

Chapter 22

Configuring Congestion Point Classification with the SRC CLI

This chapter describes how to use the SRC command-line interface (CLI) to configure congestion point classification in the SRC Admission Control Plug-In (SRC-ACP) application. You can use the CLI to configure SRC-ACP on a Solaris platform or on a C-series Controller.

You can also use SRC configuration applications to configure congestion point classification on a Solaris platform. See *Chapter 26, Providing Admission Control with SRC-ACP on a Solaris Platform*.

Topics in this chapter include:

- Overview of Congestion Point Classification on page 315
- Configuration Statements for Congestion Point Classification on page 316
- Classifying Congestion Points on page 316
- Defining a Congestion Point Profile on page 321

Overview of Congestion Point Classification

Congestion point classification allows you to automate and scale the configuration of congestion points. SRC-ACP uses classification scripts to determine which congestion point to load for a subscriber. SRC-ACP can select the congestion point from congestion point profiles or subscriber profiles.

Congestion Point Classification Scripts

The congestion point classification scripts consist of targets and criteria.

- A target is the result of the classification script. The result of congestion point classification scripts is an LDAP search string that is used to find a unique congestion point in the directory. If no classification scripts are configured, the result of congestion point classification scripts is an LDAP search string for the subscriber profile of the particular subscriber.

- Criteria are match criteria. The script attempts to match criteria in the script to information sent from the router. Match criteria for a congestion point classification script might be a subscriber distinguished name (DN) or an interface name.

Each script can have multiple targets, and each target can have multiple criteria. When an object needs classification, the script processes the targets in turn. Within each target, the script processes criteria sequentially. When it finds that the classification criteria for a target match, it returns the target to SRC-ACP.

Because classification scripts examine criteria sequentially as the criteria appear in the script, you should put more specific criteria at the beginning of the script and less specific criteria at the end of the script.

Congestion Point Profiles

Congestion point profiles are used to share congestion points that are generated based on dynamic configuration information. SRC-ACP uses congestion point profiles to determine the set of congestion points based on the classification script results.

Configuration Statements for Congestion Point Classification

Use the following configuration statements to configure congestion point classification at the [edit] hierarchy level.

```
shared acp congestion-point-classifier rule name {
  target target;
  script script;
}
```

```
shared acp congestion-point-classifier rule name condition name ...
```

```
shared congestion-points congestion-point-profile name {
  expression [expression...];
}
```

For detailed information about each configuration statement, see the *SRC-PE CLI Command Reference*.

Classifying Congestion Points

Use the following configuration statements to configure congestion point classification scripts:

```
shared acp congestion-point-classifier rule name {
  target target;
  script script;
}
```

```
shared acp congestion-point-classifier rule name condition name ...
```

Changes that you make to classification scripts do not affect subscriber sessions that are already established.

Configuring Targets and Criteria for Classification Scripts

To define a target and criteria for the congestion point classification script:

1. From configuration mode, access the configuration statement that configures congestion point scripts. In this sample procedure, the scripts are configured in the config group.

```
user@host# edit shared acp group config congestion-point-classifier rule name
```

Enter a name for the congestion point classification script.

2. Specify the target for the classification script.

```
[edit shared acp group config congestion-point-classifier rule name]
user@host# set target target
```

For information about classification targets, see *Configuring Congestion Point Classification Targets* on page 318.

3. Specify the classification criteria for the target.

```
[edit shared acp group config congestion-point-classifier rule name]
user@host# set script script
```

For information about classification criteria, see *Selecting Congestion Point Classification Criteria* on page 318.

Configuring Classification Scripts Contents for Classification Scripts

To use the contents of a classification script to another object for the congestion point classification script:

- Access the configuration statement that configures congestion point scripts from configuration mode. In this sample procedure, the scripts are configured in the config group.

```
user@host# edit shared acp group config congestion-point-classifier rule name
condition name ...
```

Enter a name for the congestion point classification script and the name of the classification script that you want to use.

Configuring Congestion Point Classification Targets

The target of the congestion point classification script is an LDAP search string. The search string uses a syntax similar to an LDAP URL (see RFC 2255—The LDAP URL Format (December 1997)). The syntax is:

```
baseDN [ ? [ attributes ] [ ? [ scope ] [ ? [ filter ] ] ] ]
```

- baseDN—Distinguished name (DN) of the object where the LDAP search starts.
- attributes—Is ignored.
- scope—Scope of search in the directory:
 - base—Default; searches the base DN only.
 - one—Searches the direct children of the base DN.
 - sub—Searches the complete subtree below the base DN.
- filter—An RFC 2254-style LDAP search filter expression; for example, (uniqueId = <-userName- >). See RFC 2254—The String Representation of LDAP Search Filters (December 1997).

With the exception of baseDN all the fields are optional.

The result of the LDAP search must be exactly one directory object. If no object or more than one object is found, congestion points for the subscriber are not loaded and all service activations for the subscriber are denied.

Selecting Congestion Point Classification Criteria

Congestion point classification criteria define match criteria that are used to find the congestion point profile. Use the fields in this section to define classification criteria.

accountingId

- Value of directory attribute accountingUserId.

authUserId

- Identifier that a subscriber uses for authentication.
- Value—Username

dhcpPacket

- Content of the DHCP discover request.
- Value—Byte array
 - First 4 octets—Gateway IP address (giaddr field)
 - Remaining octets—DHCP options

For more information, see RFC 2131—Dynamic Host Configuration Protocol (March 1997) and RFC 2132—DHCP Options and BOOTP Vendor Extensions (March 1997).

domain

- Name of the domain used for secondary authentication.
- Value—Valid domain name
- Example—domain = “isp99.com”

ifRadiusClass

- RADIUS class attribute on the JUNOSe interface.
- Value—RADIUS class name
- Example—ifRadiusClass = “acpe”

ifSessionId

- Identifier for RADIUS accounting on the JUNOSe interface.

interfaceAlias

- Description of the interface.
- Value—Interface description that is configured on the JUNOSe router with the `interface ip description` command
- Example—interfaceAlias = “dhcp-subscriber12”

interfaceDescr

- Alternate name for the interface that is used by SNMP. This name is a system-generated name.
- Value
 - On a JUNOSe router, the format of the description is
ip<slot>/<port>.<subinterface>
 - On the JUNOS routing platform, interfaceDescr is the same as interfaceName.
- Example—interfaceDescr = “IP3/1”

interfaceName

- Name of the interface.
- Value
 - Name of the interface in your router CLI syntax
 - FORWARDING_INTERFACE for routing instance (used by traffic mirroring)
- Example—For JUNOSe routers: interfaceName = “fastEthernet6/0”
For JUNOS routing platforms: interfaceName = “fe-0/1/0.0”
For forwarding interface: interfaceName = “FORWARDING_INTERFACE”

loginName

- Subscriber's login name.
- Value—Login name
- Guidelines—The format of the login name varies. A loginName can be of form subscriber, domain\subscriber, subscriber@domain, or as otherwise defined by the login setup of the manager.
- Example—idp@idp

nasIp

- IP address of the router.
- Value—Byte array
 - For IPv4 address—4 octets in network byte order
 - For IPv6 address—16 octets in network byte order

nasPort

- Port identifier of an interface.
- Value—Includes interface name and additional layer 2 information
- Example—nasPort = “fastEthernet 3/1” (There is a space between fastEthernet and slot number 3/1 in the nasPort field.)

portId

- Identifier of VLAN or virtual circuit.
- Value—String; for a virtual circuit, use the format <VPI> / <VCI>

primaryUserName

- PPP login name or the public DHCP username.
- Value—Subscriber name
- Example—primaryUserName = “peter”

radiusClass

- RADIUS class attribute of the service definition.
- Value—RADIUS class name
- Example—radiusClass = “Premium”

routerName

- Name of virtual router.
- Value—Virtual router name in the format <virtualRouter> @ <router>
- Example—routerName = “default@e_series5”

sessionId

- Identifier of RADIUS session for the subscriber session.

serviceBundle

- Content of the RADIUS vendor-specific attribute for the service bundle.
- Value—Name of a service bundle
- Example—serviceBundle = “goldSubscriber”

sspHost

- Name of host on which the SAE is installed.

userDn

- DN of a subscriber in the directory.
- Value—DN of a subscriber profile

userIp

- IP address of the subscriber.
- Value—Byte array
 - For IPv4 address—4 octets in network byte order
 - For IPv6 address—16 octets in network byte order

userMacAddress

- Media access control (MAC) address of the DHCP subscriber.
- Value—Valid MAC address
- Example—userMacAddress = “00:11:22:33:44:55”

userType

- Type of subscriber.

Defining a Congestion Point Profile

You can create a congestion point profile that automatically performs congestion point classification. This profile supports only access network mode for SRC-ACP.

Use the following configuration statements to configure congestion point profiles:

```
shared congestion-points congestion-point-profile name {
    expression [expression...];
}
```

To define a congestion point profile:

1. From configuration mode, access the configuration statement that configures congestion point profiles.

```
user@host# edit shared congestion-points congestion-point-profile name
```

Enter a name for the profile.

- Specify congestion point expressions.

```
[edit shared congestion-points congestion-point-profile name]
user@host# set expression [expression...]
```

For information about congestion point expressions, see *Congestion Point Expressions* on page 322.

Congestion Point Expressions

You can enter a congestion point expression by using the syntax listed in this section. You can also embed Python scripting expressions within the congestion point expression.

If you embed Python expressions within a congestion point expression, use the escape sequence `<- then ->` to enclose the Python expression. See *Methods for Use with Scripting Expressions* on page 323 and *Match Criteria for Congestion Point Classification* on page 323.

The syntax for a congestion point expression is:

```
< NetworkDevice > / < NetworkInterface > [ / < CongestionPoint > ]
```

- `< NetworkDevice >` —Network device listed in the directory.

For information about network devices, see *SRC-PE Network Guide, Part 2, Using Juniper Networks Routers in the SRC Network*.

- `< NetworkInterface >` —Network interface listed in the directory.

For information about interfaces, see *SRC-PE Subscribers and Subscriptions Guide, Chapter 6, Classifying Interfaces and Subscribers with the SRC CLI*.

- `< CongestionPoint >` —(Optional) Name of an instance of a congestion point that is automatically created.

If one of the elements with the path contains a slash (`/`), use a backslash (`\`) as an escape character for the slash. For example, `\.`

Expressions in Templates for Congestion Point Profiles

You can create a congestion point profile to be used as a template for other profiles. Templates simplify management of congestion points. Rather than configuring each congestion point individually, you can create templates to define common parameters for a class of individual congestion points.

For example, in an environment in which VLAN interfaces GigabitEthernet1/0.1 through GigabitEthernet1/0.1000 have the same available bandwidth, you can specify the characteristics of the VLAN interface once and have SRC-ACP create the congestion points based on the template configuration.

When a congestion point expression has the third element (<CongestionPoint>), SRC-ACP uses the <NetworkDevice>/<NetworkInterface> part of the expression to load the congestion point from the directory, and uses it as a template to create a congestion point in memory for subscriber. The <CongestionPoint> part of the expression distinguishes each congestion point (available bandwidth) created from this template.

Methods for Use with Scripting Expressions

SRC-ACP provides the following methods to use in scripting expressions:

- `slot(nasPortId)`—Collects the slot number from the `nasPortId` or `interfaceName`
Example—`slot("atm 4/5:0.32")` = = "4"
- `port(nasPortId)`—Collects the port number from the `nasPortId` or `interfaceName`
Example—`port("atm 4/5:0.32")` = = "5"
- `l2id(nasPortId)`—Collects the layer 2 ID from the `nasPortId` (VLAN id or ATM vpi.vci)
Example—`l2id("atm 4/5:0.32")` = = "0.32"
- `escape(string)`—Replaces any slash with the escape sequence `\`
Example—`escape("atm 4/5")` = = "atm 4\5"

Match Criteria for Congestion Point Classification

You can use the match criteria in Python scripting expressions for a congestion point expression. For more information about the match criteria, see *Selecting Congestion Point Classification Criteria* on page 318.

