Cisco ASR 5000 Series Peer-to-Peer Detection Administration Guide

Version 10.0

Last Updated June 30, 2010

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

Text Part Number: OL-22960-01

Cisco ASR 5000 Series Peer-to-Peer Detection Administration Guide

CONTENTS

About this Guide

This document pertains to features and functionality that run on and/or that are related to the Cisco® ASR 5000 Chassis, formerly the Starent Networks ST40.

Conventions Used

The following tables describe the conventions used throughout this documentation.

Icon	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.
	Electro-Static Discharge (ESD)	Alerts you to take proper grounding precautions before handling a product.

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

Command Syntax Conventions	Description
{ **`keyword`** or *`variable`* }	Required keywords and variables are surrounded by grouped brackets. Required keywords and variables are those components that are required to be entered as part of the command syntax.

Command Syntax Conventions	Description
[**keyword** or *variable*]	Optional keywords or variables, or those that a user may or may not choose to use, are surrounded by square brackets.
|	With some commands there may be a group of variables from which the user chooses one. These are called alternative variables and are documented by separating each variable with a vertical bar (also known as a pipe filter). Pipe filters can be used in conjunction with required or optional keywords or variables. For example: **{ nonce | timestamp }** OR [**count** *number_of_packets* | **size** *number_of_bytes*]

Contacting Customer Support

Use the information in this section to contact customer support.

For New Customers: 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 is required to this site. Please contact your local sales or service representative for additional information.

For Existing Customers with support contracts through Starent Networks: Refer to the support area of https://support.starentnetworks.com/ for up-to-date product documentation or to submit a service request. A valid username and password is required to this site. Please contact your local sales or service representative for additional information.

IMPORTANT: For warranty and repair information, please be sure to include the Return Material Authorization (RMA) tracking number on the outside of the package.

Chapter 1
Peer-to-Peer Overview

This chapter provides an overview of the Peer-to-Peer (P2P) in-line services.

The System Administration Guide provides basic system configuration information, and the product administration guides provide procedures to configure basic functionality of core network service. It is recommended that you select the configuration example that best meets your service model, and configure the required elements for that model, as described in the respective product Administration Guide, before using the procedures in this chapter.

This chapter covers the following topics:

Supported Platforms and Products

P2P is an in-line service supported on ASR 5000 running 3GPP, 3GPP2, LTE and WiMAX core network services.

Licenses

P2P is a licensed feature, requiring the [600-00-7605] *Peer-to-Peer Detection Bundle 1k Sessions* license. For information on core network licenses and other requirements, please contact your local sales representative.

For information on license requirements for any customer-specific features, please contact your local sales/service representative.

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

P2P Overview

P2P is a term used in two slightly different contexts. At a functional level, it means protocols that interact in a peering manner, in contrast to client-server manner. There is no clear differentiation between the function of one node or another. Any node can function as a client, a server, or both—a protocol may not clearly differentiate between the two. For example, peering exchanges may simultaneously include client and server functionality, sending and receiving information. P2P utilizes the Enhanced Charging Service (ECS) functionality. For information about ECS, refer to the *Enhanced Charging Services Administration Guide*

Detecting P2P protocols requires recognizing, in real time, some uniquely identifying characteristic of the protocols. Typical packet classification only requires information uniquely typed in the packet header of packets of the stream(s) running the particular protocol to be identified. In fact, many P2P protocols can be detected by simple packet header inspection. However, some P2P protocols are different, preventing detection in the traditional manner. This is designed into some P2P protocols to purposely avoid detection. The creators of these protocols purposely do not publish specifications. A small class of P2P protocols is stealthier and more challenging to detect. For some protocols, no set of fixed markers can be identified with confidence as unique to the protocol.

Operators care about P2P traffic because of the behavior of some P2P applications (for example, Bittorrent, Skype, and eDonkey). Most P2P applications can hog the network bandwidth such that 20% P2P users can generate as much traffic as generated by the rest 80% non-P2P users. This can result into a situation where non-P2P users may not get enough network bandwidth for their legitimate use because of excess usage of bandwidth by the P2P users. Network operators need to have dynamic network bandwidth / traffic management functions in place to ensure fair distributions of the network bandwidth among all the users. And this would include identifying P2P traffic in the network and applying appropriate controlling functions to the same (for example, content-based premium billing, QoS modifications, and other similar treatments).

The P2P detection technology makes use of innovative and highly accurate protocol behavioral detection techniques. This P2P solution can detect the following protocols and their capabilities in real time:

- ActiveSync
- Aimini
- AppleJuice
- Ares
- Battlefield
- BitTorrent
    - File downloading and uploading (plain / encrypted BitTorrent)
    - Un-encrypted, plain-encrypted, and RC4-encrypted file transfer
- Ddlink
- DirectConnect
- eDonkey
    - File uploading and downloading (plain / encrypted eDonkey)
- FastTrack
- Feidian
- FileTopia

- Freenet
- Fring
- Gadu-Gadu
- Gnutella
- Google Talk
    - Voice
    - Non-voice
- Half-Life 2
- HamachiVPN
- IAX
- iMesh
- IPTV
- IRC
- iSkoot
- Jabber
- Manolito
- MSN
    - Voice
    - Non Voice
- Mute
- Nimbuzz
- ooVoo
- OpenFT
- Orb
- Oscar / AoL
    - Voice
    - Non Voice
- Paltalk
- Pando
- Pandora
- PoPo
- PPLive
- PPStream
- QQ
- QQgame
- QQLive
- Quake

- RDP
- SecondLife
- Skinny
- Skype
    - Voice
    - Non Voice
- Slingbox
- SopCast
- SoulSeek
- Steam
- TVAnts
- TVUPlayer
- UUSee
- VPN-X
- VTun
- Warcraft3
- WinMX
- Winny
- World of Warcraft
- Xbox
- Yahoo
    - Voice
    - Non Voice
- Zattoo

When P2P protocols are detected, statistics reporting and postpaid charging policy are supported. Per-protocol statistics via bulkstats and via report records including:

- UDR types: Summarizing data usage for a given content type
- EDR types: Specific to a particular event
- e-GCDRs: Specific to 3GPP

Upon detection of a P2P protocol for a particular flow, one of the following actions can be applied:

- Blocking P2P traffic—blocking protocol(s) and discarding traffic
- Bandwidth policing—limiting the bandwidth, applied per PDP context per P2P application type
- Flow policing—limiting the number of simultaneous P2P flows
- QoS support—including policing
- TOS marking—applied per P2P protocol type

- Prepaid and postpaid charging support for the following P2P protocols:
  - ActiveSync
  - AppleJuice
  - Ares
  - Battlefield
  - BitTorrent
  - DirectConnect
  - eDonkey
  - FastTrack
  - Filetopia
  - Fring
  - Gadu-Gadu
  - Gnutella
  - Google Talk
  - iMesh
  - IRC
  - iSkoot
  - Jabber
  - Manolito
  - MSN voice/non-voice
  - Mute
  - Nimbuzz
  - ooVoo
  - Orb
  - Oscar
  - Paltalk
  - Pando
  - PoPo
  - PPLive
  - PPStream
  - QQ
  - QQLive
  - Skype voice/non-voice
  - Slingbox
  - SopCast
  - SoulSeek
  - UUSee

  - Winny
  - Yahoo voice/non-voice
  - Zattoo
- Prepaid and postpaid P2P content-based billing
- Statistics reporting—analyzing per-protocol statistics using bulkstats

P2P Voice Call Duration

The P2P product has the capability to detect network traffic created by P2P VoIP clients such as Skype, Yahoo, MSN, Gtalk, Oscar. The VoIP call duration is a direct indication to the revenue impact of the network operator. The P2P product is well poised to process the network traffic online to detect and control the VoIP presence, and generate records that can be used to calculate the VoIP call durations.

Random Drop Charging Action

The random drop charging action is added as an option to degrade P2P voice calls. This is achieved by randomly dropping packets of the voice calls over the voice call period.

Voice data is encoded in multiple packets by the codec. Since there is a possibility of packets being dropped in a network, the codec replicates the same information across multiple packets. This provides resilience to random packet drops in the network. For a considerable degradable voice quality, a chunk of packets need to be dropped. By this way, the codec will be unable to decode the required voice information. The chunk size for achieving degradation of voice call varies from one protocol to another.

The Random Drop decision has to be made once for a chunk of packets. By choosing the random drop time from a configured range, the drop is achieved at random seconds within a configured range. The packets will drop within a known period of time. For example, if a voice call happens for 2 minutes and if we configure a drop interval of 12–15 seconds, then a packet will be dropped within the first 15 seconds of the voice call.

IMPORTANT: This feature is applicable only for VOIP calls.

Dynamic Signature Updates

P2P traffic detection is tricky because most of the P2P protocol details are proprietary, and the protocol characteristics change frequently. As these P2P standards are proprietary, there is a tight coupling between the peers too (all the peers need to understand the protocols). Since P2P detection depends heavily on the known traffic characteristics the detection can suffer if the P2P protocol changes, if some existing traffic characteristics were not known (new use case scenarios), if one P2P traffic characteristic matches with another P2P traffic (false positives), and if there are flaws (bugs) in the detection logic. Whenever such degradation in P2P detection logic is identified, the P2P detection engine needs to be fine tuned or enhanced further to improve the detection accuracy.

In the earlier releases, the P2P detection logic was part of the chassis software load (ASR 5000 software), to continue to detect new traffic patterns based on the changing traffic characteristics, operators needed to upgrade the complete software with the updated logic.

This release supports dynamic upgrades of the P2P detection logic (signatures) alone on an active ASR 5000 without warranting a full software upgrade, and hence without a software restart or reboot. This is implemented through signature files.

IMPORTANT: This release supports dynamic upgrades of detection logic for the following P2P protocols: Bittorrent, DirectConnect, eDonkey, Gnutella, Skype, and Yahoo.

IMPORTANT: Dynamic signature updates may not work in all situations, and software updates may be required to update the detection logic in use on a system.

In an initial software build, all the detection logic is embedded in the code. If in a subsequent software build, there are updates to the detection logic, the changes are made available as a P2P signature file. If the initial build supports the Dynamic Signature Updates feature, this signature file can be loaded on the system to update the detection capability.

In case a P2P signature file is already available for a software build, when the configuration file is loaded on the system, it will take the lastest version. If a different P2P signature file is manually loaded on that system, every time the system reboots, it will load the default version.

A P2P signature file can support upgrade for multiple P2P protocols that are enabled for dynamic upgrade. Operators can selectively upgrade the detection for specific protocol(s). Patches can be rolled down with out any negative impact to the system. If an incorrect signature file is loaded by mistake, the version information in signature file will not match the current protocol detection version and the system will not be affected.

The signature files are provided on a need basis, or periodically whenever a new P2P detection software version is integrated with the software. A signature file can contain the rules for several protocols. The P2P signature file is packaged as a delivery kit for release. For more information, contact your local sales representative.

P2P Protocol Detection Software Versions

Every released signature file has a file version. This version number is used to determine which file is the latest and newest to load during upgrade or reboot. On the boxer, the signature file version and the syntax is validated, in case of failure, the signatures will not be loaded into memory.

Enabling and Disabling P2P Dynamic Signature Updates

The P2P Dynamic Signature Update feature can be enabled and disabled from the CLI.

Disabling the P2P Dynamic Update feature instructs the system not to load and apply the signature files. An already loaded signature file can be unloaded (removed) from the system's memory too.

CLI show commands can be used to view details of loaded signature file, and the P2P as well as the individual protocol detection software versions.

Loading and Unloading P2P Signature File

Loading Signature File

If a P2P signature file is already available for a software build, the system loads the file from the default location, which is "*/usr/lib/p2p-rules.xml*".

Operators can load P2P signature files present in the system's Flash directory from the CLI. A P2P signature file loaded from the Flash directory must always be available in the Flash directory. In this case, based on the signature files' version numbers, the P2P engine loads the latest file available between the default file and the new file specified in the configuration.

Loading of rules is a two-stage process. First, from the signature file the signatures are loaded to all the Session Managers (SessMgrs). Once all the SessMgrs are able to parse the signatures successfully, the signatures are activated. If any SessMgr reports failure in parsing the signatures, the activation will not be done. A deactivate message will be sent to the managers so that any SessMgrs that successfully parsed the signatures will unload them.

When, on a system, the signature file containing the rules are loaded for the first time, new calls generated after loading the rules would use these rules.

There can only be a maximum of two signature files loaded on the system's memory at any point of time. If a loaded signature file has active calls, and the operator loads a newer version of the rule file, the older file will be removed from the memory once all the calls referring to it have ended. All calls generated after loading the new file will use the newer file.

Considering the memory used for loading the signature files, the number of active versions that can be loaded is restricted to two. Suppose we currently have a patch D1 loaded and running, and have an update D2. After loading D2 in memory, D1 will still be active in memory because there may be some call lines using this version. Loading a new patch D3 has to wait till D1 is removed from the memory.

IMPORTANT: In case of session recovery, when subscriber call is recovered, it will always use the active version of the P2P signature file available in the memory.

Unloading Signature File

When a signature file is unloaded from the CLI, the SessCtrl sends request to all the SessMgrs to unload the file from memory. The SessMgr maintains the reference count for the version loaded into the memory. If the reference count is zero, the rules are deleted from the memory. If there are some sessions using the version to be unloaded, the version is marked for unloading. When there are no references to the version, it is deleted from the memory.

How P2P Works

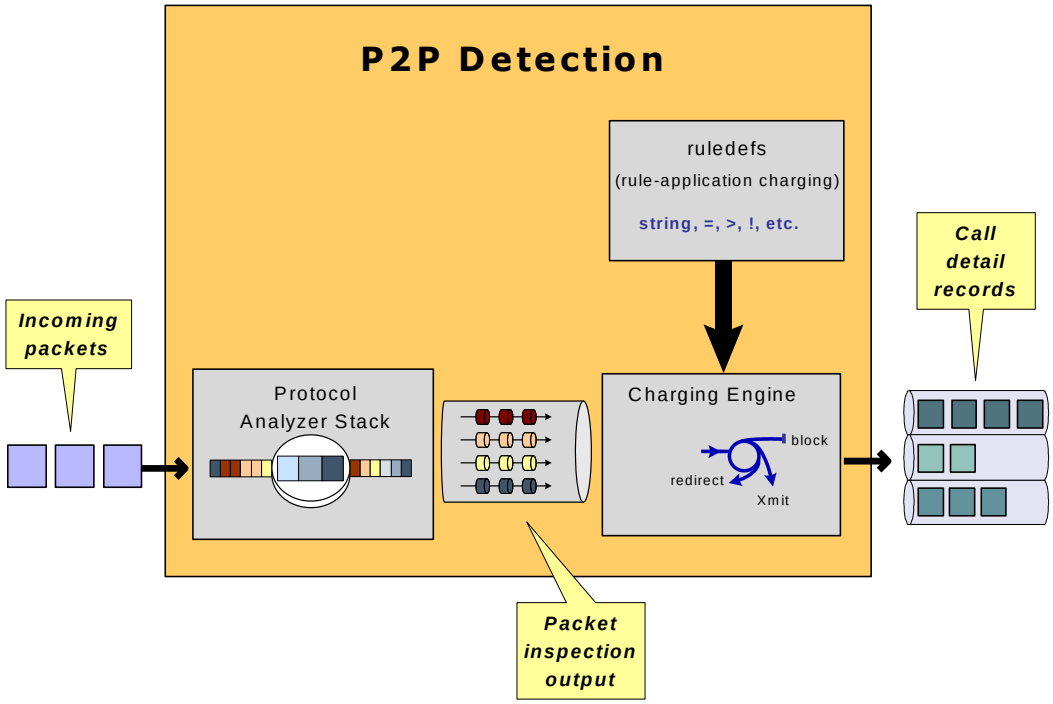
P2P interfaces to a PCRF Diameter Gx interface to accept policy ACLs and rulebases from a PDF. P2P supports real-time dynamic policy updates during a subscriber session. This includes modifying the subscriber's policy rules during an active session by means of ACL name and Rulebase name.

In Rel. 7 Gx interface, a Charging Rulebase will be treated as a group of ruledefs. A group of ruledefs enables grouping rules into categories, so that charging systems can base the charging policy on the category. When a request contains names of several Charging Rulebases, groups of ruledefs of the corresponding names are activated. For P2P rules to work in the group of ruledefs, P2P detection has to be enabled in the rulebase statically.

Static policy is supported initially. A default subscriber profile is assumed and can be overwritten on the gateway. Per-subscriber static policy is pulled by the gateway from the AAA service at subscriber authentication.

The following figure illustrates how packets travel through the system using P2P detection. The packets are investigated and then handled appropriately using ruledefs for charging.

Figure 1. Overview of Packet Processing in ECSv2



Advantages of P2P Processing Before DPI

- Some protocols like BitTorrent and Orb use HTTP traffic for initial setup. If P2P analysis is done after HTTP, it is possible that these protocols may go undetected.
- Protocols like Skype use well known ports (like 80 & 443). In these scenarios, the HTTP engine reports these as invalid packets. For protocol detection, it is desirable to have P2P detection before Deep Packet Inspection (DPI).
- Stateless detection of protocols based on signature will be easier when the P2P analysis is done before DPI.

P2P Session Recovery

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. ACS recovery is accomplished using this checkpointed information.

IMPORTANT: In order for session recovery to work there should be at least four packet processing cards (PSCs/PSC2s), 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 PSC/PSC2.

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 PSC/PSC2 Failure

When a packet processing card 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.

Limitations

This section lists the limitations of P2P detection in this release.

Skype

- The Skype detection cannot detect traffic of most of the third-party plug-ins. The plug-ins use Skype only for marketing and presentation purposes such as opening a window within a Skype window or modifying the main Skype window with buttons or sounds. These plug-ins do NOT use the Skype protocol to transfer data over the network.
- Other than Skype Voice, all detected Skype traffic is marked as Skype. Distinctions between different data types within Skype (i.e. text chat, file transfer, and so on) cannot be made.
- Skype voice detection may not be accurate if it happens with other traffic (file transfer, video, etc.) on the same flow.

eDonkey

- The eDonkey client eMule supports a protocol named Kademlia. This protocol is an implementation of a DHT (Distributed Hash Table). Kademlia is only used for searching new peers which have the file the user wants to download. The download itself uses the eDonkey protocol. However, the Kademlia protocol is not detected as eDonkey.
- The eDonkey client eMule supports a text chat that is not detected as eDonkey.

Yahoo

Yahoo! HTTP downloads for yahoo games, images and ads that come during yahoo messenger startup are not detected as Yahoo!. If configured, these can be passed on to the HTTP analyzer for HTTP Deep Packet Inspection.

MSN

MSN HTTP downloads such as MSN Games and MSN Applications are not detected. Traffic from these MSN applications use a different protocol for their traffic.

BitTorrent

- Some clients (like Azureus 3.0) provide an advanced user interface which can include an embedded web browser. These are not detected as BitTorrent. Also other features like chat or instant messaging are not detected as BitTorrent. These features are client specific and not related to the BitTorrent protocol.
- Certain clients also display advertisements. These images are downloaded through plain HTTP and are not detected as BitTorrent.

Jabber

- Most clients that use Jabber for IM offer other services like Voice Call or File Transfer. These services are not detected as Jabber.
- Jabber with SSL encryption cannot be detected, because it uses SSL.

Gnutella / Morpheus

- Some of the clients that use Gnutella protocol for file sharing can also use other file sharing protocols. The part of traffic that follows Gnutella Protocol will only be detected as Gnutella.
- Client specific patterns which are not part of the Gnutella Protocol will not be detected as Gnutella. UDP contributes to about 20-30 % of most Gnutella clients. Detection is based on some strange patterns in the first packet of these UDP flows. Untested Gnutella clients may have more strange patterns, causing drop in the detection %.
- The Morpheus Client creates a lot of TCP flows, without any string pattern in the application header. These flows are not currently detected.

Winny

The Winny client also supports bbs. This is currently not detected.

FastTrack

SSL packets and HTTP packets from the Kazaa client is not detected. Only data transfer is detected.

Gadu-Gadu

Radio traffic passes through HTTP and is not detected.

Other Limitations

- Most of the heuristic analysis for a subscriber is stateful and depends on building an internal state based on certain patterns seen by the analyzer. Patterns occur over multiple packets in a single flow and over multiple flows for a subscriber. If the system loses the state (due to a task failure for example), then the detection can fail for the affected subscribers/flows after recovery.

Most P2P protocols emit these patterns regularly (sometimes as early as the next flow created by the application). When the system sees the pattern again, it re-learns the subscriber state and starts detecting the protocol.

- In this release, P2P rules cannot be combined with UDP and TCP rules in one ruledef.

Chapter 2
Peer-to-Peer Detection Configuration

This chapter describes how to configure the Peer-to-Peer (P2P) Detection feature.

The following topics are covered in this chapter:

Configuring System for P2P Detection Support

This section lists the high-level steps to configuring the system with enhanced charging services for P2P Detection support in conjunction with ECS services.

To configure the system for P2P Detection support with ECS:

Step 1 Set initial configuration parameters such as activating PACs/PSCs and modifying the local context as described in the Initial Configuration section.

Step 2 Enable the Enhanced Charging service with P2P and set basic ECS parameters such as service configuration, Ruledefs, charging actions, and EDRs as described in the P2P Detection Configuration section.

Step 3 Save the changes to system configuration as described in the Save the Configuration section.

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.

Initial Configuration

To perform initial system configuration for P2P detection support with ECS:

Step 1 Specify the role of the PACs/PSCs in the chassis as described in the Activating PACsPSCs section.

Step 2 Enable ACS as described in the Enabling Enhanced Charging section.

Step 3 Set local system management parameters as described in the Modifying the Local Context section.

Activating PACs/PSCs

Use the following configuration example to activate two PACs/PSCs, placing one in “active” mode and labeling the other as redundant:

```
configure
        card <slot_number>
           redundancy card-mode [ -noconfirm ]
           exit
```

```
card <slot_number>
   mode active pac/psc
   end
```

Enabling Enhanced Charging

Use the following configuration example to enable enhanced charging on the system:

```
configure
        require active-charging
        end
```

Modifying the Local Context

Use the following configuration example to set the default subscriber and AAA group in the local context:

```
configure
        context local
            interface <interface>
              ip address <address/mask>
              ip arp timeout <timeout>
              exit
           server ftpd
           exit
           server sshd
              subsystem sftp
              exit
           server telnetd
              exit
           subscriber default
              exit
           administrator <security_admin> encrypted password <password> ftp
```

```
    aaa group default
       exit
    gtpp group default
       exit
    ip route <route> SPIO1
    exit
    port ethernet <slot/port>
       no shutdown
       bind interface <interface> local
    exit
snmp engine-id local <id_number>
end
```

P2P Detection Configuration

To configure P2P Detection with ACS:

Step 1 Create the ACS service as described in the Creating the Active Charging Service section.

Step 2 Configure P2P detection rules as described in the Configuring P2P Detection Rules section.

Step 3 Configure the charging action as described in the Configuring the Charging Action section.

Step 4 Configure the rulebase as described in the Configuring the Rulebase section.

Step 5 *Optional:* Set EDR formats as described in the Setting EDR Formats section.

Step 6 Enable DSCP settings as described in the Enable DSCP Marking section.

Step 7 *Optional:* Configure P2P Dynamic Signature Updates functionality as described in the Configuring P2P Dynamic Signature Updates section.

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 Active Charging Service

Use the following configuration example to create the ACS service:

```
configure
         active-charging service <acs_service_name> [ -noconfirm ]
         end
```

Configuring P2P Detection Rules

Use the following configuration example to set the P2P detection protocols in the ACS and the rule definitions for each P2P protocol. Note that the following example includes configuration to report voice and non-voice components for Skype, Yahoo, and MSN separately.

```
configure
         active-charging service <acs_service_name>
            p2p-detection protocol all
            ruledef <charging_ruledef_actsync>
               p2p protocol = actsync
               exit
            ruledef <charging_ruledef_aimini>
               p2p protocol = aimini
               exit
            ruledef <charging_ruledef_applejuice>
               p2p protocol = applejuice
               exit
            ruledef <charging_ruledef_ares>
               p2p protocol = ares
               exit
            ruledef <charging_ruledef_battlefld>
               p2p protocol = battlefld
               exit
            ruledef <charging_ruledef_bittorrent>
```

```
   p2p protocol = bittorrent
   exit
ruledef <charging_ruledef_ddlink>
   p2p protocol = ddlink
   exit
ruledef <charging_ruledef_directconnect>
   p2p protocol = directconnect
   exit
ruledef <charging_ruledef_edonkey>
   p2p protocol = edonkey
   exit
ruledef <charging_ruledef_fasttrack>
   p2p protocol = fasttrack
   exit
ruledef <charging_ruledef_feidian>
   p2p protocol = feidian
   exit
ruledef <charging_ruledef_filetopia>
   p2p protocol = filetopia
   exit
ruledef <charging_ruledef_freenet>
   p2p protocol = freenet
   exit
ruledef <charging_ruledef_fring>
   p2p protocol = fring
   exit
ruledef <charging_ruledef_gadugadu>
   p2p protocol = gadugadu
   exit
```

```
ruledef <charging_ruledef_gnutella>
   p2p protocol = gnutella
   exit
ruledef <charging_ruledef_gtalk>
   p2p protocol = gtalk
   exit
ruledef <charging_ruledef_halflife2>
   p2p protocol = halflife2
   exit
ruledef <charging_ruledef_hamachivpn>
   p2p protocol = hamachivpn
   exit
ruledef <charging_ruledef_iax>
   p2p protocol = iax
   exit
ruledef <charging_ruledef_imesh>
   p2p protocol = imesh
   exit
ruledef <charging_ruledef_iptv>
   p2p protocol = iptv
   exit
ruledef <charging_ruledef_irc>
   p2p protocol = irc
   exit
ruledef <charging_ruledef_iskoot>
   p2p protocol = iskoot
   exit
ruledef <charging_ruledef_jabber>
   p2p protocol = jabber
```

```
   exit
ruledef <charging_ruledef_manolito>
   p2p protocol = manolito
   exit
ruledef <charging_ruledef_msn>
   p2p protocol = msn
   exit
ruledef <charging_ruledef_mute>
   p2p protocol = mute
   exit
ruledef <charging_ruledef_nimbuzz>
   p2p protocol = nimbuzz
   exit
ruledef <charging_ruledef_oovoo>
   p2p protocol = oovoo
   exit
ruledef <charging_ruledef_openft>
   p2p protocol = openft
   exit
ruledef <charging_ruledef_orb>
   p2p protocol = orb
   exit
ruledef <charging_ruledef_oscar>
   p2p protocol = oscar
   exit
ruledef <charging_ruledef_paltalk>
   p2p protocol = paltalk
   exit
ruledef <charging_ruledef_pando>
```

```
   p2p protocol = pando
   exit
ruledef <charging_ruledef_pandora>
   p2p protocol = pandora
   exit
ruledef <charging_ruledef_popo>
   p2p protocol = popo
   exit
ruledef <charging_ruledef_pplive>
   p2p protocol = pplive
   exit
ruledef <charging_ruledef_ppstream>
   p2p protocol = ppstream
   exit
ruledef <charging_ruledef_qq>
   p2p protocol = qq
   exit
ruledef <charging_ruledef_qqgame>
   p2p protocol = qqgame
   exit
ruledef <charging_ruledef_qqlive>
   p2p protocol = qqlive
   exit
ruledef <charging_ruledef_quake>
   p2p protocol = quake
   exit
ruledef <charging_ruledef_rdp>
   p2p protocol = rdp
   exit
```

```
ruledef <charging_ruledef_secondlife>
   p2p protocol = secondlife
   exit
ruledef <charging_ruledef_skinny>
   p2p protocol = skinny
   exit
ruledef <charging_ruledef_skype>
   p2p protocol = skype
   exit
ruledef <charging_ruledef_slingbox>
   p2p protocol = slingbox
   exit
ruledef <charging_ruledef_sopcast>
   p2p protocol = sopcast
   exit
ruledef <charging_ruledef_soulseek>
   p2p protocol = soulseek
   exit
ruledef <charging_ruledef_steam>
   p2p protocol = steam
   exit
ruledef <charging_ruledef_tvants>
   p2p protocol = tvants
   exit
ruledef <charging_ruledef_tvuplayer>
   p2p protocol = tvuplayer
   exit
ruledef <charging_ruledef_uusee>
   p2p protocol = uusee
```

```
   exit
ruledef <charging_ruledef_vpnx>
   p2p protocol = vpnx
   exit
ruledef <charging_ruledef_vtun>
   p2p protocol = vtun
   exit
ruledef <charging_ruledef_warcft3>
   p2p protocol = warcft3
   exit
ruledef <charging_ruledef_winmx>
   p2p protocol = winmx
   exit
ruledef <charging_ruledef_winny>
   p2p protocol = winny
   exit
ruledef <charging_ruledef_wofwarcraft>
   p2p protocol = wofwarcraft
   exit
ruledef <charging_ruledef_xbox>
   p2p protocol = xbox
   exit
ruledef <charging_ruledef_yahoo>
   p2p protocol = yahoo
   exit
ruledef <charging_ruledef_zattoo>
   p2p protocol = zattoo
   exit
```

```
# Configuration to report voice and non-voice components for GTalk, MSN, Oscar,
Skype, and Yahoo separately:
            ruledef <charging_ruledef_gtalk_voice>
               p2p protocol = gtalk
               p2p traffic-type = voice
               exit
            ruledef <charging_ruledef_gtalk_non_voice>
               p2p protocol = gtalk
               p2p traffic-type != voice
               exit
            ruledef <charging_ruledef_msn_voice>
               p2p protocol = msn
               p2p traffic-type = voice
               exit
            ruledef <charging_ruledef_msn_non_voice>
               p2p protocol = msn
               p2p traffic-type != voice
               exit
            ruledef <charging_ruledef_oscar_voice>
               p2p protocol = oscar
               p2p traffic-type = voice
               exit
            ruledef <charging_ruledef_oscar_non_voice>
               p2p protocol = oscar
               p2p traffic-type != voice
               exit
            ruledef <charging_ruledef_skype_voice>
               p2p protocol = skype
               p2p traffic-type = voice
```

```
   exit
ruledef <charging_ruledef_skype_non_voice>
   p2p protocol = skype
   p2p traffic-type != voice
   exit
ruledef <charging_ruledef_yahoo_voice>
   p2p protocol = yahoo
   p2p traffic-type = voice
   exit
ruledef <charging_ruledef_yahoo_non_voice>
   p2p protocol = yahoo
   p2p traffic-type != voice
   exit
ruledef <charging_ruledef_non_voice>
   p2p traffic-type = voice
   exit
ruledef <charging_ruledef_voice>
   p2p traffic-type != voice
   exit
ruledef <routing_ruledef_dns-tcp>
   tcp either-port = 53
   rule-application routing
   exit
ruledef <routing_ruledef_dns-udp>
   udp either-port = 53
   rule-application routing
   exit
ruledef <routing_ruledef_ftp-control>
   tcp either-port = 21
```

```
    rule-application routing
    exit
ruledef <routing_ruledef_ftp-data>
    tcp either-port = 20
    rule-application routing
    exit
ruledef <routing_ruledef_http>
    tcp either-port = 80
    rule-application routing
    exit
ruledef <routing_ruledef_https>
    tcp either-port = 443
    rule-application routing
    exit
ruledef <routing_ruledef_imap>
    tcp either-port = 143
    rule-application routing
    exit
ruledef <routing_ruledef_mms-wapcl-ct>
    wsp content type = application/vnd.wap.mms-message
    rule-application routing
    exit
ruledef <routing_ruledef_mms_http_ct>
    http content type = application/vnd.wap.mms-message
    rule-application routing
    exit
ruledef <routing_ruledef_mms_http_url>
    http url ends-with .mms
    rule-application routing
```

```
   exit
ruledef <routing_ruledef_mms_wapcl-url>
   wsp url ends-with .mms
   rule-application routing
   exit
ruledef <routing_ruledef_pop3>
   tcp either-port = 110
   rule-application routing
   exit
ruledef <routing_ruledef_rtsp>
   tcp either-port = 554
   rule-application routing
   exit
ruledef <routing_ruledef_rtsp-8556>
   tcp either-port = 8556
   rule-application routing
   exit
ruledef <routing_ruledef_sdp>
   sip content type = application/sdp
   rule-application routing
   exit
ruledef <routing_ruledef_sip>
   udp either-port = 5060
   rule-application routing
   exit
ruledef <routing_ruledef_smtp>
   tcp either-port = 25
   rule-application routing
   exit
```

```
ruledef <routing_ruledef_wap2.0>
   tcp either-port = 8080
   rule-application routing
   exit
ruledef <routing_ruledef_wsp-connection-less>
   udp either-port = 9200
   rule-application routing
   exit
ruledef <routing_ruledef_wsp-connection-oriented>
   udp either-port = 9201
   ip protocol = 51
   ip protocol = 50
   ip protocol = 47
   ip downlink = TRUE
   ip uplink = TRUE
   ip any-match = TRUE
   tcp any-match = TRUE
   udp dst-port = 5000
   rule-application routing
   end
```

Notes:

- If in a ruledef the rule-application is not specified, by default the system configures the ruledef as a charging ruledef.

Configuring the Charging Action

Use the following configuration example to configure the charging actions:

```
configure
        active-charging service <acs_service_name>
           charging-action <charging_action_name1>
```

```
               flow limit-for-bandwidth direction downlink peak-data-rate 4000
peak-burst-size 1024 violate-action discard committed-data-rate 3200 committed-
burst-size 512 exceed-action discard

               exit

            charging-action <charging_action_name2>

               content-id 1

               exit

            charging-action <charging_action_name3>

               flow action terminate-flow

               end
```

Configuring the Rulebase

Use the following configuration example to configure the rulebases for P2P. This configuration also enables the P2P analyzer to detect the P2P applications configured for the Active Charging Service. Note that the following example includes configuration to report voice and non-voice components for GTalk, MSN, Oscar, Skype, and Yahoo separately.

```
configure

         active-charging service <acs_service_name>

            rulebase <rulebase_name>

               action priority <priority> ruledef <charging_ruledef_actsync>
charging-action <charging_action_name>

               action priority <priority> ruledef <charging_ruledef_aimini>
charging-action <charging_action_name>

               action priority <priority> ruledef <charging_ruledef_applejuice>
charging-action <charging_action_name>

               action priority <priority> ruledef <charging_ruledef_ares>
charging-action <charging_action_name>

               action priority <priority> ruledef <charging_ruledef_battlefld>
charging-action <charging_action_name>

               action priority <priority> ruledef <charging_ruledef_bittorrent>
charging-action <charging_action_name>

               action priority <priority> ruledef <charging_ruledef_ddlink>
charging-action <charging_action_name>

               action priority <priority> ruledef
<charging_ruledef_directconnect> charging-action <charging_action_name>
```

```
                action priority <priority> ruledef <charging_ruledef_edonkey>
charging-action <charging_action_name>

                action priority <priority> ruledef <charging_ruledef_fasttrack>
charging-action <charging_action_name>

                action priority <priority> ruledef <charging_ruledef_feidian>
charging-action <charging_action_name>

                action priority <priority> ruledef <charging_ruledef_filetopia>
charging-action <charging_action_name>

                action priority <priority> ruledef <charging_ruledef_freenet>
charging-action <charging_action_name>

                action priority <priority> ruledef <charging_ruledef_fring>
charging-action <charging_action_name>

                action priority <priority> ruledef <charging_ruledef_gadugadu>
charging-action <charging_action_name>

                action priority <priority> ruledef <charging_ruledef_gnutella>
charging-action <charging_action_name>

                action priority <priority> ruledef <charging_ruledef_gtalk>
charging-action <charging_action_name>

                action priority <priority> ruledef <charging_ruledef_halflife2>
charging-action <charging_action_name>

                action priority <priority> ruledef <charging_ruledef_hamachivpn>
charging-action <charging_action_name>

                action priority <priority> ruledef <charging_ruledef_iax>
charging-action <charging_action_name>

                action priority <priority> ruledef <charging_ruledef_imesh>
charging-action <charging_action_name>

                action priority <priority> ruledef <charging_ruledef_iptv>
charging-action <charging_action_name>

                action priority <priority> ruledef <charging_ruledef_irc>
charging-action <charging_action_name>

                action priority <priority> ruledef <charging_ruledef_iskoot>
charging-action <charging_action_name>

                action priority <priority> ruledef <charging_ruledef_jabber>
charging-action <charging_action_name>

                action priority <priority> ruledef <charging_ruledef_manolito>
charging-action <charging_action_name>
```

```
                action priority <priority> ruledef <charging_ruledef_msn>
charging-action <charging_action_name>

                action priority <priority> ruledef <charging_ruledef_mute>
charging-action <charging_action_name>

                action priority <priority> ruledef <charging_ruledef_nimbuzz>
charging-action <charging_action_name>

                action priority <priority> ruledef <charging_ruledef_oovoo>
charging-action <charging_action_name>

                action priority <priority> ruledef <charging_ruledef_openft>
charging-action <charging_action_name>

                action priority <priority> ruledef <charging_ruledef_orb>
charging-action <charging_action_name>

                action priority <priority> ruledef <charging_ruledef_oscar>
charging-action <charging_action_name>

                action priority <priority> ruledef <charging_ruledef_paltalk>
charging-action <charging_action_name>

                action priority <priority> ruledef <charging_ruledef_pando>
charging-action <charging_action_name>

                action priority <priority> ruledef <charging_ruledef_pandora>
charging-action <charging_action_name>

                action priority <priority> ruledef <charging_ruledef_popo>
charging-action <charging_action_name>

                action priority <priority> ruledef <charging_ruledef_pplive>
charging-action <charging_action_name>

                action priority <priority> ruledef <charging_ruledef_ppstream>
charging-action <charging_action_name>

                action priority <priority> ruledef <charging_ruledef_qq>
charging-action <charging_action_name>

                action priority <priority> ruledef <charging_ruledef_qqgame>
charging-action <charging_action_name>

                action priority <priority> ruledef <charging_ruledef_qqlive>
charging-action <charging_action_name>

                action priority <priority> ruledef <charging_ruledef_quake>
charging-action <charging_action_name>

                action priority <priority> ruledef <charging_ruledef_rdp>
charging-action <charging_action_name>
```

```
                action priority <priority> ruledef <charging_ruledef_secondlife>
charging-action <charging_action_name>

                action priority <priority> ruledef <charging_ruledef_skinny>
charging-action <charging_action_name>

                action priority <priority> ruledef <charging_ruledef_skype>
charging-action <charging_action_name>

                action priority <priority> ruledef <charging_ruledef_slingbox>
charging-action <charging_action_name>

                action priority <priority> ruledef <charging_ruledef_sopcast>
charging-action <charging_action_name>

                action priority <priority> ruledef <charging_ruledef_soulseek>
charging-action <charging_action_name>

                action priority <priority> ruledef <charging_ruledef_steam>
charging-action <charging_action_name>

                action priority <priority> ruledef <charging_ruledef_tvants>
charging-action <charging_action_name>

                action priority <priority> ruledef <charging_ruledef_tvuplayer>
charging-action <charging_action_name>

                action priority <priority> ruledef <charging_ruledef_uusee>
charging-action <charging_action_name>

                action priority <priority> ruledef <charging_ruledef_vpnx>
charging-action <charging_action_name>

                action priority <priority> ruledef <charging_ruledef_vtun>
charging-action <charging_action_name>

                action priority <priority> ruledef <charging_ruledef_warcft3>
charging-action <charging_action_name>

                action priority <priority> ruledef <charging_ruledef_winmx>
charging-action <charging_action_name>

                action priority <priority> ruledef <charging_ruledef_winny>
charging-action <charging_action_name>

                action priority <priority> ruledef <charging_ruledef_wofwarcraft>
charging-action <charging_action_name>

                action priority <priority> ruledef <charging_ruledef_xbox>
charging-action <charging_action_name>

                action priority <priority> ruledef <charging_ruledef_yahoo>
charging-action <charging_action_name>
```

```
                action priority <priority> ruledef <charging_ruledef_zattoo>
charging-action <charging_action_name>

# Configuration to report voice and non-voice components for Oscar, GTalk, MSN,
Skype, and Yahoo separately:

                action priority <priority> ruledef <charging_ruledef_gtalk_voice>
charging-action <charging_action_name>

                action priority <priority> ruledef
<charging_ruledef_gtalk_non_voice> charging-action <charging_action_name>

                action priority <priority> ruledef
<charging_ruledef_msn_non_voice> charging-action <charging_action_name>

                action priority <priority> ruledef <charging_ruledef_voice>
charging-action <charging_action_name>

                action priority <priority> ruledef <charging_ruledef_oscar_voice>
charging-action <charging_action_name>

                action priority <priority> ruledef
<charging_ruledef_oscar_non_voice> charging-action <charging_action_name>

                action priority <priority> ruledef <charging_ruledef_skype_voice>
charging-action <charging_action_name>

                action priority <priority> ruledef
<charging_ruledef_skype_non_voice> charging-action <charging_action_name>

                action priority <priority> ruledef <charging_ruledef_yahoo_voice>
charging-action <charging_action_name>

                action priority <priority> ruledef
<charging_ruledef_yahoo_non_voice> charging-action <charging_action_name>

                action priority <priority> ruledef <charging_ruledef_msn_voice>
charging-action <charging_action_name>

                action priority <priority> ruledef <charging_ruledef_non_voice>
charging-action <charging_action_name>

                route priority <priority> ruledef <routing_ruledef_http> analyzer
http

                route priority <priority> ruledef <routing_ruledef_wap2.0>
analyzer http

                route priority <priority> ruledef <routing_ruledef_https>
analyzer secure-http

                route priority <priority> ruledef <routing_ruledef_imap> analyzer
imap
```

```
                route priority <priority> ruledef <routing_ruledef_pop3> analyzer
pop3

                route priority <priority> ruledef <routing_ruledef_smtp> analyzer
smtp

                route priority <priority> ruledef <routing_ruledef_dns-udp>
analyzer dns

                route priority <priority> ruledef <routing_ruledef_dns-tcp>
analyzer dns

                route priority <priority> ruledef <routing_ruledef_ftp-control>
analyzer ftp-control

                route priority <priority> ruledef <routing_ruledef_ftp-data>
analyzer ftp-data

                route priority <priority> ruledef <routing_ruledef_rtsp> analyzer
rtsp

                route priority <priority> ruledef <routing_ruledef_rtsp-8556>
analyzer rtsp

                route priority <priority> ruledef <routing_ruledef_sip> analyzer
sip

                route priority <priority> ruledef <routing_ruledef_wsp-
connection-less> analyzer wsp-connection-less

                route priority <priority> ruledef <routing_ruledef_wsp-
connection-oriented> analyzer wsp-connection-oriented

                route priority <priority> ruledef <routing_ruledef_sdp> analyzer
sdp

                route priority <priority> ruledef <routing_ruledef_mms-wapcl-ct>
analyzer mms

                route priority <priority> ruledef <routing_ruledef_mms_wapcl-url>
analyzer mms

                route priority <priority> ruledef <routing_ruledef_mms_http_ct>
analyzer mms

                route priority <priority> ruledef <routing_ruledef_mms_http_url>
analyzer mms

                rtp dynamic-flow-detection

                p2p dynamic-flow-detection

                end
```

Notes:

- For information about the list of protocols that support prepaid and postpaid charging, refer to the *Peer-to-Peer Overview* chapter of this guide.

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 post-processing. Event Detail Records (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. This configuration aids in capturing the detected P2P protocol data in the EDR.

Use the following example to set the EDR configuration:

```
configure
        active-charging service <ecs_service>
           edr-format <edr_flow_format>
              rule-variable traffic type priority <priority>
              rule-variable voip-duration priority <priority>
              attribute sn-start-time format seconds priority <priority>
              attribute sn-end-time format seconds priority <priority>
              attribute radius-calling-station-id priority <priority>
              rule-variable ip server-ip-address priority <priority>
              attribute sn-server-port priority <priority>
              attribute sn-app-protocol priority <priority>
              attribute sn-parent-protocol priority <priority>
              rule-variable ip protocol priority <priority>
              rule-variable p2p protocol priority <priority>
              attribute sn-volume-amt ip bytes uplink priority
<priority>
              attribute sn-volume-amt ip bytes downlink priority
<priority>
              attribute sn-volume-amt ip pkts uplink priority <priority>
              attribute sn-volume-amt ip pkts downlink priority
<priority>
```

```
                    rule-variable bearer 3gpp charging-id priority <priority>
                    rule-variable bearer 3gpp imei priority <priority>
                    rule-variable bearer 3gpp rat-type priority <priority>
                    rule-variable bearer 3gpp user-location-information
      priority <priority>
                    end
```

Notes:

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

Enable DSCP Marking

Use the following configuration example to enable DSCP marking in the configuration:

```
configure
        context ggsn
           interface <interface>
              ip address <address/mask>
              ip arp timeout <timeout>
              exit
           subscriber default
              ip context-name <context_name>
              exit
           apn <apn_name>
              selection-mode sent-by-ms
              accounting-mode none
              ip access-group <access_group_name> in
              ip access-group <access_group_name> out
              ip source-violation ignore
              ip qos-dscp conversational pt streaming pt interactive 1 pt
interactive 2 pt interactive 3 pt background pt
```

```
            ip qos-dscp interactive 1 allocation-retention-priority 1 pt
interactive 1 allocation-retention-priority 2 pt interactive 1 allocation-
retention-priority 3 pt

            ip qos-dscp interactive 2 allocation-retention-priority 1 pt
interactive 2 allocation-retention-priority 2 pt interactive 2 allocation-
retention-priority 3 pt

            ip qos-dscp interactive 3 allocation-retention-priority 1 pt
interactive 3 allocation-retention-priority 2 pt interactive 3 allocation-
retention-priority 3 pt

            ip context-name <context_name>

            ip address pool name <pool_name>

            active-charging rulebase <rulebase_name>

            exit

         aaa group default

            exit

         gtpp group default

            exit

         ggsn-service GGSN

            retransmission-timeout <retransmission_timeout>

            max-retransmission <max_retransmission>

            plmn unlisted-sgsn home

            bind address <ip_address>

            exit

      context <context_name>

         ip access-list <access_list_name>

            redirect css service <acs_service> ip any any

            exit

         ip pool <pool_name> <ip_address/mask> static

         interface <interface>

            ip address <ip_address/mask>

            ip arp timeout <timeout>
```

```
            exit
         subscriber default
            exit
         radius group default
            exit
         gtpp group default
            exit
         ip route <ip_address/mask> <interface>
         exit
port ethernet <interface>
      no shutdown
      bind interface <interface> ggsn
      exit
port ethernet <interface>
      no shutdown
      bind interface <interface> <context_name>
      end
```

Notes:

- *<acs_service>* is the name of the ACS service; no CSS service needs to be configured.

Configuring P2P Dynamic Signature Updates

This section describes how to enable and configure the P2P Dynamic Signature Updates feature.

Enabling/Disabling P2P Dynamic Signature Updates

To enable the P2P Dynamic Signature Updates feature, use the following configuration example:

```
configure
      active-charging service <acs_service_name>
         default p2p-dynamic-rules file
         end
```

Notes:

- On enabling the P2P Dynamic Signature Updates feature, if a P2P signature file is available at the default location, the system loads it. Default location for the signature file is "*/usr/lib/p2p-rules.xml*".

Loading/Enabling Signatures

To enable the P2P Dynamic Signature Updates feature, and load a specific signature file (from other than the default location) to the memory, use the following configuration example:

```
configure

        active-charging service <acs_service_name>

           p2p-dynamic-rules { file <location>  | protocol [ all | bittorrent |
directconnect | edonkey | gnutella | skype | yahoo + ] }

           end
```

Notes:

- *<location>* must be one of the following:

  [file:]{/flash | /pcmcia1 | /hd-raid}[/*<directory>***]/***<filename>*

- The **protocol** keyword and options can be used to selectively enable signatures for specific protocol(s).
- This release supports dynamic signature updates only for the following protocols: BitTorrent, DirectConnect, eDonkey, Gnutella, Skype, and Yahoo.

Unloading/Disabling Signatures

To disable the P2P Dynamic Signature Updates feature, and delete any signatures in the memory, use the following configuration example:

```
configure

        active-charging service <acs_service_name>

           no p2p-dynamic-rules { file | protocol [ all | bittorrent |
directconnect | edonkey | gnutella | skype | yahoo + ] }

           end
```

Notes:

- The **no p2p-dynamic-rules file** command disables the P2P Dynamic Signature Updates feature, also any/specified signature(s) already loaded in the memory is unloaded. If there are any active sessions using the file, it changes the file status to inactive. And, when the sessions are cleared, the file is removed from the memory.
- The **no p2p-dynamic-rules protocol [all | bittorrent | directconnect | edonkey | gnutella | skype | yahoo +]** command disables the execution of signatures loaded in the memory for a specific protocol or all protocols.

Saving the Configuration

Refer to the *Verifying and Saving Your Configuration* chapter of this guide to save changes made to the system configuration for P2P Detection.

Verifying the Configuration

This section explains how to review the configurations after saving them in a *.cfg* file as described in *Verifying and Saving Your Configuration* chapter and also to retrieve errors and warnings within an active configuration for a service.

Viewing System Configuration

The following configuration example displays the active configuration for a service:

```
configure

        context <context_name>

        end

show configuration [ card <card_num> | context <name> [ radius group [ all |
name <group> ] ] | port <slot/port> | srp ] [ showsecrets ] [ url <url> ] [
verbose ] [ | { grep <grep_options> | more } ]
```

Viewing Service Configuration Errors

The following configuration example displays the errors in configuration for a service:

```
configure

        context <context_name>

        end

show configuration errors section active-charging [ verbose ] [ | { grep
<grep_options> | more } ]
```

Gathering P2P Statistics

In the following table, the first column lists what statistics to gather, the second column lists an action to perform, and the third column describes what information is displayed or what information to look for in the resulting output.

Table 1. Gathering Statistics

Statistics Wanted	Action to Perform	Information to Look For
Analyzer statistics	At the Exec Mode prompt, enter the following command: **show active-charging analyzer statistics name p2p verbose**	The output of this command displays the analyzer statistics if a P2P analyzer is used. Since the analyzer statistics are not bound to any service, the traffic information per gateway can be obtained.
Ruledef statistics	At the Exec Mode prompt, enter the following command: **show active-charging ruledef statistics name** *<name>*	The output of this command displays the Ruledef statistics including the packet count, byte count and hits.
P2P flow statistics	At the Exec Mode prompt, enter the following command: **show active-charging flows type p2p traffic-type voice** **show active-charging flows type p2p traffic-type non-voice**	The output of this command displays the number of P2P voice and non-voice flows.
Charging Action information	At the Exec Mode prompt, enter the following command: **show active-charging charging-action statistics**	The output of this command displays the charging action information and corresponding statistics configured in the active charging service.
Transmit and Receive data	At the Exec Mode prompt, enter the following command: **show active-charging sessions tx-data** *<operator> <bytes>* **show active-charging sessions rx-data** *<operator> <bytes>*	The output of the command displays the information for sessions that have received or transmitted data which matches the criteria.
Sessions using specific protocol	At the Exec Mode prompt, enter the following command: **show active-charging sessions type P2P application** *<protocol>*	The output of this command displays information for the sessions using the specified protocol.
Total and current P2P and P2P voice flows	At the Exec Mode prompt, enter the following command: **show active-charging subsystem all**	The output of this command displays total and current P2P flow and P2P voice flow statistics, and total number of subscribers.
Dynamic signature files information	At the Exec Mode prompt, enter the following command: **show active-charging p2p-dynamic-rules verbose**	The output of this command displays P2P dynamic signature file information.
Voice Statistics	At the Exec Mode prompt, enter the following command: **show active-charging analyzer statistics name p2p application [gtalk | msn | oscar | skype | yahoo]**	The output of this command displays the voice and non-voice analyzer statistics for voice supported protocols (MSN, Yahoo, GTalk, Skype, Oscar).

The P2P analyzer tracks all P2P protocols for both uplink and downlink packets and bytes statistics. For additional statistics, refer to the *Gathering P2P Statistics* section in the *P2P Service Configuration* chapter of the *Peer-to-Peer Detection Administration Guide*.

Supported Bulk Statistics

For information on P2P bulk statistics and bulk statistics configuration and collection, refer to the *Bulk Statistics Configuration Mode Commands* chapter of the *Command Line Interface Reference*, and the *Statistics and Counters Reference*.

P2P Reports

The P2P reports provide details of the bandwidth consumption of P2P traffic over time. These reports are used to analyze network performance, identify the customer trends, network usage patterns, and network categorization.

IMPORTANT: In StarOS 9.0 and earlier releases, the P2P reporting functionality was available in the Web Element Manager software. For more information, refer to the *WEM Online Help* documentation.

IMPORTANT: In StarOS 10.0 and later releases, the P2P reporting functionality is supported in inPilot. For more information, refer to the *inPilot Online Help* documentation.

The following bandwidth usage reports are supported:

- Cumulative analyzer count - representing the total bandwidth consumed by the P2P traffic in bits/sec. Daily, monthly or yearly reports are supported.
- Total bandwidth consumed P2P traffic against other protocols like HTTP, RTSP, etc. Daily or monthly reports are supported.
- Per protocol type - total bandwidth consumed by the individual P2P protocol traffic in packets/sec or bytes/sec plotted against time range or date range. Daily reports are supported. The graph uses separate colors to differentiate among the multiple protocol types.
- The number of active users per application for specified date/time range. Daily reports are supported.
- Analysis of the percentage of total bandwidth consumed by P2P traffic from the total subscriber traffic. Weekly reports are supported.

IMPORTANT: For additional information about viewing reports, refer to the *Web Element Manager Online Help System*.

Chapter 3
Verifying and Saving Your Configuration

This chapter describes how to save the system configuration.

Verifying the Configuration

You can use a number of command to verify the configuration of your feature, service, or system. Many are hierarchical in their implementation and some are specific to portions of or specific lines in the configuration file.

Feature Configuration

In many configurations, specific features are set and need to be verified. Examples include APN and IP address pool configuration. Using these examples, enter the following commands to verify proper feature configuration:

`show apn all`

The output displays the complete configuration for the APN. In this example, an APN called apn1 is configured.

```
access point name (APN): apn1

authentication context: test

pdp type: ipv4

Selection Mode: subscribed

ip source violation: Checked drop limit: 10

accounting mode: gtpp No early PDUs: Disabled

max-primary-pdp-contexts: 1000000 total-pdp-contexts: 1000000

primary contexts: not available total contexts: not available

local ip: 0.0.0.0

primary dns: 0.0.0.0 secondary dns: 0.0.0.0

ppp keep alive period : 0 ppp mtu : 1500

absolute timeout : 0 idle timeout : 0

long duration timeout: 0 long duration action: Detection

ip header compression: vj

data compression: stac mppc deflate compression mode: normal

min compression size: 128

ip output access-group: ip input access-group:

ppp authentication:

allow noauthentication: Enabled imsi

 authentication:Disabled
```

Enter the following command to display the IP address pool configuration:

show ip pool

The output from this command should look similar to the sample shown below. In this example, all IP pools were configured in the *isp1* context.

```
context : isp1:

+-----Type: (P) - Public (R) - Private

 | (S) - Static (E) - Resource

 |

 |+----State: (G) - Good (D) - Pending Delete (R)-Resizing

 ||

 ||++--Priority: 0..10 (Highest (0) .. Lowest (10))

 ||||

 ||||+-Busyout: (B) - Busyout configured

 |||| |||||| vvvvv Pool Name Start Address Mask/End Address Used Avail

 ----- --------- --------------- --------------- -------- --------

PG00 ipsec 12.12.12.0 255.255.255.0 0 254 PG00

pool1 10.10.0.0 255.255.0.0 0 65534 SG00

vpnpool 192.168.1.250 192.168.1.254 0 5 Total Pool Count: 5
```

IMPORTANT: Many features can be configured on the system. There are show commands specifically for these features. Refer to the *Command Line Interface Reference* for more information.

Service Configuration

Verify that your service was created and configured properly by entering the following command:

show *<service_type> <service_name>*

The output is a concise listing of the service parameter settings similar to the sample displayed below. In this example, a P-GW service called pgw is configured.

```
Service name : pgw1

Service-Id : 1

Context : test1
```

```
Status : STARTED

Restart Counter : 8

EGTP Service : egtp1

LMA Service : Not defined

Session-Delete-Delay Timer : Enabled

Session-Delete-Delay timeout : 10000(msecs)

PLMN ID List : MCC: 100, MNC: 99

Newcall Policy : None
```

Context Configuration

Verify that your context was created and configured properly by entering the following command:

show context name *<name>*

The output shows the active context. Its ID is similar to the sample displayed below. In this example, a context named *test1* is configured.

```
Context Name     ContextID     State

------------     ---------     -----

test1            2          Active
```

System Configuration

Verify that your entire configuration file was created and configured properly by entering the following command:

show configuration

This command displays the entire configuration including the context and service configurations defined above.

Finding Configuration Errors

Identify errors in your configuration file by entering the following command:

show configuration errors

This command displays errors it finds within the configuration. For example, if you have created a service named "service1", but entered it as "srv1" in another part of the configuration, the system displays this error.

You must refine this command to specify particular sections of the configuration. Add the **section** keyword and choose a section from the help menu:

show configuration errors section ggsn-service

or

show configuration errors section aaa-config

If the configuration contains no errors, an output similar to the following is displayed:

```
##################################################################################

Displaying Global
AAA-configuration errors
##################################################################################

Total 0 error(s) in this section !
```

Saving the Configuration

Save system configuration information to a file locally or to a remote node on the network. You can use this configuration file on any other systems that require the same configuration.

Files saved locally can be stored in the SPC's/SMC's CompactFlash or on an installed PCMCIA memory card on the SPC/SMC. Files that are saved to a remote network node can be transmitted using either FTP, or TFTP.

Saving the Configuration on the Chassis

These instructions assume that you are at the root prompt for the Exec mode:

```
[local]host_name#
```

To save your current configuration, enter the following command:

```
save configuration url [-redundant] [-noconfirm] [showsecrets] [verbose]
```

Keyword/Variable	Description
url	Specifies the path and name to which the configuration file is to be stored. *url* may refer to a local or a remote file. *url* must be entered using one of the following formats: • **{ /flash | /pcmcia1 | /pcmcia2 }** *[/dir] /file_name* • **file:/{ /flash | /pcmcia1 | /pcmcia2 }** *[/dir] /file_name* • **tftp://**{ *ipaddress*|*host_name*[*:port#*]} [*/directory*] */file_name* • **ftp://**[*username*[*:pwd*] *@*] { *ipaddress*|*host_name*} [*:port#*] [*/directory*] */file_name* • **sftp://**[*username*[:*pwd*] *@*] { *ipaddress*|*host_name*} [*:port#*] [*/directory*] */file_name* **/flash** corresponds to the CompactFlash on the SPC/SMC. **/pcmcia1** corresponds to PCMCIA slot 1. **/pcmcia2** corresponds to PCMCIA slot 2. *ipaddress* is the IP address of the network server. *host_name* is the network server's *hostname*. *port#* is the network server's logical port number. Defaults are: • tftp: 69 - data • ftp: 20 - data, 21 - control • sftp: 115 - data Note: *host_name* can only be used if the **networkconfig** parameter is configured for DHCP and the DHCP server returns a valid nameserv er.dx *username* is the username required to gain access to the server if necessary. *password* is the password for the specified username if required. */directory* specifies the directory where the file is located if one exists. */file_name* specifies the name of the configuration file to be saved. Note: Configuration files should be named with a .cfg extension.
-redundant	Optional: This keyword directs the system to save the CLI configuration file to the local device, defined by the url variable, and then automatically copy that same file to the like device on the Standby SPC/SMC, if available. Note: This keyword will only work for like local devices that are located on both the active and standby SPCs/SMCs. For example, if you save the file to the /pcmcia1 device on the active SPC/SMC, that same type of device (a PC-Card in Slot 1 of the standby SPC/SMC) must be available. Otherwise, a failure message is displayed. Note: If saving the file to an external network (non-local) device, the system disregards this keyword.

Keyword/Variable	Description
-noconfirm	Optional: Indicates that no confirmation is to be given prior to saving the configuration information to the specified filename (if one was specified) or to the currently active configuration file (if none was specified).
showsecrets	Optional: This keyword causes the CLI configuration file to be saved with all passwords in plain text, rather than their default encrypted format.
verbose	Optional: Specifies that every parameter that is being saved to the new configuration file should be displayed.

IMPORTANT: The **-redundant** keyword is only applicable when saving a configuration file to local devices .This command does not synchronize the local file system. If you have added, modified, or deleted other files or directories to or from a local device for the active SPC/SMC, then you must synchronize the local file system on both SPCs/SMCs.

To save a configuration file called system.cfg to a directory that was previously created called cfgfiles on the SPC's/SMC's CompactFlash, enter the following command:

```
save configuration /flash/cfgfiles/system.cfg
```

To save a configuration file called simple_ip.cfg to a directory called host_name_configs using an FTP server with an IP address of 192.168.34.156 on which you have an account with a username of administrator and a password of secure, use the following command:

```
save configuration
ftp://administrator:secure@192.168.34.156/host_name_configs/
simple_ip.cfg
```

To save a configuration file called init_config.cfg to the root directory of a TFTP server with a hostname of config_server, enter the following command:

save configuration *tftp://config_server/init_config.cfg*

Chapter 4
Sample Peer-to-Peer Configuration in an ECS Service

This appendix contains a sample Peer-to-Peer (P2P) configuration within an ECS service that includes the examples from the procedures in *Peer-to-Peer Detection Configuration* chapter.

```
configure
   license key "\
VER=1|C1M=SanDiskSDCFJ-4096|C1S=116919K2106K0235|DOI=1217844147|DOE=12\
33741747|ISS=1|NUM=26914|CMT=bngnc18,_chassis1,_|LSP=100000|LS0|LSF=10000
0|SIG=MCwCFFABNedEgGb8fAw8u0lvwxbWbJEBAhQvpG9YREYRFDDElzNUBuZv3kbHQw"
   system hostname host_name
   autoconfirm
   crash enable encrypted url 057285fc2112177777b5e7a716356c3e332f12f89
   card 1
      mode active psc
      exit
   card 4
      mode active psc
      exit
   card 16
      mode active psc
      exit
   require active-charging
   context local
      interface spio1
         ip address 1.2.3.4 255.255.255.0
         exit
```

```
      server ftpd
         exit
      ssh key
0d94d7812a224fd97a58d9c6dab47bd7b318e705d1ee91d45254ef1286be8ef5cc271cf3d
05656652014d69a568d099664ed2354369ce6481772a2dbf0f37ad20dc1e2b765d8c9f041
759c0e1e8a9e53e3975b1724329d1a2012bf0221cc132014a1224cdfc45ca7 len 461
      ssh key
75f41778bab0a173ee6e4e79c102638966c38eb5490fe46be064007e6951792a6abaf2733
c4f4972318eb3b77f85d8925d4aae335dedfa0619f03cdfb3f35fef82cfa97eb1b2517654
aad83afc2c7c5c08d76e2e4e9d8edaddd280f7963c227ff8f122ecefb9d8e0 len 457
type v2-dsa
      server sshd
         subsystem sftp
         exit
      server telnetd
         exit
      subscriber default
         exit
      administrator admin encrypted password abc123def456ghi ftp
      aaa group default
         exit
      gtpp group default
         exit
      ip route 0.0.0.0 0.0.0.0 1.2.3.4 spio1
         exit
     exit
   port ethernet 24/1
      no shutdown
      bind interface spio1 local
      exit
   ntp
      enable
```

```
    server 10.6.1.1
    exit
 snmp engine-id local 87e55bf69c4c479d
active-charging service service_1
    p2p-detection protocol all
    p2p-dynamic-rules file /net/user/xmls/p2p-all-0.2.xml
     ruledef ch_actsync
       p2p protocol = actsync
       exit
     ruledef ch_aimini
       p2p protocol = aimini
       exit
     ruledef ch_applejuice
       p2p protocol = applejuice
       exit
    ruledef ch_ares
       p2p protocol = ares
       exit
     ruledef ch_battlefld
       p2p protocol = battlefld
       exit
    ruledef ch_bittorrent
       p2p protocol = bittorrent
       exit
    ruledef ch_ddlink
       p2p protocol = ddlink
       exit
    ruledef ch_directconnect
       p2p protocol = directconnect
```

```
   exit
ruledef ch_edonkey
   p2p protocol = edonkey
   exit
ruledef ch_fasttrack
   p2p protocol = fasttrack
   exit
ruledef ch_feidian
   p2p protocol = feidian
   exit
ruledef ch_filetopia
   p2p protocol = filetopia
   exit
 ruledef ch_freenet
   p2p protocol = freenet
   exit
 ruledef ch_fring
   p2p protocol = fring
   exit
ruledef ch_gadugadu
   p2p protocol = gadugadu
   exit
ruledef ch_gnutella
   p2p protocol = gnutella
   exit
ruledef ch_gtalk
   p2p protocol = gtalk
   exit
ruledef ch_halflife2
```

```
   p2p protocol = halflife2
   exit
ruledef ch_hamachivpn
   p2p protocol = hamachivpn
   exit
 ruledef ch_iax
   p2p protocol = iax
   exit
ruledef ch_imesh
   p2p protocol = imesh
   exit
 ruledef ch_iptv
   p2p protocol = iptv
   exit
ruledef ch_irc
   p2p protocol = irc
   exit
 ruledef ch_iskoot
   p2p protocol = iskoot
   exit
ruledef ch_jabber
   p2p protocol = jabber
   exit
ruledef ch_manolito
   p2p protocol = manolito
   exit
ruledef ch_msn
   p2p protocol = msn
   exit
```

```
ruledef ch_mute
   p2p protocol = mute
   exit
 ruledef ch_nimbuzz
   p2p protocol = nimbuzz
   exit
ruledef ch_oovoo
   p2p protocol = oovoo
   exit
 ruledef ch_openft
   p2p protocol = openft
   exit
ruledef ch_orb
   p2p protocol = orb
   exit
ruledef ch_oscar
   p2p protocol = oscar
   exit
 ruledef ch_paltalk
   p2p protocol = paltalk
   exit
ruledef ch_pando
   p2p protocol = pando
   exit
 ruledef ch_pandora
   p2p protocol = pandora
   exit
ruledef ch_popo
   p2p protocol = popo
```

```
   exit
ruledef ch_pplive
   p2p protocol = pplive
   exit
ruledef ch_ppstream
   p2p protocol = ppstream
   exit
ruledef ch_qq
   p2p protocol = qq
   exit
 ruledef ch_qqgame
   p2p protocol = qqgame
   exit
ruledef ch_qqlive
   p2p protocol = qqlive
   exit
 ruledef ch_quake
   p2p protocol = quake
   exit
 ruledef ch_rdp
   p2p protocol = rdp
   exit
 ruledef ch_secondlife
   p2p protocol = secondlife
   exit
ruledef ch_skinny
   p2p protocol = skinny
   exit
ruledef ch_skype
```

```
   p2p protocol = skype
   exit
ruledef ch_slingbox
   p2p protocol = slingbox
   exit
ruledef ch_sopcast
   p2p protocol = sopcast
   exit
ruledef ch_soulseek
   p2p protocol = soulseek
   exit
ruledef ch_steam
   p2p protocol = steam
   exit
 ruledef ch_tvants
   p2p protocol = tvants
   exit
ruledef ch_tvuplayer
   p2p protocol = tvuplayer
   exit
ruledef ch_uusee
   p2p protocol = uusee
   exit
ruledef ch_vpnx
   p2p protocol = vpnx
   exit
ruledef ch_vtun
   p2p protocol = vtun
   exit
```

```
 ruledef ch_warcft3
    p2p protocol = warcft3
    exit
ruledef ch_winmx
    p2p protocol = winmx
    exit
ruledef ch_winny
    p2p protocol = winny
    exit
ruledef ch_wofwarcraft
    p2p protocol = wofwarcraft
    exit
ruledef ch_xbox
    p2p protocol = xbox
    exit
ruledef ch_yahoo
    p2p protocol = yahoo
    exit
ruledef ch_zattoo
    p2p protocol = zattoo
    exit
ruledef ch_voice_gtalk
    p2p protocol = gtalk
    p2p traffic-type = voice
    rule-application charging
    exit
ruledef ch_voice_msn
    p2p protocol = msn
    p2p traffic-type = voice
```

```
    rule-application charging
    exit
ruledef ch_voice_oscar
    p2p protocol = oscar
    p2p traffic-type = voice
    rule-application charging
    exit
ruledef ch_voice_skype
    p2p protocol = skype
    p2p traffic-type = voice
    rule-application charging
    exit
ruledef ch_voice_yahoo
    p2p protocol = yahoo
    p2p traffic-type = voice
    rule-application charging
    exit
ruledef ch_voice
    p2p traffic-type = voice
    rule-application charging
    exit
ruledef ch_non_voice_gtalk
    p2p protocol = gtalk
    p2p traffic-type != voice
    rule-application charging
    exit
ruledef ch_non_voice_msn
    p2p protocol = msn
    p2p traffic-type != voice
```

```
   rule-application charging
   exit
ruledef ch_non_voice_oscar
   p2p protocol = oscar
   p2p traffic-type != voice
   rule-application charging
   exit
ruledef ch_non_voice_skype
   p2p protocol = skype
   p2p traffic-type != voice
   rule-application charging
   exit
ruledef ch_non_voice_yahoo
   p2p protocol = yahoo
   p2p traffic-type != voice
   rule-application charging
   exit
ruledef ch_non_voice
   p2p traffic-type != voice
   rule-application charging
   exit
ruledef rt_dns-tcp
   tcp either-port = 53
   rule-application routing
   exit
ruledef rt_dns-udp
   udp either-port = 53
   rule-application routing
   exit
```

```
ruledef rt_ftp-control
   tcp either-port = 21
   rule-application routing
   exit
ruledef rt_ftp-data
   tcp either-port = 20
   rule-application routing
   exit
ruledef rt_http
   tcp either-port = 80
   rule-application routing
   exit
ruledef rt_https
   tcp either-port = 443
   rule-application routing
   exit
ruledef rt_imap
   tcp either-port = 143
   rule-application routing
   exit
ruledef rt_mms-wapcl-ct
   wsp content type = application/vnd.wap.mms-message
   rule-application routing
   exit
ruledef rt_mms_http_ct
   http content type = application/vnd.wap.mms-message
   rule-application routing
   exit
ruledef rt_mms_http_url
```

```
   http url ends-with .mms
   rule-application routing
   exit
ruledef rt_mms_wapcl-url
   wsp url ends-with .mms
   rule-application routing
   exit
ruledef rt_pop3
   tcp either-port = 110
   rule-application routing
   exit
ruledef rt_rtsp
   tcp either-port = 554
   rule-application routing
   exit
ruledef rt_rtsp-8556
   tcp either-port = 8556
   rule-application routing
   exit
ruledef rt_sdp
   sip content type = application/sdp
   rule-application routing
   exit
ruledef rt_sip
   udp either-port = 5060
   rule-application routing
   exit
ruledef rt_smtp
   tcp either-port = 25
```

```
      rule-application routing
      exit
   ruledef rt_wap2.0
      tcp either-port = 8080
      rule-application routing
      exit
   ruledef rt_wsp-connection-less
      udp either-port = 9200
      rule-application routing
      exit
   ruledef rt_wsp-connection-oriented
      udp either-port = 9201
      ip protocol = 51
      ip protocol = 50
      ip protocol = 47
      ip downlink = TRUE
      ip uplink = TRUE
      ip any-match = TRUE
      tcp any-match = TRUE
      udp dst-port = 5000
      rule-application routing
      exit
   charging-action ca_BWC
      flow limit-for-bandwidth direction downlink peak-data-rate 4000
peak-burst-size 1024 violate-action discard committed-data-rate 3200
committed-burst-size 512 exceed-action discard
      exit
   charging-action ca_nothing
      content-id 1
      exit
```

```
      charging-action ca_terminate
         flow action terminate-flow
         exit
      rulebase base_1
         action priority 500 ruledef ch_actsync charging-action
ca_nothing
         action priority 501 ruledef ch_aimini charging-action ca_nothing
         action priority 502 ruledef ch_applejuice charging-action
ca_nothing
         action priority 503 ruledef ch_ares charging-action ca_nothing
         action priority 504 ruledef ch_battlefld charging-action
ca_nothing
         action priority 505 ruledef ch_bittorrent charging-action
ca_nothing
         action priority 506 ruledef ch_ddlink charging-action ca_nothing
         action priority 507 ruledef ch_directconnect charging-action
ca_nothing
         action priority 508 ruledef ch_edonkey charging-action
ca_nothing
         action priority 509 ruledef ch_fasttrack charging-action
ca_nothing
         action priority 510 ruledef ch_feidian charging-action
ca_nothing
         action priority 511 ruledef ch_filetopia charging-action
ca_nothing
         action priority 512 ruledef ch_freenet charging-action
ca_nothing
         action priority 513 ruledef ch_fring charging-action ca_nothing
         action priority 514 ruledef ch_gadugadu charging-action
ca_nothing
         action priority 515 ruledef ch_gnutella charging-action
ca_nothing
         action priority 516 ruledef ch_gtalk charging-action ca_nothing
         action priority 517 ruledef ch_halflife2 charging-action
ca_nothing
```

```
        action priority 518 ruledef ch_hamachivpn charging-action
ca_nothing
        action priority 519 ruledef ch_iax charging-action ca_nothing
        action priority 520 ruledef ch_imesh charging-action ca_nothing
        action priority 521 ruledef ch_iptv charging-action ca_nothing
        action priority 522 ruledef ch_irc charging-action ca_nothing
        action priority 523 ruledef ch_iskoot charging-action ca_nothing
        action priority 524 ruledef ch_jabber charging-action ca_nothing
        action priority 525 ruledef ch_manolito charging-action
ca_nothing
        action priority 526 ruledef ch_msn charging-action ca_nothing
        action priority 527 ruledef ch_mute charging-action ca_nothing
        action priority 528 ruledef ch_nimbuzz charging-action
ca_nothing
        action priority 529 ruledef ch_oovoo charging-action ca_nothing
        action priority 530 ruledef ch_openft charging-action ca_nothing
        action priority 531 ruledef ch_orb charging-action ca_nothing
        action priority 532 ruledef ch_oscar charging-action ca_nothing
        action priority 533 ruledef ch_paltalk charging-action
ca_nothing
        action priority 534 ruledef ch_pando charging-action ca_nothing
        action priority 535 ruledef ch_pandora charging-action
ca_nothing
        action priority 536 ruledef ch_pplive charging-action ca_nothing
        action priority 537 ruledef ch_ppstream charging-action
ca_nothing
        action priority 538 ruledef ch_qq charging-action ca_nothing
        action priority 539 ruledef ch_qqgame charging-action ca_nothing
        action priority 540 ruledef ch_qqlive charging-action ca_nothing
        action priority 541 ruledef ch_quake charging-action ca_nothing
        action priority 542 ruledef ch_rdp charging-action ca_nothing
```

```
         action priority 543 ruledef ch_secondlife charging-action
ca_nothing

         action priority 544 ruledef ch_skinny charging-action ca_nothing

         action priority 545 ruledef ch_skype charging-action ca_nothing

         action priority 546 ruledef ch_slingbox charging-action
ca_nothing

         action priority 547 ruledef ch_sopcast charging-action
ca_nothing

         action priority 548 ruledef ch_soulseek charging-action
ca_nothing

         action priority 549 ruledef ch_steam charging-action ca_nothing

         action priority 550 ruledef ch_tvants charging-action ca_nothing

         action priority 551 ruledef ch_tvuplayer charging-action
ca_nothing

         action priority 552 ruledef ch_uusee charging-action ca_nothing

         action priority 553 ruledef ch_vpnx charging-action ca_nothing

         action priority 554 ruledef ch_vtun charging-action ca_nothing

         action priority 555 ruledef ch_warcft3 charging-action
ca_nothing

         action priority 556 ruledef ch_winmx charging-action ca_nothing

         action priority 557 ruledef ch_winny charging-action ca_nothing

         action priority 558 ruledef ch_wofwarcraft charging-action
ca_nothing

         action priority 559 ruledef ch_xbox charging-action ca_nothing

         action priority 560 ruledef ch_yahoo charging-action ca_nothing

         action priority 561 ruledef ch_zattoo charging-action ca_nothing

         action priority 562 ruledef ch_voice_oscar charging-action
ca_nothing

         action priority 563 ruledef ch_voice_gtalk charging-action
ca_nothing

         action priority 564 ruledef ch_voice_msn charging-action
ca_nothing

         action priority 565 ruledef ch_voice_skype charging-action
ca_nothing
```

```
        action priority 566 ruledef ch_voice_yahoo charging-action
ca_nothing

        action priority 567 ruledef ch_non_voice_oscar charging-action
ca_nothing

        action priority 568 ruledef ch_non_voice_gtalk charging-action
ca_nothing

        action priority 569 ruledef ch_non_voice_msn charging-action
ca_nothing

        action priority 570 ruledef ch_non_voice_skype charging-action
ca_nothing

        action priority 571 ruledef ch_non_voice_yahoo charging-action
ca_nothing

        action priority 572 ruledef ch_voice charging-action ca_nothing

        action priority 573 ruledef ch_non_voice charging-action
ca_nothing

        route priority 10 ruledef rt_http analyzer http

        route priority 12 ruledef rt_wap2.0 analyzer http

        route priority 15 ruledef rt_https analyzer secure-http

        route priority 20 ruledef rt_imap analyzer imap

        route priority 25 ruledef rt_pop3 analyzer pop3

        route priority 30 ruledef rt_smtp analyzer smtp

        route priority 35 ruledef rt_dns-udp analyzer dns

        route priority 36 ruledef rt_dns-tcp analyzer dns

        route priority 40 ruledef rt_ftp-control analyzer ftp-control

        route priority 41 ruledef rt_ftp-data analyzer ftp-data

        route priority 45 ruledef rt_rtsp analyzer rtsp

        route priority 46 ruledef rt_rtsp-8556 analyzer rtsp

        route priority 50 ruledef rt_sip analyzer sip

        route priority 55 ruledef rt_wsp-connection-less analyzer wsp-
connection-less

        route priority 56 ruledef rt_wsp-connection-oriented analyzer
wsp-connection-oriented

        route priority 60 ruledef rt_sdp analyzer sdp
```

```
      route priority 65 ruledef rt_mms-wapcl-ct analyzer mms
      route priority 66 ruledef rt_mms_wapcl-url analyzer mms
      route priority 67 ruledef rt_mms_http_ct analyzer mms
      route priority 68 ruledef rt_mms_http_url analyzer mms
      rtp dynamic-flow-detection
      p2p dynamic-flow-detection
      exit
   rulebase default
      exit
   exit
context isp
   ip access-list list_1
      redirect css service service_1   ip any any
      exit
   ip pool pool1 9.8.7.6 255.255.255.0 static
   interface inet
      ip address 8.7.6.5 255.255.255.0
      exit
   subscriber default
      exit
   aaa group default
      exit
   gtpp group default
      exit
   ip route 0.0.0.0 0.0.0.0 7.6.5.4 inet
      exit
   exit
context ggsn
   interface ggsn-ingress
```

```
      ip address 6.5.4.3 255.255.255.0
      exit
   subscriber default
      exit
   apn radius.com
      selection-mode sent-by-ms
      accounting-mode none
      ip access-group list_1 in
      ip access-group list_1 out
      ip source-violation ignore
      ip context-name isp
      active-charging rulebase base_1
      exit
   aaa group default
      exit
   gtpp group default
      exit
   ggsn-service ggsn
      retransmission-timeout 1
      max-retransmission 1
      gtpu udp-checksum insert
      plmn unlisted-sgsn home
      bind address 5.4.3.2
      exit
   exit
port ethernet 17/1
   medium speed 1000 duplex full
   no shutdown
   bind interface ggsn-ingress ggsn
```

```
   exit
port ethernet 20/1
   medium speed 1000 duplex full
   no shutdown
   bind interface inet isp
   exit
task facility sessmgr start aggressive
task facility acsmgr start aggressive
end
```