Application Detection and Control Configuration

This chapter describes how to configure the Application Detection and Control (ADC) feature.

This chapter covers the following topics:

Configuring Dynamic Software Upgrade

This section describes how to install and configure the dynamic software upgrade plugin in the ASR 5500.

How to perform Dynamic Software Upgrade

The procedure for the dynamic software upgrade is as follows:

Step 1   Obtain the patch TAR file from your designated Cisco representative.

Step 2   Copy the patch file into the */flash* directory of the ASR 5500 chassis by the TFTP or FTP method.

```
patch plugin plugin_name filepath
patch plugin  plugin_name binary_path certificate certificate_path signature signature_path
```

For example:

```
 patch plugin p2p /flash/libp2p-1.97.354.so.tgz
patch plugin p2p /flash/patch_libp2p-2.67.1.1496.so.tgz signature
/flash/patch_libp2p-2.67.1.1496.so.tgz.signature
```

Note   Specifying **signature** *signature_path* is mandatory in Trusted Builds and optional in Normal Builds.

In the above example, when P2P binary file (along with signature file) is getting patched into the system, StarOS verifies the signature and accept/reject the P2P binary file.

Step 3   After the file has been copied, install the plugin using the following **install plugin** command from the Exec mode.

```
install plugin plugin_name patch_file_name
```

For example:

```
install plugin p2p patch_libp2p-1.97.354.so.tgz
patch file patch_libp2p-1.97.354.so.tgz installed successfully
```

> **Important** The plugin will be unpacked into */flash/patch/p2p* directory. Ensure that there is enough space in the */flash* directory before installing a given patch. Verify if the file is installed correctly using the **dir /flash/patch/p2p/[patch_version_number.so]** command. For example: dir /flash/patch/p2p/libp2p-1.97.354.so

Step 4 After the patch has been successfully installed, the patch version must be configured before it can be loaded into the system. To configure the patch before any other version of patch, first check the existing plugin configuration using the following command. This command entered in the Exec mode is used to list the priorities on the configured patches.

show plugin *plugin_name*

For example:

show plugin *p2p*

> **Important** The data is read from */flash/module.sys* and if it is not available, reads the default priorities from */etc/plugin.conf* (read only) and lists the priorities.

Step 5 Configure the plugin using the **plugin** command from the Global Configuration mode, and enters the Plugin Configuration mode.

plugin *<plugin_name>*

module priority *<number>* **version** *<module_version>*

For example:

```
configure
  plugin p2p
  module priority 1 version 1.97.354
```

The plugin name must match the name of the plugin which has been copied to and unpacked on the system or an error message is displayed.

> **Important** The above configuration will be internally stored in */flash/module.sys* so that the current configuration survives an ASR 5500 reload. Ensure that the file is not deleted by mistake.

Step 6 Enter the following command to update the specified module running in the system. Wait 5-10 seconds for the update to occur on all the PSC cards.

update module *<plugin_name>*

For example:

```
update module p2p
Update to module p2p version 1.97.354 successful
```

When the above command is issued, the priorities configured using **module priority** command is converted into a Module priority List and then the module with least priority is loaded. If it fails to load, the module with next higher priority is attempted and so on till a successful load occurs.

> **Important** If none of the configured modules load properly, then the system automatically tries to load the default patch that comes along with the currently loaded build.

Step 7 Enter the following command to rollback a running patch version in the system:

rollback module *<plugin_name>*

For example:

```
rollback module p2p
rollback plugin p2p module priority 10 library /var/opt/lib/libp2p-1.97.357.so load successful
```

When the above command is issued, the system automatically tries to load the patch with next highest priority in the VPL and if that fails, it tries to load the next one and so on.

Step 8 Enter the following command in Exec mode to view the status of the currently running VPL. Ensure that the patch with least priority is loaded. Others will have the "loaded" column displayed as "no".

show module *<plugin_name>* **[verbose]**

For example:

```
show module p2p
Module p2p
Priority   version  loaded    location      update/rollback time    status
    1   1.97.354    no   /var/opt/lib  FRI Feb 28 07:15:42 2014   success
    2   1.97.357    yes  /var/opt/lib  FRI Feb 28 07:15:42 2014   success
    X   1.82.118    no      /lib            (never)            N/A
```

Step 9 Delete older patch files. Unconfigure plugin from configuration and manually delete the older patch files from */flash/patch/<plugin_name>* directory to save disk space if necessary.

plugin *<plugin_name>*
no module priority *<plugin_number>*

For example:

```
configure
  plugin p2p
    no module priority 1
    end
del /flash/patch/p2p/libp2p-1.97.354.so
del /flash/patch/p2p/patch_libp2p-1.97.354.so.tgz
```

Step 10 For an ICSR setup:

If the ASR 5500 system is in ICSR mode (geographical redundancy) then the operator has to repeat the above steps for update/rollback in both the systems individually. If it is not done, then after an SRP switchover the new active ASR 5500 comes up with an outdated plugin priority which can lead to loading an older version of patch for a particular plugin. Using the **srp validate-configuration** command, check for the same plugin module priority and raise error if it is different across the active-standby pair.

Install Software with New Naming Convention Scheme

The procedure for installing software with new naming convention scheme is as follows:

Before you begin

Step 1 Obtain the patch TAR file from your designated Cisco representative.

Step 2 Copy the patch file into the */flash* directory of the ASR 5500 chassis by the TFTP or FTP method.

patch plugin *plugin_name filepath*

For example:

```
patch plugin p2p /flash/patch_libp2p-2.65.0.1468.so.tgz
```

Step 3 After the file has been copied, install the plugin using the following **install plugin** command from the Exec mode.

install plugin *plugin_name patch_file_name*

For example:

```
install plugin p2p patch_ patch_libp2p-2.65.0.1468.so.tgz
patch file patch_libp2p-2.65.0.1468.so.tgz installed successfully
```

Important The plugin will be unpacked into */flash/patch/p2p* directory. Ensure that there is enough space in the */flash* directory before installing a given patch. Verify if the file is installed correctly using the **dir /flash/patch/p2p/[patch_version_number.so]** command. For example: dir /flash/patch/p2p/libp2p-2.65.0.1468.so

Step 4 After the patch has been successfully installed, the patch version must be configured before it can be loaded into the system. To configure the patch before any other version of patch, first check the existing plugin configuration using the following command. This command entered in the Exec mode is used to list the priorities on the configured patches.

show plugin *plugin_name*

For example:

show plugin *p2p*

Important The data is read from */flash/module.sys* and if it is not available, reads the default priorities from */etc/plugin.conf* (read only) and lists the priorities.

Step 5 Configure the plugin using the **plugin** command from the Global Configuration mode, and enters the Plugin Configuration mode.

plugin *<plugin_name>*

module priority *<number>* **version** *<module_version>*

For example:

```
configure
  plugin p2p
  module priority 1 version 2.65.1468
```

The plugin name must match the name of the plugin which has been copied to and unpacked on the system or an error message is displayed.

Important The above configuration will be internally stored in */flash/module.sys* so that the current configuration survives an ASR 5500 reload. Ensure that the file is not deleted by mistake.

Step 6 Enter the following command to update the specified module running in the system. Wait 5-10 seconds for the update to occur on all the PSC cards.

update module *<plugin_name>*

For example:

```
update module p2p
Update to module p2p version 2.65.1468 successful
```

When the above command is issued, the priorities configured using **module priority** command is converted into a Module priority List and then the module with least priority is loaded. If it fails to load, the module with next higher priority is attempted and so on till a successful load occurs.

> **Important** If none of the configured modules load properly, then the system automatically tries to load the default patch that comes along with the currently loaded build.

Step 7 Enter the following command to rollback a running patch version in the system:

`rollback module` *`<plugin_name>`*

For example:

```
rollback module p2p
rollback plugin p2p module priority 10 library /var/opt/lib/libp2p-2.65.1468.so load suc-cessful
```

When the above command is issued, the system automatically tries to load the patch with next highest priority in the VPL and if that fails, it tries to load the next one and so on.

Step 8 Enter the following command in Exec mode to view the status of the currently running VPL. Ensure that the patch with least priority is loaded. Others will have the "loaded" column displayed as "no".

`show module` *`<plugin_name>`* **`[ verbose ]`**

Step 9 Delete older patch files. Unconfigure plugin from configuration and manually delete the older patch files from */flash/patch/<plugin_name>* directory to save disk space if necessary.

`plugin` *`<plugin_name>`*
`no module priority` *`<plugin_number>`*

For example:

```
configure
  plugin p2p
    no module priority 1
    end
del /flash/patch/p2p/libp2p-2.65.1468.so
del /flash/patch/p2p/patch_libp2p-2.65.1468.so.tgz
```

Step 10 For an ICSR setup:

If the ASR 5500 system is in ICSR mode (geographical redundancy) then the operator has to repeat the above steps for update/rollback in both the systems individually. If it is not done, then after an SRP switchover the new active ASR 5500 comes up with an outdated plugin priority which can lead to loading an older version of patch for a particular plugin. Using the **srp validate-configuration** command, check for the same plugin module priority and raise error if it is different across the active-standby pair.

What to do next

Configuring System for ADC

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

To configure the system for ADC support with ECS:

Step 1 Set initial configuration parameters such as modifying the local context as described in the Initial Configuration section.

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

Step 3   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.

Initial Configuration

To perform initial system configuration for ADC support with ECS:

Step 1   Enable ACS as described in the Enabling Enhanced Charging, on page 6 section.

Step 2   Set local system management parameters as described in the Modifying the Local Context, on page 6 section.

Enabling Enhanced Charging

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

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

> ☞
>
> **Important**   After you configure the **require active-charging** 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.

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 <ipv4/ipv6_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
```

ADC Configuration

To configure ADC with ACS:

Step 1   Create the ACS service as described in the Creating the Active Charging Service, on page 8 section.

Step 2   Configure ADC rules as described in the Configuring ADC Rules section.

Step 3   Configure P2P protocol groups as described in the Configuring P2P Protocol Groups, on page 9 section.

Step 4   Configure behavioral traffic as described in the Configuring Behavioral Detection, on page 10 section.

Step 5   Configure the P2P Advertisement server correlation group as described in the Configuring P2P Advertisement server, on page 11 section.

Step 6   Configure SSL renegotiation as described in the Configuring SSL Renegotiation section.

Step 7   Configure analyzers as described in the Configuring Analyzers section.

Step 8   Configure the charging action as described in the Configuring the Charging Action, on page 12 section.

Step 9   Configure the rulebase as described in the Configuring the Rulebase section.

Step 10   *Optional:* Set EDR formats as described in the Setting EDR Formats section.

Step 11   Configure the IP protocol and server port mapping for EDRs as described in the Configuring IP Protocol and Server port mapping, on page 15 section.

Step 12   Configure the P2P events for generation of EDRs as described in the Configuring EDR for P2P Events, on page 15 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
```

☞

Important   The **p2p-dynamic-rules protocol all** CLI command under Active Charging service is deprecated, and not supported in 12.0 and later releases.

Configuring ADC Rules

Use the following configuration example to set the P2P detection protocols in the ACS and the rule definitions for each P2P protocol.

☞

Important   The list of P2P protocols will be populated based on the currently installed plugin.

```
configure
 active-charging service <acs_service_name>
 p2p-detection protocol all
 ruledef <charging_ruledef_actionvoip>
 p2p protocol = actionvoip
 exit
 ruledef <charging_ruledef_facebook>
 p2p protocol = facebook
 exit
 ruledef <charging_ruledef_jabber>
 p2p protocol = jabber
 exit
 ruledef <charging_ruledef_skype>
 p2p protocol = skype
 exit
# Configuration example to report audio, file transfer, instant messaging, video, voipout
and unclassified components:
 ruledef <charging_ruledef_<protocol>_audio>
 p2p protocol = <protocol>
 p2p traffic-type = audio
 exit
 ruledef <charging_ruledef_<protocol>_ft>
 p2p protocol = <protocol>
 p2p traffic-type = file-transfer
 exit
 ruledef <charging_ruledef_<protocol>_im>
 p2p protocol = <protocol>
 p2p traffic-type = im
 exit
 ruledef <charging_ruledef_<protocol>_video>
 p2p protocol = <protocol>
```

```
p2p traffic-type = video
exit
ruledef <charging_ruledef_<protocol>_voipout>
p2p protocol = <protocol>
p2p traffic-type = voipout
exit
ruledef <charging_ruledef_<protocol>_unclassified>
p2p protocol = <protocol>
p2p traffic-type = unclassified
exit
end
```

Configuring ADC Ruledef Types

Use the following configuration to configure ADC ruledefs in support of the ADC over Gx feature:

```
configure
    active-charging service service_name
        ruledef adc_rule_type1
            p2p protocol = protocol_name
            p2p protocol-group = protocol_group
            p2p behavioral = behavioral_list
            multi-line-or all-lines
            exit
        ruledef adc_rule_type2
            p2p protocol = protocol_name
            p2p traffic-type = traffic-type
            exit
        ruledef adc_rule_type2
            p2p any-match = TRUE
            exit
```

Sample Ruledef Configuration for Windows Updates

Use the following ruledef CDP configuration for detecting Windows Updates:

```
Ruledef Name: windowsupdate
tls sni ends-with windowsupdate.com
tls sni ends-with update.microsoft.com
http host ends-with windowsupdate.com
http host ends-with update.microsoft.com
tls set-app-proto windowsupdate
Rule Application Type: Charging
Copy Packet to Log: Disabled
Tethered Flow Check: Disabled
Multi-line OR: All Lines
```

Configuring P2P Protocol Groups

Use the following configuration example to configure P2P protocol groups:

```
configure
    active-charging service <acs_service_name>
        ruledef <ruledef_name>
```

```
                         [ no ] p2p protocol-group <operator> <group_list>
                         end
```

Notes:

<group_list> must be one of the following:

- **anonymous-access**
- **business**
- **communicator**
- **cloud**
- **e-mail**
- **e-news**
- **e-store**
- **internet-privacy**
- **filesharing**
- **gaming**
- **p2p-anon-filesharing**
- **p2p-filesharing**
- **remote-control**
- **social-nw-gaming**
- **social-nw-generic**
- **social-nw-videoconf**
- **standard**
- **streaming**
- **untagged**

For more information, refer to the *ACS Ruledef Configuration Mode Commands* chapter of the *Command Line Interface Reference*.

Configuring Behavioral Detection

Use the following configuration example to configure behavioral detection of unidentified traffic:

```
configure
     active-charging service <acs_service_name>
          [ no ] p2p-detection behavioral <behavioral_list>
          end
```

Use the following to define rule expressions to match behavioral detection type — P2P, Video, VoIP, Behavioral Upload or Behavioral Download.

```
configure
     active-charging service <acs_service_name>
          ruledef <ruledef_name>
               [ no ] p2p behavioral <operator> <behavioral_list>
               end
```

Notes:

- Here the *<behavioral_list>* is the list of supported behavioral detection logic populated from the currently loaded P2P plugin. The supported behavioral list is:

    - **download**: Detects unknown flows which are data download using behavioral analysis

- **p2p**: Detects P2P and file sharing protocols using behavioral analysis

- **upload**: Detects unknown flows which are data upload using behavioral analysis

- **video**: Detects video flows using behavioral analysis

- **voip**: Detects VoIP (voice and video) protocols using behavioral analysis

Behavioral P2P, behavioral video and behavioral VoIP are meant for zero day detection of P2P/file sharing protocols, video traffic and VoIP traffic respectively. Behavioral Upload/Download must detect flows of non-standard ports that cannot be detected by ECS. This feature is meant only for statistical purposes (not for charging purposes). For more information, refer to the *ACS Configuration Mode Commands* and *ACS Ruledef Configuration Mode Commands* chapters of the *Command Line Interface Reference*.

Configuring P2P Advertisement server

Use the following configuration to configure the P2P Advertisement server correlation group:

```
configure
   active-charging service acs_service_name
      [ no ] p2p ads-group ads_group_name
         [ no ] ad-source operator http_host_name/ssl_server_name
         [ no ] map-to-application { p2p_list } +
         end
```

Notes:

- On entering this command, the CLI prompt changes to the P2P Advertisement Server Group Configuration Mode:

  [context_name]hostname(config-acs-p2p-ads)#

- *ads_group_name* must be an alphanumeric string of 1 through 63 characters.

- The following two commands are supported in the new P2P Advertisement Server Group Configuration Mode.

  - **ad-source** *operator http_host_name/ssl_server_name*: Configures the P2P Advertisement source that can be a HTTP host or SSL server. SSL supports the Server Name indication (SNI) field.

    operator can be "=", "contains", "ends-with" or "starts-with".

    http_host_name/ssl_server_name must be an alphanumeric string of 1 through 127 characters.

  - **map-to-application** *p2p_list*: Configures the P2P advertisement application that will map the advertisement group to the corresponding application/protocol. *p2p_list* is the list of protocols/applications supported in the P2P plugin. The maximum number of map-to-application rule lines that can be configured is equal to the number of the applications present in *p2p_list*.

- The existing analyzer statistics and EDR will accumulate P2P related statistics based on the ads-group configuration. Bulk statistics will accumulate "ads" subtype statistics for the configured protocol in p2p-ads-group.

- The existing ruledef configuration will be used to configure any charging action.

For more information, refer to the *ACS Configuration Mode Commands* and *P2P Advertisement Server Group Configuration Mode Commands* chapters of the Command Line Interface Reference.

Configuring SSL Renegotiation

Use the following configuration example to configure SSL renegotiation:

```
configure
     active-charging service <acsservice_name>
          [ no ] p2p-detection attribute { <attribute_list> [
<sub_attribute_name> <sub_attribute_value> ] }
               end
```

Notes:

- Here the *<attribute_list>* is the list of configurable P2P detection attributes populated from the currently loaded P2P plugin.

  Supported attribute: **ssl-renegotiation**

- *<sub_attribute_name>* is the list of configurable P2P detection sub-attributes related to the attribute selected from the attribute list. This list is populated from the currently loaded P2P plugin.

  Supported sub-attributes if selected attribute is **ssl-renegotiation**:

  - **max-entry-per-sessmgr**

  - **id-reduce-factor**

- *<sub_attribute_value>* is the value of the selected sub-attribute. If sub-attribute is not specified, the default value set in the P2P plugin will be used.

For more information, refer to the *ACS Configuration Mode Commands* chapter of the *Command Line Interface Reference*.

Configuring Analyzers

Use the following configuration example to configure analyzers for ECS analysis:

```
configure
     require active-charging
     active-charging service <acs_service_name>
          [ no ] p2p-detection ecs-analysis { all | ftp | http | rtsp
| sip }
               end
```

> **Important** After you configure **require active-charging** and **active-charging service***<acs_service_name>* commands, 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.

For more information, refer to the *ACS Configuration Mode Commands* chapter of the *Command Line Interface Reference*.

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.

```
configure
 active-charging service <acs_service_name>
 rulebase <rulebase_name>
 action priority action_priority { [ dynamic-only { adc [ mute ] } |
static-and-dynamic | timedef timedef_name ] { group-of-ruledefs
ruledefs_group_name | ruledef ruledef_name } charging-action charging_action_name [
 monitoring-key monitoring_key ] [ description description ] }
# Configuration
example to detect P2P applications configured for the Active Charging
Service:
 action priority <priority> ruledef <charging_ruledef_actionvoip> charging-action
 <charging_action_name>
 action priority <priority> ruledef <charging_ruledef_facebook> charging-action
<charging_action_name>
 action priority <priority> ruledef <charging_ruledef_jabber> charging-action
<charging_action_name>
 action priority <priority> ruledef <charging_ruledef_skype> charging-action
<charging_action_name>
# Configuration example to report audio, file transfer, instant messaging, video, voipout
and unclassified components:
 action priority <priority> ruledef <charging_ruledef_protocol_audio>
charging-action <charging_action_name>
 action priority <priority> ruledef <charging_ruledef_protocol_ft> charging-action
 <charging_action_name>
 action priority <priority> ruledef <charging_ruledef_protocol_im> charging-action
 <charging_action_name>
 action priority <priority> ruledef <charging_ruledef_protocol_video>
charging-action <charging_action_name>
 action priority <priority> ruledef <charging_ruledef_protocol_voipout>
charging-action <charging_action_name>
 action priority <priority> ruledef <charging_ruledef_protocol_unclassified>
charging-action <charging_action_name>
 end
 end
```

Notes:

- The **adc** keyword option specifies the ruledef to be given as ADC rule. This predefined rule can be activated from PCRF/Gx. This can be configured only with the **dynamic-only** keyword and optional along with ruledef. Group-of-ruledefs is not supported in this release.
- The **mute** keyword is optional and can be configured only with the **adc** keyword. This keyword option will disable ADC application reporting to PCRF, that is, will mute the Application Start and Application Stop notifications to PCRF/Gx. Detection of protocols in the rule will still happen. Whenever the application traffic matches the specified ruledef for the first time in that flow, it is considered as Application Start. At the end of flow, it is considered as Application Stop.

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 p2p 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-port-service-name priority <priority>
         attribute sn-app-protocol priority <priority>
         attribute sn-parent-protocol priority <priority>
         rule-variable ip protocol priority <priority>
         attribute sn-ip-protocol-name priority <priority>
         rule-variable p2p protocol priority <priority>
         rule-variable p2p protocol-group 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
```

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*.

Configuring IP Protocol and Server port mapping

Use the following configuration to enable IP protocol and server port mapping for EDRs:

```
configure
 active-charging service <acs_service_name>
 [ default | no ] edr-ipproto-port-map
 end
```

For information, refer to the *ACS Configuration Mode Commands* chapter of the *Command Line Interface Reference*.

Configuring EDR for P2P Events

ADC supports sub-protocol detection for certain applications like Skype, Yahoo, MSN, GTalk, etc. Along with sub-protocol tracking, the application detection logic can track start and end duration of audio/video flow. This audio-end/video-end tracking heavily depends on the audio/video patterns used by the application detection logic. For example, in the case of audio flow start for a sub-protocol tracking, application detection logic begins tracking the start time and if that flow toggles to another sub-protocol that is video or unclassified, the end time for that sub-protocol is set and the EDR for that flow is dumped. Since certain applications use simultaneous flows for audio/video, parallel flow is tracked for all such flows instead of tracking separated audio/video flows.

Use the following configuration to generate EDRs for P2P events. This command is associated with the Dynamic Software Upgrade process.

```
configure
   active-charging service <acs_service_name>
      rulebase <rulebase_name>
         edr p2p <p2p_event_list> [ charging-edr <charging_edr_format_name> |
edr-format <edr_format_name> | reporting-edr <reporting_edr_format_name> ] +
         end
```

☞

Important   When the 1.97.357 ADC Plugin is used, Voice Duration details in EDR logs are missing. This has been fixed in plugins later than 1.97.357.

Notes:

- The plugin supports only the "audio-end" and "video-end" events. The P2P event list can be any P2P event that is supported by the plugin.

- The EDR generated for audio-end/video-end must not be used for flow analysis, and must be used only for audio/video duration analysis.

- For audio/video duration analysis (for example, VCD reports), the "sn-closure-reason" field must be checked to identify the reason for closure and the appropriate EDR chosen to generate the reports. The "sn-closure-reason" field will be set to **14** for capturing audio-end and video-end generated EDRs. For example, in case of audio-end events, all VoIP call related statistics such as VoIP duration and bytes/packets are captured along with traffic-type field set as "audio" and sn-closure-reason set as "14".

For information, refer to the *ACS Rulebase Configuration Mode Commands* chapter of the *Command Line Interface Reference*.

Gathering ADC 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	**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	**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	**show active-charging flows type p2p traffic-type audio** **show active-charging flows type p2p traffic-type file-transfer** **show active-charging flows type p2p traffic-type im** **show active-charging flows type p2p traffic-type video** **show active-charging flows type p2p traffic-type voipout** **show active-charging flows type p2p traffic-type unclassified**	The output of this command displays the number of P2P audio, file transfer, instant messaging, video, voipout and unclassified flows.
Charging Action information	**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	**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.

Statistics Wanted	Action to Perform	Information to Look For
Sessions using specific protocol	**show active-charging sessions type p2p application** *<protocol>* **show active-charging sessions full all**	The output of this command displays information for the sessions using the specified protocol.
Total and current P2P flows and P2P audio, file-transfer, im, video, voipout, or unclassified flows	**show active-charging subsystem all**	The output of this command displays total and current P2P flows and P2P audio/file-transfer/instant messaging/video/voipout/unclassified flow statistics, and total number of subscribers.
Voice Statistics	**show active-charging analyzer statistics name p2p application [actionvoip | facetime | gtalk | icall | jumblo | magicjack | msn | oscar | plingm | rynga | skype | smartvoip | talkatone | voipdiscount | vopium | yahoo] verbose**	The output of this command displays the voice and non-voice analyzer statistics for voice supported protocols.
P2P Protocol Group Statistics	**show active-charging analyzer statistics name p2p protocol-group wide all verbose**	The output of this command displays the P2P protocol group statistics if a P2P analyzer is used.
Subscriber Readdress Statistics	**show subscribers callid <callid> adc readdress statistics**	In support of the ADC over Gx feature, the output of this command displays readdress statistics at subscriber level for a given call ID.

Supported Bulk Statistics

ADC bulk statisticsa are available as part of the P2P schema. If detection of a specific P2P protocol is enabled, bulk statistics for that protocol will be automatically generated using the Dynamic Software Upgrade plugin installed on the chassis. In the case of protocols that support sub-classification (audio/video/unclassified), the bulk statistics will be generated for each of the supported sub-classifications per protocol and also the corresponding cumulative count.

For information on ADC 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 9.0 and earlier releases, the P2P reporting functionality was available in the Web Element Manager software. For more information, refer to the *Web Element Manager Online Help* documentation.

☞

Important In 10.0 and later releases, the P2P reporting functionality is supported in Mobility Unified Reporting (MUR) / Mobility Unified Reporting and Analytics (MURAL) system. For more information on MUR, refer to the *Mobility Unified Reporting Online Help* documentation. For more information on MURAL support, refer to the *MURAL Installation and Administration Guide* and *MURAL User Guide*.

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* documentation.