BW control	—	Yes	—
Next Hop	—	Yes	—
TCP State based rules	—	Yes	—
Post-processing Rules	—	Yes	—
Flow limit -	—	Yes	—
DSCP / IP TOS	—	Yes	—
ICSR	—	Yes	—
Session Recovery	—	Yes	—
Content Filtering (CF) Static	—	—	Yes
CF Dynamic	—	—	Yes
Socket Migration	—	—	Yes
Blacklisting	—	Yes	—
ICAP	—	—	Yes

Feature	Fastpath Optimized	Fastpath Eligible	Unsupported
NAT	Yes	—	—
SFW	—	—	Yes
Video transrating	—	—	Yes
MVG CAE Readdressing	—	—	Yes
MVG Pacing	—	—	Yes
MVG Link Monitoring	—	—	Yes
MVG Header Insertion	—	—	Yes
ADC/P2P Refer to the Note below this table.	—	Yes	—
SIP-ALG (App Layer Gateway)	—	—	Yes
H323 - ALG	—	—	Yes
DCCA	—	Yes	—
IPv6	Yes	—	—
Flow Readdress	—	—	Yes
Idle-timeout handling	Yes	—	—
Connection termination (2MSL)	—	Yes	—
TCP Proxy	—	—	Yes
QOS	—	Yes	—
Lawful Intercept	—	Yes	—
HTTP/HTTPS	—	Yes	—
Non HTTP L7 protocols	—	—	Yes
NON UDP/TCP flows	—	—	Yes
Tethering detection	—	—	Yes
Gx	—	Yes	—
Gy	—	Yes	—
HEE	—	Yes	—

Feature	Fastpath Optimized	Fastpath Eligible	Unsupported
Radius	—	Yes	—
Diameter	—	Yes	—
L4 checksum	—	—	Yes
TCP link monitoring	—	—	Yes
Header enrichment	—	—	Yes
Wimax Hotlining	—	—	Yes
Parsing Error Detection Denial	—	—	Yes
IP only Byte Counting/Charging	Yes	—	—
DNS Snoop	—	Yes	—
ICMP	—	—	Yes
Data Record generation	—	Yes	—
Fair Usage	—	Yes	—
SPI	Yes	—	—

Important Note that all ADC protocols are not Fastpath eligible. Refer to the *ADC Administration Guide* for more information.

Even when a flow is supported for Fastpath, some packets for the flow are not eligible to be processed in Fastpath. When a packet is not eligible, the packet is processed in the existing ECS data path. The following table provides information on the packet-level support in Fastpath:

Table 8: Packet-level Support

Packet Handling Feature	Fastpath Eligible	No Support
Valid UDP/TCP in order pkts	Yes	—
OOO Packet handling	—	Yes
TCP Retransmissions	Yes	—
IP Fragmentation	—	Yes
TCP Handshaking	—	Yes
TCP Termination	—	Yes

Packet Handling Feature	Fastpath Eligible	No Support
First packet of Flow	—	Yes
Gx Rule Update	—	Yes
Invalid L3/L4 packet	Yes	—
Packet already queued	—	Yes

Limitations and Dependencies

The following are limitations to the TRM feature:

- TRM is supported only on the ASR 5500 platform.
- TRM is limited to flows with no protocol routing rules with the exception of HTTP and HTTPS flows. All other flows are not supported and TRM does not have any impact on other flows.
- When TRM is enabled, the following functionalities are affected:
  - Per direction rule matching.
  - TCP state rules for the duration of the TRM transaction.
  - Configuring delay charging when the TRM feature is enabled impacts only packets outside transaction boundaries. All packets within the transaction boundary will be applied to the application (i.e. HTTP).
- Once a flow is classified to a ruledef (first packet in flow for UDP or the first data packet after the 3-way handshake for a TCP flow), TRM will attempt to use that matched rule for the duration of the transaction. This might result in the ruledefs such as those with mid-transaction TCP states or packet direction to be ignored for the flow.

Configuring Transactional Rule Matching Feature

Use the following configuration to enable the Transactional Rule Matching (TRM) feature.

Important The TRM feature is supported in SSI platform; earlier it was restricted only to ASR 5500.

```
configure
  active-charging service <ecs_service_name>
    rulebase <rulebase_name>
      transactional-rule-matching
      end
```

Notes:

- Use the **no transactional-rule-matching** command or **default transactional-rule-matching** command to disable transactional rule matching.

- Transactional rule matching is disabled by default.

Important From 16.0 release, **Transactional Rule Matching** and **Fastpath** functionalities have been merged, and will be governed by only the **transactional-rule-matching** keyword alone. The keyword **fastpath** independently can no longer be used to turn on or turn off this functionality.

Verifying the TRM Configuration

To verify your configuration, in the Exec mode, enter the following command:

show active-charging rulebase name *<rulebase_name>*

Monitoring and Troubleshooting the Transactional Rule Matching feature

This section provides information on the bulk statistics and show commands available to support this feature.

show active-charging rulebase statistics

The output of this command displays the TRM statistics.

- TRM Statistics:
  - Bypassed rule-matching
  - Rule-matching bypass triggered
  - Failed to create dynamic flow element
  - Flow cleared, rule not found
  - Flow cleared, rule stats not found
  - Flow cleared, group not found
  - Flow cleared, group rule error
  - Flow cleared, rule error
  - Flow cleared, rule expired
  - Flow cleared, pkts not forwarded
  - Flow cleared, pkts buffered
  - Flow cleared, SEF event
  - Flow cleared, egcdr bucket idle time out
  - FastPath Eligible Flows
  - FastPath Packets

- FastPath Failures

show active-charging rulebase statistics name

The output of this command displays the TRM statistics.

- TRM Statistics:
  - Bypassed rule-matching
  - Rule-matching bypass triggered
  - Failed to create dynamic flow element
  - Flow cleared, rule not found
  - Flow cleared, rule stats not found
  - Flow cleared, group not found
  - Flow cleared, group rule error
  - Flow cleared, rule error
  - Flow cleared, rule expired
  - Flow cleared, pkts not forwarded
  - Flow cleared, pkts buffered
  - Flow cleared, SEF event
  - Flow cleared, egcdr bucket idle time out
  - FastPath Eligible Flows
  - FastPath Packets
  - FastPath Failures

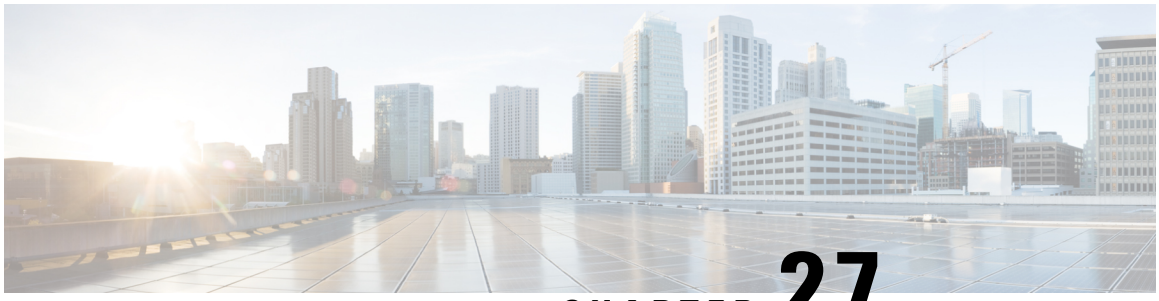
Bulk Statistics

Bulk statistics reporting for the TRM feature is supported.

The following bulk statistics are available in the ECS schema:

- trm-rule-match-bypassed
- trm-rule-match-bypass-triggered
- fp-eligible-flows
- fp-packets
- fp-failures

For more information on these bulk statistics, see the *ECS Schema Statistics* chapter of the *Statistics and Counters Reference*.



CHAPTER 27

URL-based Re-addressing

This chapter describes the URL-based re-addressing feature and provides detailed information on the following topics:

Feature Description

The URL-based re-addressing feature is applied based on L7 rule matching for HTTP URLs in addition to re-addressing charging action based on L3/L4 rule matching. HTTP request with specific token or complete URL must be redirected to a separate server and must be transparent to the UE.

Flow-based re-addressed connection

Flow-based re-addressed connection is the default behavior of this feature. In this type, after a HTTP connection is setup with Original Server, all subsequent requests will be sent to it until the URL-based re-addressing rule matches. This behavior holds true even for multiple concatenated HTTP requests in one packet.

How It Works

This section describes how the URL re-addressing feature works.

Call Flows

The following call flow explains the URL HTTP Request Re-addressing feature.

Figure 13: URL HTTP Request Re-addressing

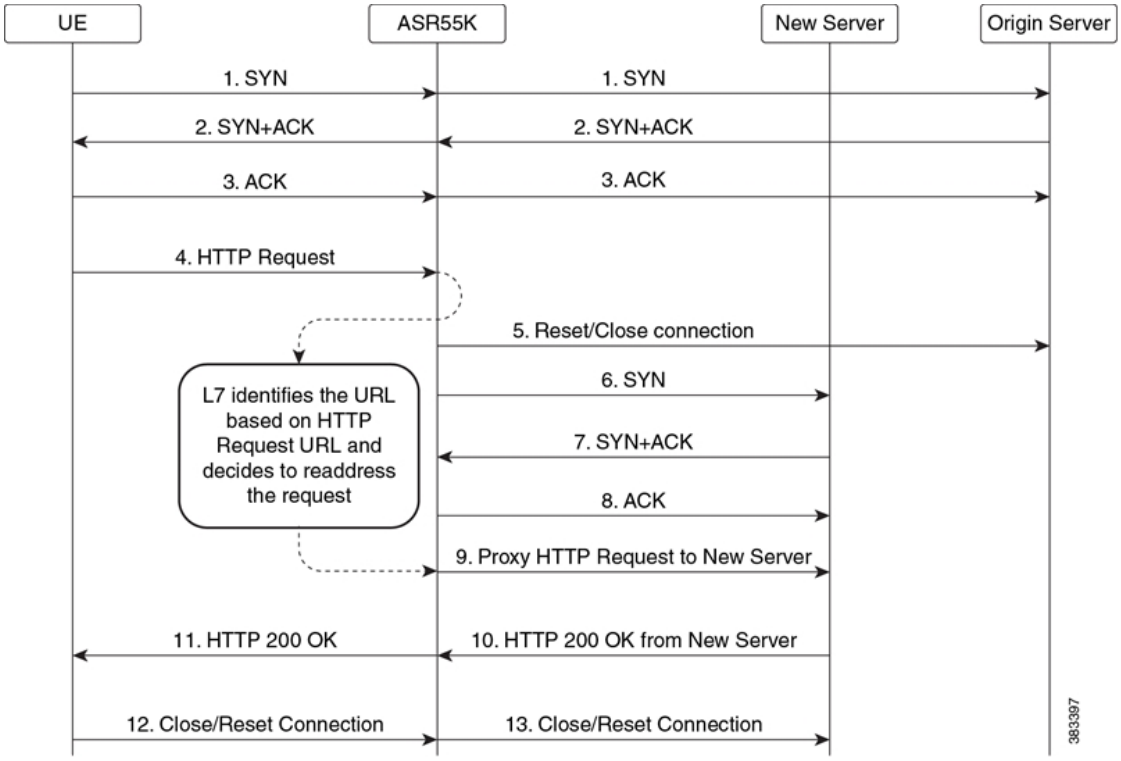


Table 9: URL HTTP Request Re-addressing

Step	Description
1—3	UE sets up a TCP connection with the Origin Server (OS) by sending SYN. The TCP three-way handshake takes place between UE and the Origin Server.
4	UE sends a HTTP request to the OS which passes through the ASR 5500 L7 DPI rule-matching. The URL of the request contains a known token, domain name, or a token configured at ASR 5500 (in a ruledef).
5—8	ASR 5500 using L7 DPI recognizes that the request is for New Server. ASR 5500 breaks/closes the existing TCP connection with OS and establishes a new connection with the New Server. TCP proxy is used to maintain TCP connection between UE and ASR 5500.
9	ASR 5500 sends the HTTP Request destined for Origin Server to New Server.
10	New Server sends back the content in HTTP Response.

Step	Description
11	ASR 5500 proxies the content back to the UE.
12	UE closes the TCP connection.
13	ASR 5500 closes the connection with the New Server.

Configuring URL-based Re-addressing

Use the following configuration in the ACS Charging Action Configuration Mode to configure the URL server to re-address for the specified charging action.

The URL-based re-addressing feature is configured and enabled using the **charging-action** command options within an Active Charging Service.

```
configure
  active-charging service <service_name>
    charging-action <charging_action_name>
      flow action url-readdress server <ipv4_address> [ port <port_number> ]
      no flow action
      end
```

Monitoring and Troubleshooting the URL-based Readdressing feature

This section provides information on the show commands available to support this feature.

show active-charging charging-action statistics name

The output of this command displays the statistics for readdressing failures due to flow without SYN and duplicate key failures. This command also displays the number of packets discarded on readdressing failure if the **discard-on-failure** keyword is enabled else this number will be zero.

- Readdressing Failures Statistics(Packets):
  - Non SYN flow
  - Duplicate Key
  - Dropped Pkts

show active-charging sessions full all

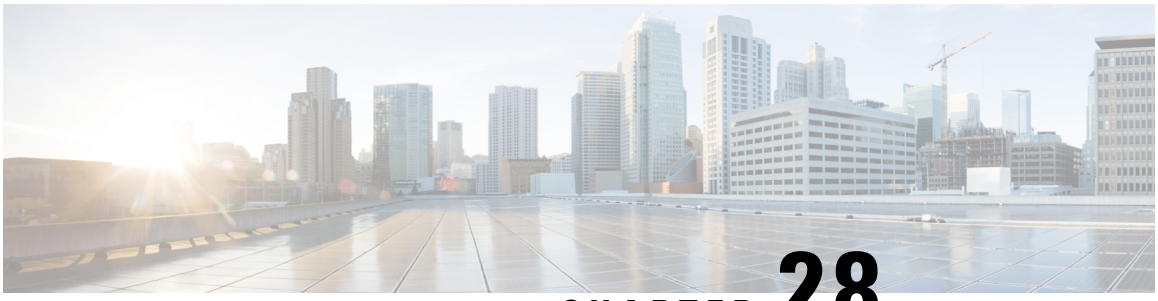
The output of this command displays the statistics for readdressing failures due to flow without SYN and duplicate key failures. This command also displays the number of packets discarded on readdressing failure if the **discard-on-failure** keyword is enabled else this number will be zero.

- Total Readdressing Failure Packets
- Non SYN flow
- Duplicate Key
- Dropped Pkts

show active-charging subsystem all

The output of this command displays the statistics for readdressing failures due to flow without SYN and duplicate key failures. This command also displays the number of packets discarded on readdressing failure if the **discard-on-failure** keyword is enabled else this number will be zero.

- Readdressing Failures Statistics (Packets):
  - Non SYN flow
  - Duplicate Key
  - Dropped Pkts



CHAPTER 28

X-Header Insertion and Encryption

Feature Summary and Revision History

Table 10: Summary Data

Applicable Products and Functional Area	• P-GW • SAEGW
Applicable Platforms	• ASR 5500 • VPC - DI • VPC - SI
Feature Default	Disabled - License Required
Related Changes in This Release	Not applicable
Related Documentation	• *Command Line Interface Reference* • *ECS Administration Guide* • *Statistics and Counters Reference*

Revision History

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

Revision Details	Details
In this release, the TLS_RSA_WITH_AES_256_GCM_SHA_384 algorithm is introduced for enhanced security.	21.9
First introduced.	Pre 21.2

Feature Description

The X-Header Insertion and X-Header Encryption features, collectively known as Header Enrichment, enables to append headers to HTTP/WSP GET and POST request packets, and HTTP Response packets for use by end applications, such as mobile advertisement insertion (MSISDN, IMSI, IP address, user-customizable, and so on).

Important In this release, the X-Header Insertion and Encryption features are supported only on the GGSN and P-GW.

Following are the software requirements for the new TLS_RSA_WITH_AES_256_GCM_SHA_384 attribute in CDR:

- Configure AES-256-GCM-sha384 encryption algorithm with 256-bit keys. This configuration is same as the one used for the RC4MD5 encryption.
- Use the existing re-encryption timeout CLI, which is used at rulebase level, for re-encryption.
- For AES-GCM encryption, use the optional **salt** flag. This flag is used to randomize the keys, which are generated from the passphrase, and the Initialization Vectors (IV).

License Requirements

X-Header Insertion and X-Header Encryption are both licensed Cisco features. 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 the *Software Management Operations* chapter in the *System Administration Guide*.

X-Header Insertion

This section provides an overview of the X-Header Insertion feature.

Extension header (x-header) fields are the fields not defined in RFCs or standards but can be added to headers of protocol for specific purposes. The x-header mechanism allows additional entity-header fields to be defined without changing the protocol, but these fields cannot be assumed to be recognizable by the recipient. Unrecognized header fields should be ignored by the recipient and must be forwarded by transparent proxies.

The X-Header Insertion feature enables inserting x-headers in HTTP/WSP GET and POST request packets and HTTP response packets. Operators wanting to insert x-headers in HTTP/WSP request and HTTP response packets, can configure rules for it. The charging-action associated with the rules will contain the list of x-headers to be inserted in the packets.

For example, if you want to insert the field *x-rat-type* in the HTTP header with a value of *rat-type*, the header inserted should be:

x-rat-type: geran

where, *rat-type* is *geran* for the current packet.

X-Header Encryption

This section provides an overview of the X-Header Encryption feature.

X-Header Encryption enhances the X-header Insertion feature to increase the number of fields that can be inserted, and also enables encrypting the fields before inserting them.

If x-header insertion has already happened for an IP flow (because of any x-header format), and if the current charging-action has the first-request-only flag set, x-header insertion will not happen for that format. If the first-request-only flag is not set in a charging-action, then for that x-header format, insertion will continue happening in any further suitable packets in that IP flow.

Changes to x-header format configuration will not trigger re-encryption for existing calls. The changed configuration will however, be applicable for new calls. The changed configuration will also apply at the next re-encryption time to those existing calls for which re-encryption timeout is specified. If encryption is enabled for a parameter while data is flowing, since its encrypted value will not be available, insertion of that parameter will stop.

Important Recovery of flows is not supported for this feature.

TCP OOO Packets

ECS handles TCP OOO packets in two ways depending on the rulebase configuration:

Transmit Immediately: If the rulebase is configured to transmit immediately for TCP OOO packets, the OOO packets will be forwarded immediately, and a copy of this packet will be added to the OOO queue for analysis.

Transmit After Reordering: If the rulebase is configured to transmit after reordering for TCP OOO packets, the OOO packets will be added to the OOO queue for analysis. Header insertion on OOO request packets occurs on reordering packets that are received before the OOO request timeout. When a reordering packet is received, the queued packets are forwarded. However, if a reordering packet is not received before the OOO queue timeout, the queued packet will be forwarded without any analysis done to those packets.

Important When TCP OOO processing has been configured in the rulebase, a session manager crash might be observed due to overlapping TCP segments and/or reordering packet arriving within TCP OOO configured timeout value or default value (5 sec). This issue can be resolved by changing the rulebase configuration for TCP OOO packets from **transmit after-reordering** to **transmit immediately**.

In 20 and later releases, TCP OOO packets will be buffered for HTTP traffic until the header enrichment is completed. The header enrichment is supported on either first request packet or all request packets, response packets, or on both. So following packets after the header enrichment is complete will not require buffering of OOO packets and the packets can be transmitted immediately. This will improve memory optimization

and network performance. The out-of-order packets will be buffered as per the x-header configuration in any of the charging-action for the subscriber's rulebase.

- If x-header insertion is only for the request packet, then out-of-order buffering will be supported till the header completion of the request packet.
- If x-header insertion is only for the response packet, then out-of-order buffering will be supported till the header completion of the response packet.
- If x-header insertion is for both request and response packets, then out-of-order buffering will be supported till the completion of HTTP headers for that packet.

Limitations of buffering TCP OOO packets:

- This enhancement will be supported only for HTTP flows of x-header enrichment feature.
- In case of pipeline flows with multiple transactions, if a new OOO request/response is received while the previous request/response is still going on, then x-header insertion will not work for the new request/response of that flow.

IP Fragmented Packets

ECS can perform Header Enrichment to IP fragmented packets when all the fragments are received before the reassembly timeout. If the packet size after Header Enrichment exceeds the MSS of the session, the reassembled packet gets segmented, the multiple segments are forwarded.

Limitations to the Header Insertion and Encryption Features

This section lists known limitations to insertion and encryption of x-header fields in HTTP/WSP request and HTTP response packets.

The following are limitations to insertion and encryption of x-header fields in HTTP headers.

Limitations in StarOS 14.0 and later releases:

- Header insertion does not occur for packets with incomplete HTTP headers.
- If a flow has x-header insertion and later some IP fragments are received for which reassembly fails, sequence space of that segment will be mismatched.
- ECS does not support applying more than one modifying action on an inbound packet before sending it on the outbound interface. For example, if header insertion is applied on a packet, then the same packet is not allowed to be modified for NAT/ALG and MSS insertion.
- Header enrichment works only for the first request of a packet with concatenated requests, when the packets are buffered at DCCA. There are no limitations on header enrichment for single GET or pipelined GET requests.
- Header enrichment works for packets at DCCA only when the packets pending of header insertion is buffered.
- Receive window will not be considered during header enrichment. That is, after header enrichment if packet exceeds receive window, ECS will not truncate the packet.
- The maximum bytes per request after header enrichment is 2400 bytes. If concatenated requests exist, a maximum of 2400 bytes after header enrichment can be inserted.

If due to header insertion, the packet size exceeds this limit, the behavior is unpredictable.

- Only those x-header fields in header portion of application protocol that begin with "x-" are parsed at HTTP analyzer. In URL and data portion of HTTP any field can be parsed.
- EDR generation for x-header fields in Response packets will not be supported.

Limitations in StarOS 12.3 and earlier releases:

- The packet size is assumed to be less than "Internal MED MTU size, the size of header fields inserted". If the total length of packet exceeds the internal MTU size, header insertion will not occur after the addition of fields.
- Header insertion occurs for both HTTP GET and POST requests. However, for POST requests, the resulting packet size will likely be larger than for GET requests due to the message body contained in the request. If the previous limitation applies, then POST request will suffer a bigger limit due to this.
- Header insertion does not occur for retransmitted packets.
- Header insertion does not occur for packets with incomplete HTTP headers.
- Header insertion does not occur for TCP OOO and IP fragmented packets.
- If a flow has x-header insertion and later some IP fragments are received for which reassembly fails, sequence space of that segment will be mismatched.
- ECS does not support applying more than one modifying action on an inbound packet before sending it on the outbound interface. For example, if header insertion is applied on a packet, then the same packet is not allowed to be modified for NAT/ALG and MSS insertion.
- If a packet is buffered by ICAP, header insertion will not occur for that packet.
- Receive window will not be considered during header enrichment. That is, after header enrichment if packet exceeds receive window, ECS will not truncate the packet.
- Packet size limit is 2400 bytes, if due to header insertion packet size exceeds this limit, behavior is unpredictable.
- Only those x-header fields in header portion of application protocol that begin with "x-" are parsed at HTTP analyzer. In URL and data portion of HTTP any field can be parsed.

The following are limitations to insertion and encryption of x-header fields in WSP headers:

- x-header fields are not inserted in IP fragmented packets.
- In case of concatenated request, x-header fields are only inserted in first GET or POST request (if rule matches for the same). X-header fields are not inserted in the second or later GET/POST requests in the concatenated requests. For example, if there is ACK+GET in packet, x-header is inserted in the GET packet. However, if GET1+GET2 is present in the packet and rule matches for GET2 and not GET1 x-header is still inserted in GET2. In case of GET+POST also, x-header is not inserted in POST.
- In case of CO, x-header fields are not inserted if the WTP packets are received out of order (even after proper re-ordering).
- If route to MMS is present, x-headers are not inserted.

- x-headers are not inserted in WSP POST packet when header is segmented. This is because POST contains header length field which needs to be modified after addition of x-headers. In segmented WSP headers, header length field may be present in one packet and header may complete in another packet.
- x-headers are not inserted in case of packets buffered at DCCA.

Supported X-Headers

This section provides information on the different x-headers supported by ECS.

ECS supports insertion of various x-header fields in the HTTP/WSP GET and POST request packets and HTTP response packets. The x-headers are inserted at the end of the HTTP/WSP header.

The following bearer-related x-headers are supported:

- 3gpp

  The following 3GPP associated fields are supported:

  - apn
  - charging-characteristics
  - charging-id
  - imei
  - imsi
  - qos
  - rat-type
  - s-mcc-mnc
  - sgsn-address

- acr
- customer-id
- ggsn-address
- mdn
- msisdn-no-cc
- radius-string
- radius-calling-station-id
- session-id
- sn-rulebase
- subscriber-ip-address
- username
- user-profile
- uli

The following HTTP-related x-headers are supported:

- host
- url

In addition, ECS also allows string constants to be inserted as an x-header. For more information on configuring the x-header formats, see the *insert* command section in the *ACS x-Header Format Configuration Mode Commands* chapter of the *Command Line Interface Reference*.

X-Header Enrichment Anti-Spoofing

This section provides an overview of the x-Header Enrichment Anti-Spoofing feature.

The Header Enrichment feature allows operators to encrypt and insert subscriber-specific fields as x-headers in to the HTTP headers of URL requests. However, this might leave the header open to spoofing by malicious external devices. The Anti-Spoofing feature enables deletion and modification of the existing x-header fields to protect the operators and subscribers from spoofing, and provides a fraud detection mechanism when an external portal is used for a subscriber or content authorization.

The feature detects and removes user-generated HTTP headers if the header name is similar to the header name used in the x-header format, and when multiple entries of the same field exist in the header, all the similar entries are removed and one with a modified value is inserted at the end of the HTTP header.

When anti-spoofing is enabled, and if the HTTP header in the GET or POST request spawns across more than one packet, the packets with incomplete HTTP header will be buffered. The buffered packets are sent out once the HTTP header is completed.

The Anti-Spoofing feature is disabled by default and can be enabled/disabled at a field level in the CLI.

Limitations to the Anti-Spoofing Feature

- Header enrichment does not occur if a route to the MMS analyzer exist in the rulebase.
- Header enrichment works only for the first request of a packet with concatenated requests, when the packets are buffered at DCCA.
- If a packet is buffered by ICAP, header insertion will not occur for that packet.
- ECS will not be able to perform header enrichment when all fragments are not received before reassembly timeout in the case of IP fragments packets.
- ECS does not perform more than one flow action which modifies the inbound packet before sending it on the outbound interface.
- If the HTTP GET or POST header is not completed in three packets, anti-spoofing will occur only for the last packet in which the header completes, as buffering supported only up to a maximum of two packets.
- Though insertion of fields is allowed without having "x-" in the field name, extension header fields that do not start with "x-" are not deleted.
- The anti-spoofing feature will not be supported for x-headers inserted in Response messages.

Supported Encryption Methods

In Release 21.9, the TLS_RSA_WITH_AES_256_GCM_SHA_384 algorithm is introduced for enhanced security.

The supported types of encryption for encrypting x-header values are RSA, RC4MD5, and AES-256-GCM-SHA384. These encryption types are explained in the following sections.

RSA

You can configure RSA encryption by using an encryption certificate. With this encryption, during a call, the configured fields of the **xheader-format** command are encrypted. When the charging action is hit for the traffic, then the encrypted values of the configured fields are inserted in the HTTP header.

Important The encrypted values change only if the re-encryption timer times out. As RSA encryption consumes time, it is not used each time a field value changes during a call.

RC4MD5

You can configure RC4MD5 encryption at charging action level. For traffic, an encryption that is configured at a charging level takes precedence over rulebase.

In RC4MD5 encryption, the MD5 hash of the key, which is a 128-bit value, is used to encrypt the value using the RC4 encryption. The base 64 value of the value that is received from the RC4 encryption is then inserted in the x-header.

When the charging action is hit for the traffic, then the encrypted values of the configured fields are inserted in the HTTP header.

AES-256-GCM-SHA384

You can configure AES-256-GCM-SHA384 encryption at charging action level. For traffic, an encryption that is configured at a charging action level takes precedence over rulebase.

In AES-256-GCM-SHA384 encryption, the SHA384 hash of the key, which is 384 bits value, is used to encrypt the value using the AES-GCM algorithm. The base 64 of this encrypted value is then inserted in the x-header.

How It Works

This section describes the steps involved to configure the X-Header Insertion and X-Header Encryption features.

X-Header Insertion

The following steps describe how X-Header Insertion works:

Step 1 Creating/configuring a ruledef to identify the HTTP/WSP packets in which the x-headers must be inserted.

Step 2 Creating/configuring a rulebase and configuring the charging-action, which will insert the x-header fields into the HTTP/WSP packets.

Step 3 Creating/configuring the x-header format.

Step 4 Configuring insertion of the x-header fields based on the message type in the charging action.

X-Header Encryption

The following steps describe how X-Header Encryption works:

Step 1 X-header insertion, encryption, and the encryption certificate is configured in the CLI.

Step 2 When the call gets connected, and after each regeneration time, the encryption certificate is used to encrypt the strings.

Step 3 When a packet hits a ruledef that has x-header format configured in its charging-action, x-header insertion into that packet is done using the given x-header-format.

Step 4 If x-header-insertion is to be done for fields which are marked as encrypt, the previously encrypted value is populated for that field accordingly.

Configuring X-Header Insertion and Encryption

This section describes how to configure the X-Header Insertion and Encryption features, collectively known as Header Enrichment.

Configuring X-Header Insertion

This section describes how to configure the X-Header Insertion feature.

Important This feature is license dependent. Please contact your Cisco account representative for more information.

To configure the X-Header Insertion feature:

Step 1 Create/configure a ruledef to identify the HTTP packets in which the x-headers must be inserted. For information on how to create/configure ruledefs, see the *Configuring Rule Definitions* section in the *Enhanced Charging Service Configuration* chapter.

Step 2 Create/configure a rulebase and configure the charging-action, which will insert the x-header fields into the HTTP packets. For information on how to create/configure rulebases, see the *Configuring Rulebase* section in the *Enhanced Charging Service Configuration* chapter.

Step 3 Create the x-header format as described in Creating the X-Header Format, on page 263.

Step 4 Configure the x-header format as described in Configuring the X-Header Format, on page 264.

Step 5 Configure insertion of the x-header fields as described in Configuring Charging Action for Insertion of X-Header Fields, on page 264.

Creating the X-Header Format

To create an x-header format, use the following configuration:

```
configure
  active-charging service ecs_service_name
```

```
xheader-format xheader_format_name
end
```

Configuring the X-Header Format

To configure an x-header format, use the following configuration:

```
configure
  active-charging service ecs_service_name
    xheader-format xheader_format_name
      insert xheader_field_name { string-constant xheader_field_value | variable
 { bearer { 3gpp { apn | charging-characteristics | charging-id | imei |
 imsi | qos | rat-type | s-mcc-mnc | sgsn-address } | acr | customer-id
| ggsn-address | mdn | msisdn-no-cc |  radius-string |
radius-calling-station-id | session-id | sn-rulebase |
subscriber-ip-address | username } [ encrypt ] | http { host | url } }
       end
```

Configuring Charging Action for Insertion of X-Header Fields

To configure a charging action for insertion of x-header fields, use the following configuration:

```
configure
  active-charging service ecs_service_name
    charging-action charging_action_name
     xheader-insert xheader-format xheader_format_name [ encryption { rc4md5
 | aes-256-gcm-sha384 [ salt ] } [ encrypted ] key key ] [
first-request-only ] [ msg-type { response-only | request-and-response }
 ] [ -noconfirm ]
       end
```

Note

- If rc4md5 encryption is configured in the charging action, it will take precedence over RSA certificate based encryption for flows hitting particular charging action.
- X-header insertion in HTTP Response packets can be enabled/disabled using the **msg-type** keyword.
  - **response-only**: When configured in charging-action, x-header will be inserted in HTTP Response packets with specified x-header format.
  - **request-and-response**: When configured in charging-action, x-header will be inserted in both HTTP Request and Response packets with same x-header format.

Configuring X-Header Encryption

This section describes how to configure the X-Header Encryption feature.

Important This feature is license dependent. Please contact your Cisco account representative for more information.

To configure the X-Header Encryption feature:

Step 1 Configure X-Header Insertion as described in Configuring X-Header Insertion, on page 263.

Step 2 Create/configure a rulebase and configure the encryption certificate to use and the re-encryption parameter as described in Configuring X-Header Encryption, on page 265.

Step 3 Configure the encryption certificate to use as described in Configuring Encryption Certificate, on page 265.

Configuring X-Header Encryption

To configure X-Header Encryption, use the following configuration example:

```
configure
  active-charging service ecs_service_name
    rulebase rulebase_name
      xheader-encryption certificate-name certificate_name
      xheader-encryption re-encryption period re-encryption_period
      end
```

Notes:

- This configuration enables X-Header Encryption for all subscribers using the specified rulebase *rulebase_name*.
- If the certificate is removed, ECS will continue using the copy that it has. It will only free its copy if the certificate name is removed from the rulebase.
- Changes to x-header format configuration will not trigger re-encryption for existing calls. The changed configuration will however, be applicable for new calls. The changed configuration will also apply at the next re-encryption time to those existing calls for which re-encryption timeout is specified. If encryption is enabled for a parameter while data is flowing, since its encrypted value will not be available, insertion of that parameter will stop.

Configuring Encryption Certificate

To configure the encryption certificate, use the following configuration example:

```
configure
  certificate name certificate_name pem { { data pem_certificate_data private-key
 pem [ encrypted ] data pem_pvt_key } | { url url private-key pem { [
encrypted ] data pem_pvt_key | url url } }
  end
```

Verifying the X-Header Insertion and Encryption Configuration

Enter the following command in the Exec Mode to verify your configuration:

```
show active-charging xheader-format name xheader_format_name
```

Monitoring and Troubleshooting the X-Header Insertion and Encryption feature

This section provides information on the show commands and/or their outputs available to support this feature.

show active-charging charging-action name

The output of this command displays the information for the RSA header enrichment encryption algorithm.

- Encryption Type: aes-256-gcm-sha384
- Salt : YES/NO

show active-charging charging-action statistics name

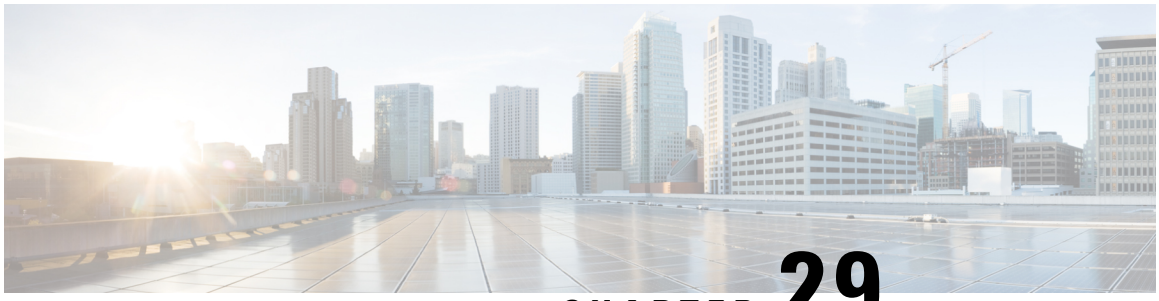
The output of this command displays statistics for X-header information.

- XHeader Information:
  - XHeader Bytes Injected
  - XHeader Pkts Injected
  - IP Frags consumed by XHeader
  - XHeader Bytes Removed
  - XHeader Pkts Removed

show active-charging rulebase statistics name

The output of this command displays the Header Enrichment statistics.

- HTTP header buffering limit reached



CHAPTER 29

Additional Keywords Added to the show subscribers and clear subscribers Commands

Feature Information

Summary Data

Status	Modified Functionality
Introduced-In Release	21.2
Modified-In Release(s)	Not Applicable
Applicable Product(s)	P-GW
Applicable Platform(s)	ASR 5500
Default Setting	Enabled
Related CDETS ID(s)	CSCvc75438
Related Changes in This Release	Not Applicable
Related Documentation	*ECS Administration Guide* *P-GW Administration Guide* *Command Line Interface Reference*

Revision History

Important Revision history details are not provided for features introduced before Release 21.2.

Revision Details	Release	Release Date
New in this release.	21.2	April 27, 2017

Feature Description

After simultaneous Inter-Chassis Session Recovery (ICSR) switchovers, a number of subscribers are losing the assigned dynamic charging rules. To track these issues, additional filters or keywords are added to the **show subscribers** and **clear subscribers** commands. These filters display the impacted subscribers, reduce the impact during switchovers, provide accurate charging, and minimize the detection and recovery time.

Command Changes

This section provides information regarding show commands and/or their outputs in support of this enhancement.

Note If the subscribers' call is up with multiple bearers and if any of the bearers match the specified criteria (without-dynamic-rule or without-override-control), the output is displayed.

show subscribers rulename

The above CLI command is enhanced to include the **rulename** keyword, which refers to the charging rule name. The available rule name options are: predefined, static, and dynamic rules.

An output similar to the **show subscribers without-dynamic-rule** command is displayed, that is, specific to the defined rule name.

show subscribers without-dynamic-rule

The above CLI command is enhanced to include the **without-dynamic-rule** keyword. A sample output for this command is provided as follows:

```
show subscribers without-dynamic-rule
+-----Access  (S) - pdsn-simple-ip     (M) - pdsn-mobile-ip  (H) - ha-mobile-ip
|     Type:   (P) - ggsn-pdp-type-ppp  (h) - ha-ipsec        (N) - lns-l2tp
|             (I) - ggsn-pdp-type-ipv4                       (G) - IPSG
|             (V) - ggsn-pdp-type-ipv6                       (C) - cscf-sip
|             (z) - ggsn-pdp-type-ipv4v6                     (A) - X2GW
|             (R) - sgw-gtp-ipv4       (O) - sgw-gtp-ipv6    (Q) - sgw-gtp-ipv4-ipv6
|             (W) - pgw-gtp-ipv4       (Y) - pgw-gtp-ipv6    (Z) - pgw-gtp-ipv4-ipv6
|             (@) - saegw-gtp-ipv4     (#) - saegw-gtp-ipv6  ($) - saegw-gtp-ipv4-ipv6
|             (&) - samog-ip           (^) - cgw-gtp-ipv6    (*) - cgw-gtp-ipv4-ipv6
|             (p) - sgsn-pdp-type-ppp  (s) - sgsn            (4) - sgsn-pdp-type-ip
|             (6) - sgsn-pdp-type-ipv6 (2) - sgsn-pdp-type-ipv4-ipv6
|             (L) - pdif-simple-ip     (K) - pdif-mobile-ip  (o) - femto-ip
|             (F) - standalone-fa
|             (e) - ggsn-mbms-ue       (U) - pdg-ipsec-ipv4
|             (E) - ha-mobile-ipv6     (T) - pdg-ssl         (v) - pdg-ipsec-ipv6
|             (f) - hnbgw-hnb          (g) - hnbgw-iu        (x) - s1-mme
```

```
|                                    (k) - PCC
|             (X) - HSGW             (n) - ePDG            (t) - henbgw-ue
|             (m) - henbgw-henb      (q) - wsg-simple-ip   (r) - samog-pmip
|             (D) - bng-simple-ip    (l) - pgw-pmip        (3) - GILAN
|             (u) - Unknown
|             (+) - samog-eogre      (%) - eMBMS-ipv4      (!) - eMBMS-ipv6
|
|+----Access  (X) - CDMA 1xRTT       (E) - GPRS GERAN      (I) - IP
||    Tech:   (D) - CDMA EV-DO       (U) - WCDMA UTRAN     (W) - Wireless LAN
||            (A) - CDMA EV-DO REVA  (G) - GPRS Other      (M) - WiMax
||            (C) - CDMA Other       (N) - GAN             (O) - Femto IPSec
||            (P) - PDIF             (S) - HSPA            (L) - eHRPD
||            (T) - eUTRAN           (B) - PPPoE           (F) - FEMTO UTRAN
||                                   (Q) - WSG             (.) - Other/Unknown
||
||+---Call    (C) - Connected        (c) - Connecting
|||   State:  (d) - Disconnecting    (u) - Unknown
|||           (r) - CSCF-Registering (R) - CSCF-Registered
|||           (U) - CSCF-Unregistered
|||
|||+--Access  (A) - Attached         (N) - Not Attached
||||  CSCF    (.) - Not Applicable
||||  Status:
||||
||||+-Link    (A) - Online/Active    (D) - Dormant/Idle
||||| Status:
|||||
|||||+Network (I) - IP               (M) - Mobile-IP       (L) - L2TP
||||||Type:   (P) - Proxy-Mobile-IP  (i) - IP-in-IP        (G) - GRE
||||||        (V) - IPv6-in-IPv4     (S) - IPSEC           (C) - GTP
||||||        (A) - R4 (IP-GRE)      (T) - IPv6            (u) - Unknown
||||||        (W) - PMIPv6(IPv4)     (Y) - PMIPv6(IPv4+IPv6) (R) - IPv4+IPv6
||||||        (v) - PMIPv6(IPv6)     (/) - GTPv1(For SAMOG)  (+) - GTPv2(For SAMOG)
||||||
||||||
vvvvvv CALLID   MSID            USERNAME              IP
TIME-IDLE
------ -------- --------------- --------------------- ----------------------------- ---------
ZTCNAR 004c9963 404005123456789 9890098900@cisco.com   6001::3:0:0:4c99:6301,22.22.0.3
00h00m29s
ZTCNAR 004c9963 404005123456789 9890098900@cisco.com   6001::3:0:0:4c99:6301,22.22.0.3
00h00m29s
ZTCNAR 004c9963 404005123456789 9890098900@cisco.com   6001::3:0:0:4c99:6301,22.22.0.3
00h00m29s

Total subscribers matching specified criteria: 3
```

show subscribers without-override-control-rule

The above CLI command is enhanced to include the **without-override-control-rule** keyword.

An output similar to the **show subscribers without-dynamic-rule** command is displayed, that is, without the override control rule.

show subscribers apn rulename

The above CLI command is enhanced to include the **rulename** <*rule_name*> keyword, which refers to the charging rule name. The available rule name options are: predefined, static, and dynamic rules.

An output similar to the **show subscribers without-dynamic-rule** command is displayed, that is, specific to the defined rule name and APN.

show subscribers apn without-dynamic-rule

The above CLI command is enhanced to include the **without-dynamic-rule** keyword.

An output similar to the **show subscribers without-dynamic-rule** command is displayed, that is, specific to the APN.

show subscribers apn without-override-control-rule

The above CLI command is enhanced to include the **without-override-control-rule** keyword.

An output similar to the **show subscribers without-dynamic-rule** command is displayed for the specified APN, which does not have the override control rule configured.

show subscribers summary rulename

The above CLI command is enhanced to include the **rulename** keyword, which refers to the charging rule name. The available rule name options are: predefined, static, and dynamic rules.

An output similar to the **show subscribers summary without-dynamic-rule** command is displayed, that is, specific to the defined rule name.

show subscribers summary without-dynamic-rule

The above CLI command is enhanced to include the **without-dynamic-rule** keyword. A sample output for this command is provided as follows:

```
show subscribers summary without-dynamic-rule
Total Subscribers:             1
Active:                        1            Dormant:                       0
LAPI Devices:                  0
pdsn-simple-ipv4:              0            pdsn-simple-ipv6:              0
pdsn-mobile-ip:                0            ha-mobile-ipv6:                0
hsgw-ipv6:                     0            hsgw-ipv4:                     0
hsgw-ipv4-ipv6:                0            pgw-pmip-ipv6:                 0
pgw-pmip-ipv4:                 0            pgw-pmip-ipv4-ipv6:            0
pgw-gtp-ipv6:                  0            pgw-gtp-ipv4:                  0
pgw-gtp-ipv4-ipv6:             3            sgw-gtp-ipv6:                  0
sgw-gtp-ipv4:                  0            sgw-gtp-ipv4-ipv6:             0
sgw-pmip-ipv6:                 0            sgw-pmip-ipv4:                 0
sgw-pmip-ipv4-ipv6:            0            pgw-gtps2b-ipv4:               0
pgw-gtps2b-ipv6:               0            pgw-gtps2b-ipv4-ipv6:          0
pgw-gtps2a-ipv4:               0            pgw-gtps2a-ipv6:               0
pgw-gtps2a-ipv4-ipv6:          0
mme:                           0            mme-embms:                     0
henbgw-ue:                     0            henbgw-henb:                 0
x2gw-enb:                      0
ipsg-rad-snoop:                0            ipsg-rad-server:               0
ha-mobile-ip:                  0            ggsn-pdp-type-ppp:             0
ggsn-pdp-type-ipv4:            0            lns-l2tp:                      0
ggsn-pdp-type-ipv6:            0            ggsn-pdp-type-ipv4v6:          0
ggsn-mbms-ue-type-ipv4:        0
pdif-simple-ipv4:              0
```

```
pdif-simple-ipv6:               0            pdif-mobile-ip:                 0
wsg-simple-ipv4:                0            wsg-simple-ipv6:                0
pdg-simple-ipv4:                0            ttg-ipv4:                       0
pdg-simple-ipv6:                0            ttg-ipv6:                       0
femto-ip:                       0
epdg-pmip-ipv6:                 0            epdg-pmip-ipv4:                 0
epdg-pmip-ipv4-ipv6:            0
epdg-gtp-ipv6:                  0            epdg-gtp-ipv4:                  0
epdg-gtp-ipv4-ipv6:             0
sgsn:                           0            sgsn-pdp-type-ppp:              0
sgsn-pdp-type-ipv4:             0            sgsn-pdp-type-ipv6:             0
sgsn-pdp-type-ipv4-ipv6:        0            type not determined:            0
sgsn-subs-type-gn:              0            sgsn-subs-type-s4:              0
sgsn-pdp-type-gn:               0            sgsn-pdp-type-s4:               0
cdma 1x rtt sessions:           0            cdma evdo sessions:             0
cdma evdo rev-a sessions:       0            cdma 1x rtt active:             0
cdma evdo active:               0            cdma evdo rev-a active:         0
hnbgw:                          0            hnbgw-iu:                       0
bng-simple-ipv4:                0
pcc:                            0
in bytes dropped:               0            out bytes dropped:              0
in packet dropped:              0            out packet dropped:             0
in packet dropped zero mbr:     0            out packet dropped zero mbr:    0
in bytes dropped ovrchrgPtn:    0            out bytes dropped ovrchrgPtn:   0
in packet dropped ovrchrgPtn:   0            out packet dropped ovrchrgPtn:  0
ipv4 ttl exceeded:              0            ipv4 bad hdr:                   0
ipv4 bad length trim:           0
ipv4 frag failure:              0            ipv4 frag sent:                 0
ipv4 in-acl dropped:            0            ipv4 out-acl dropped:           0
ipv4 in-mcast pkt dropped:      0            ipv4 in-bcast pkt dropped:      0
ipv6 bad hdr:                   0            ipv6 bad length trim:           0
ipv6 in-acl dropped:            0            ipv6 out-acl dropped:           0
ipv4 in-css-down dropped:       0            ipv4 out-css-down dropped:      0
ipv4 out xoff pkt dropped:      0            ipv6 out xoff pkt dropped:      0
ipv4 xoff bytes dropped:        0            ipv6 xoff bytes dropped:        0
ipv4 out no-flow dropped:       0
ipv4 early pdu rcvd:            0            ipv4 icmp packets dropped:      0
ipv6 input ehrpd-access drop:   0            ipv6 output ehrpd-access drop:  0
dormancy count:                 0            handoff count:                  0
pdsn fwd dynamic flows:         0            pdsn rev dynamic flows:         0
fwd static access-flows:        0            rev static access-flows:        0
pdsn fwd packet filters:        0            pdsn rev packet filters:        0
traffic flow templates:         0
```

show subscribers summary without-override-control-rule

The above CLI command is enhanced to include the **without-override-control-rule** keyword.

An output similar to the **show subscribers summary without-dynamic-rule** command is displayed, that is, without the override control rule.

clear subscribers apn rulename

The above CLI command is enhanced to include the **rulename** keyword, which refers to the charging rule name. The available rule name options are: predefined, static, and dynamic rules.

An output similar to the **clear subscribers apn without-dynamic-rule** or **clear subscribers apn without-dynamic-rule** command is displayed based on whether subscribers for a specific APN exist with the specified rule name.

clear subscribers apn without-dynamic-rule

The above CLI command is enhanced to include the **without-dynamic-rule** keyword. A sample output for this command is provided as follows:

```
clear subscribers apn cisco.com without-dynamic-rule
No subscribers match the specified criteria
```

clear subscribers apn without-override-control-rule

The above CLI command is enhanced to include the **without-override-control-rule** keyword.

A sample output for this command is provided as follows:

```
clear subscribers apn cisco.com without-override-control-rule
Number of subscribers cleared: 1
```