CHAPTER 3

Configuring Traffic Policies for Server Load Balancing

This chapter describes how to configure the ACE module to use classification (class) maps and policy maps to filter and match interesting network traffic based on various criteria and load balance the traffic to real servers in server farms using one of the ACE load-balancing predictor methods.

This chapter contains the following major sections:

- Overview of SLB Traffic Policies
- Layer 7 SLB Traffic Policy Configuration Quick Start
- Layer 3 and Layer 4 SLB Traffic Policy Configuration Quick Start
- Configuring HTTP Header Insertion, Deletion, and Rewrite
- Configuring a Layer 7 Class Map for Generic TCP and UDP Data Parsing
- Configuring a Layer 7 Class Map for SLB
- Configuring a Layer 7 Policy Map for SLB
- Configuring a Generic Protocol Parameter Map
- Configuring an HTTP Parameter Map
- Configuring an RTSP Parameter Map
- Configuring a Layer 3 and Layer 4 Class Map for SLB
- Configuring a Layer 3 and Layer 4 Policy Map for SLB
- Applying a Layer 3 and Layer 4 Policy to an Interface
- Configuring UDP Booster
Overview of SLB Traffic Policies

You classify inbound network traffic destined to, or passing through, the ACE based on a series of flow match criteria specified by a class map. Each class map defines a traffic classification, which is network traffic that is of interest to you. A policy map defines a series of actions (functions) that you want applied to a set of classified inbound traffic.

ACE traffic policies support the following server load-balancing (SLB) traffic attributes:

- Layer 3 and Layer 4 connection information—Source or destination IP address, source or destination port, virtual IP address, and IP protocol
- Layer 7 protocol information—Hypertext Transfer Protocol (HTTP) cookie, HTTP URL, HTTP header, Remote Authentication Dial-In User Service (RADIUS), Remote Desktop Protocol (RDP), Real-Time Streaming Protocol (RTSP), Session Initiation Protocol (SIP), and Secure Sockets Layer (SSL)

The three steps in the traffic classification process are as follows:

1. Create a class map using the `class-map` command and the associated `match` commands, which comprise a set of match criteria related to Layer 3 and Layer 4 traffic classifications or Layer 7 protocol classifications.
2. Create a policy map using the **policy-map** command, which refers to the class maps and identifies a series of actions to perform based on the traffic match criteria.

3. Activate the policy map by associating it with a specific VLAN interface or globally with all VLAN interfaces using the **service-policy** command to filter the traffic received by the ACE.

**Figure 3-1** provides a basic overview of the process required to build and apply the Layer 3, Layer 4, and Layer 7 policies that the ACE uses for SLB. The figure also shows how you associate the various components of the SLB policy configuration with each other.
Figure 3-1  Server Load-Balancing Configuration Flow Diagram

1. Class Map (Layer 7)
   (config)# class-map type http loadbalance CLASSMAP_L7
   Defines Layer 7 SLB match criteria applied to input traffic:
   - Class map
   - HTTP cookie
   - HTTP header
   - HTTP URL
   - Source IP address
   L7 class map associated with L7 policy map

2. Policy Map (Layer 7)
   (config)# policy-map type loadbalance first-match POLICYMAP_L7
   Specifies match criteria (class map) and action:
   - Class map
     - Drop
     - Forward
     - Insert HTTP
     - Server farm
   - Set IP TOS
   - SSL proxy client
   - Sticky server farm

3. Layer 7 HTTP Parameter Map
   (config)# parameter-map type http loadbalance HTTP_PARAMMAP
   Defines related advanced Layer 7 parameters for SLB:
   - Case sensitivity
   - URL delimiters
   - Maximum parse length for HTTP headers, URLs, and cookies
   - Response to cookie or URL exceeding max bytes
   - HTTP persistence
   - TCP server reuse
   Parameter map associated with L3/L4 policy map

4. Class Map (Layer 3 and Layer 4)
   (config)# class-map match-any CLASSMAP_L3L4
   Defines Layer 3 and Layer 4 match criteria applied to input traffic:
   - Virtual IP address
   L3/L4 class map associated with L3/L4 policy map

5. Policy Map (Layer 3 and Layer 4)
   (config)# policy-map multi-match POLICYMAP_L3L4
   Specifies Layer 3 and Layer 4 class map and Layer 7 policy map applied to input traffic:
   - Class map
     - Load balance
     - Parameter map
   Layer 3 and Layer 4 policy map applied globally to all VLAN interfaces or to a specific VLAN interface

6. Global VLAN Application
   (config)# service-policy input POLICYMAP_L3L4
   Applies the policy map to all VLANs in the context.

   Specific VLAN Application
   (config)# interface vlan 50
   (config-if)# service-policy input POLICYMAP_L3L4
   Applies the policy map to the input of a specific VLAN.
Layer 7 SLB Traffic Policy Configuration Quick Start

Table 3-1 provides a quick overview of the steps required to configure a Layer 7 HTTP class map and a Layer 7 HTTP policy map. You use a similar procedure to configure Layer 7 class maps and policy maps for other supported protocols. Each step includes the CLI command and a reference to the procedure required to complete the task. For a complete description of each feature and all the options associated with the CLI commands, see the sections following Table 3-1.

Table 3-1  Layer 7 SLB Policy Configuration Quick Start

<table>
<thead>
<tr>
<th>Task and Command Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. If you are operating in multiple contexts, observe the CLI prompt to verify that you are operating in the desired context. If necessary, change to, or directly log in to, the correct context.</td>
</tr>
<tr>
<td>host1/Admin#  changeto C1</td>
</tr>
<tr>
<td>host1/C1#</td>
</tr>
<tr>
<td>The rest of the examples in this table use the Admin context, unless otherwise specified. For details on creating contexts, see the Cisco Application Control Engine Module Administration Guide.</td>
</tr>
<tr>
<td>2. Enter configuration mode.</td>
</tr>
<tr>
<td>host1/Admin#  config</td>
</tr>
<tr>
<td>Enter configuration commands, one per line. End with CNTL/Z</td>
</tr>
<tr>
<td>host1/Admin(config)#</td>
</tr>
<tr>
<td>3. Create a Layer 7 class map for SLB. See the “Configuring a Layer 7 Class Map for SLB” section.</td>
</tr>
<tr>
<td>host1/Admin(config)#  class-map type http loadbalance match-all L7SLBCLASS</td>
</tr>
<tr>
<td>host1/Admin(config-cmap-http-lb)#</td>
</tr>
</tbody>
</table>
Table 3-1  Layer 7 SLB Policy Configuration Quick Start (continued)

Task and Command Example

4.  Configure one or more of the following match criteria for the Layer 7 SLB class map:

- Define HTTP content for load balancing. See the “Defining an HTTP Content Match for Load Balancing” section.
  
  `host1/Admin(config-cmap-http-lb)# match http content abc*123 offset 50`

- Define a cookie for HTTP load balancing. See the “Defining a Cookie for HTTP Load Balancing” section.
  
  `host1/Admin(config-cmap-http-lb)# match http cookie TESTCOOKIE1 cookie-value 123456`

- Define an HTTP header for load balancing. See the “Defining an HTTP Header for Load Balancing” section.
  
  `host1/Admin(config-cmap-http-lb)# match http header Host header-value .*cisco.com`

- Define a URL for HTTP load balancing. See the “Defining a URL for HTTP Load Balancing” section.
  
  `host1/Admin(config-cmap-http-lb)# match http url /WHATSNEW/LATEST.*`

- Define a source IP match statement. See the “Defining Source IP Address Match Criteria” section.
  
  `host1/Admin(config-cmap-http-lb)# match source-address 192.168.11.2 255.255.255.0`
Chapter 3  Configuring Traffic Policies for Server Load Balancing

Layer 7 SLB Traffic Policy Configuration Quick Start

Table 3-1  Layer 7 SLB Policy Configuration Quick Start (continued)

<table>
<thead>
<tr>
<th>Task and Command Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>5. Use the <strong>exit</strong> command to reenter configuration mode.</td>
</tr>
<tr>
<td>host1/Admin(config-cmap-http-lb)# <strong>exit</strong></td>
</tr>
<tr>
<td>host1/Admin(config)#</td>
</tr>
<tr>
<td>6. Create a Layer 7 policy map for SLB. See the “Configuring a Layer 7 Policy Map for SLB” section.</td>
</tr>
<tr>
<td>host1/Admin(config)# <strong>policy-map type loadbalance first-match</strong> L7SLBPOLICY</td>
</tr>
<tr>
<td>host1/Admin(config-pmap-lb)#</td>
</tr>
<tr>
<td>7. Associate the Layer 7 class map that you created in Step 3 with the Layer 7 policy map that you created in Step 6. See the “Associating a Layer 7 Class Map with a Layer 7 Policy Map” section.</td>
</tr>
<tr>
<td>host1/Admin(config-pmap-lb)# <strong>class</strong> L7SLBCLASS</td>
</tr>
<tr>
<td>host1/Admin(config-pmap-lb-c)#</td>
</tr>
<tr>
<td>8. Specify one or more of the following policy-map actions that you want the ACE to take when network traffic matches a class map:</td>
</tr>
<tr>
<td>• Instruct the ACE to discard packets that match a policy map. See the “Discarding Requests” section.</td>
</tr>
<tr>
<td>host1/Admin(config-pmap-lb-c)# <strong>drop</strong></td>
</tr>
<tr>
<td>• Instruct the ACE to forward packets that match a policy map without load balancing them. See the “Forwarding Requests Without Load Balancing” section.</td>
</tr>
<tr>
<td>host1/Admin(config-pmap-lb-c)# <strong>forward</strong></td>
</tr>
<tr>
<td>• Enable HTTP header insertion. See the “Configuring HTTP Header Insertion” section.</td>
</tr>
<tr>
<td>host1/Admin(config-pmap-lb-c)# <strong>insert-http Host header-value <a href="http://www.cisco.com">www.cisco.com</a></strong></td>
</tr>
<tr>
<td>• Enable load balancing to a server farm. See the “Enabling Load Balancing to a Server Farm” section.</td>
</tr>
<tr>
<td>host1/Admin(config-pmap-lb-c)# <strong>serverfarm FARM2 backup FARM3</strong></td>
</tr>
</tbody>
</table>
Specify the IP differentiated services code point (DSCP) of packets within the traffic class. See the “Configuring a Sticky Server Farm” section.

host1/Admin(config-pmap-lb-c)# set ip tos 8

If you are using SSL Initiation (ACE acting as an SSL client), specify an SSL proxy service. See the “Specifying an SSL Proxy Service” section. For more information about SSL, see the Cisco Application Control Engine Module SSL Configuration Guide.

host1/Admin(config-pmap-lb-c)# ssl-proxy client PROXY_SERVICE1

To use stickiness (connection persistence), specify a sticky server farm for load balancing. See the “Configuring a Sticky Server Farm” section.

host1/Admin(config-pmap-lb-c)# sticky-serverfarm STICKY_GROUP1

9. Before you can use a Layer 7 policy map for load balancing, you must associate it with a Layer 3 and Layer 4 SLB policy map. Create the Layer 3 and Layer 4 class map and policy map, then associate the Layer 7 policy map with the Layer 3 and Layer 4 policy map. Finally, associate the Layer 3 and Layer 4 policy map with an interface. See the following sections:

- Configuring a Layer 3 and Layer 4 Class Map for SLB
- Configuring a Layer 3 and Layer 4 Policy Map for SLB
- Applying a Layer 3 and Layer 4 Policy to an Interface

10. Display your class-map and policy-map configurations and statistics (see the “Displaying Load-Balancing Configuration Information and Statistics” section).

    host1/Admin# show running-config class-map
    host1/Admin# show running-config policy-map

11. (Optional) Save your configuration changes to flash memory.

    host1/Admin# copy running-config startup-config
Layer 3 and Layer 4 SLB Traffic Policy Configuration Quick Start

Table 3-2 provides a quick overview of the steps required to configure a Layer 3 and Layer 4 class map and a Layer 3 and Layer 4 policy map. Each step includes the CLI command and a reference to the procedure required to complete the task. For a complete description of each feature and all the options associated with the CLI commands, see the sections following Table 3-2.

Table 3-2 Layer 3 and Layer 4 SLB Policy Configuration Quick Start

<table>
<thead>
<tr>
<th>Task and Command Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. If you are operating in multiple contexts, observe the CLI prompt to verify you are operating in the desired context. Change to, or directly log in to, the correct context if necessary.</td>
</tr>
<tr>
<td>host1/Admin# changeto C1</td>
</tr>
<tr>
<td>host1/C1#</td>
</tr>
<tr>
<td>For details on creating contexts, see the Cisco Application Control Engine Module Virtualization Configuration Guide.</td>
</tr>
<tr>
<td>2. Enter configuration mode.</td>
</tr>
<tr>
<td>host1/Admin# config</td>
</tr>
<tr>
<td>Entering configuration commands, one per line. End with CNTL/Z</td>
</tr>
<tr>
<td>host1/Admin(config)#</td>
</tr>
<tr>
<td>3. Create a Layer 3 and Layer 4 SLB class map. See the “Configuring a Layer 3 and Layer 4 Class Map for SLB” section.</td>
</tr>
<tr>
<td>host1/Admin(config)# class-map L4VIPCLASS</td>
</tr>
<tr>
<td>host1/Admin(config-cmap)#</td>
</tr>
<tr>
<td>4. Define a virtual IP (VIP) address match statement. See the “Defining VIP Address Match Criteria” section.</td>
</tr>
<tr>
<td>host1/Admin(config-cmap)# match virtual-address 192.168.1.10 tcp port eq 80</td>
</tr>
<tr>
<td>5. Reenter configuration mode.</td>
</tr>
<tr>
<td>host1/Admin(config-cmap)# exit</td>
</tr>
<tr>
<td>host1/Admin(config)#</td>
</tr>
</tbody>
</table>
6. Create a Layer 3 and Layer 4 policy map. See the “Configuring a Layer 3 and Layer 4 Policy Map for SLB” section.
   
   ```
   host1/Admin(config)# policy-map multi-match L4SLBPOLICY
   host1/Admin(config-pmap)#
   ```

7. Associate the Layer 3 and Layer 4 class map that you created in Step 2 with the policy map you created in Step 4. See the “Associating a Layer 3 and Layer 4 Class Map with a Policy Map” section.
   
   ```
   host1/Admin(config-pmap)# class L4VIPCLASS
   host1/Admin(config-pmap-c)#
   ```

8. Specify one or more of the following policy-map actions that you want the ACE to take when network traffic matches a class map:

   - Enable the ACE to advertise the IP address of a virtual server as the host route. See the “Enabling the Advertising of a Virtual Server IP Address” section.
     
     ```
     host1/Admin(config-pmap-c)# loadbalance vip advertise active
     ```

   - Enable a VIP to reply to ICMP ECHO requests. For example, if a user sends an ICMP ECHO request to a VIP, this command instructs the VIP to send an ICMP ECHO-REPLY. See the “Enabling a VIP to Reply to ICMP Requests” section.
     
     ```
     host1/Admin(config-pmap-c)# loadbalance vip icmp-reply
     ```

   - Associate a Layer 7 SLB policy map with the Layer 3 and Layer 4 policy map to provide an entry point for Layer 7 classifications. See the “Associating a Layer 7 SLB Policy Map with a Layer 3 and Layer 4 SLB Policy Map” section.
     
     ```
     host1/Admin(config-pmap-c)# loadbalance policy L7SLBPOLICY
     ```

   - Enable a VIP for SLB operations.
     
     ```
     host1/Admin(config-pmap-c)# loadbalance vip inservice
     ```
9. Activate a policy map and attach it to an interface. See the “Applying a Layer 3 and Layer 4 Policy to an Interface” section.

   host1/Admin(config)# interface VLAN50
   host1/Admin(config-if)# service-policy input L4SLBPOLICY
   host1/Admin(config-if)# Ctrl-z

10. Display your class-map and policy-map configurations and statistics (see the “Displaying Load-Balancing Configuration Information and Statistics” section).

    host1/Admin# show running-config class-map
    host1/Admin# show running-config policy-map
    host1/Admin# show service-policy name [detail]

11. (Optional) Save your configuration changes to flash memory.

    host1/Admin# copy running-config startup-config
Configuring HTTP Header Insertion, Deletion, and Rewrite

This section describes action lists and how to use them to insert, rewrite, and delete HTTP headers. An action list is a named group of actions that you associate with a Layer 7 HTTP class map in a Layer 7 HTTP policy map. You can create an action list to modify an HTTP header by using the `action-list type modify http` command in configuration mode. The syntax of this command is as follows:

```
action-list type modify http name
```

For the `name` argument, enter a unique name for the action list as an unquoted text string with a maximum of 64 alphanumeric characters.

For example, enter:

```
host1/Admin(config)# action-list type modify http HTTP_MODIFY_ACTLIST
```

To remove the action list from the configuration, enter:

```
host1/Admin(config)# no action-list type modify http HTTP_MODIFY_ACTLIST
```

**Note**
You can associate a maximum of 1024 instances of the same type of regex with a Layer 4 policy map. This limit applies to all Layer 7 policy-map types, including generic, HTTP, RADIUS, RDP, RTSP, and SIP. You configure regexes in the following:

- Match statements in Layer 7 class maps
- Inline match statements in Layer 7 policy maps
- Layer 7 hash predictors for server farms
- Layer 7 sticky expressions in sticky groups
- Header insertion and rewrite (including SSL URL rewrite) expressions in Layer 7 action lists
Chapter 3  Configuring Traffic Policies for Server Load Balancing

Configuring HTTP Header Insertion, Deletion, and Rewrite

The following sections describe the HTTP actions that you can put in an action list:

- Configuring HTTP Header Insertion
- Configuring HTTP Header Rewrite
- Configuring HTTP Header Deletion

After you create an action list and associate actions with it, you must associate the action list with a Layer 7 policy map. For details, see the “Associating an Action List with a Layer 7 Policy Map” section.

For information about rewriting an HTTP redirect URL for SSL, see the Cisco Application Control Engine Module SSL Configuration Guide.

Configuring HTTP Header Insertion

When the ACE uses Network Address Translation (NAT) to translate the source IP address of a client to a VIP, servers need a way to identify that client for the TCP and IP return traffic. To identify a client whose source IP address has been translated using NAT, you can instruct the ACE to insert a generic header and string value of your choice in the client HTTP request. (For information about NAT, see the Cisco Application Control Engine Module Security Configuration Guide.)

Note

With either TCP server reuse or persistence rebalance enabled, the ACE inserts a header in every client request. For information about TCP server reuse, see the “Configuring TCP Server Reuse” section. For information about persistence rebalance, see the “Configuring HTTP Persistence Rebalance” section.

You can insert a header name and value in an HTTP request from a client, a response from a server, or both, by using the `header insert` command in action list modify configuration mode. The syntax of this command is as follows:

```
header insert { request | response | both } header_name header-value expression
```
The keywords, options, and arguments are as follows:

- **request**—Specifies that the ACE insert an HTTP header only in HTTP request packets from clients.
- **response**—Specifies that the ACE insert an HTTP header only in HTTP response packets from servers.
- **both**—Specifies that the ACE insert an HTTP header in both HTTP request packets and response packets.
- **header_name**—Identifier of an HTTP header. Enter an unquoted text string with a maximum of 255 alphanumeric characters.
- **header-value expression**—Specifies the value of the HTTP header that you want to insert in request packets, response packets, or both. Enter an unquoted text string with no spaces and a maximum of 255 alphanumeric characters. You can also use the following dynamic replacement strings:
  - `%is`—Inserts the source IP address in the HTTP header
  - `%id`—Inserts the destination IP address in the HTTP header
  - `%ps`—Inserts the source port in the HTTP header
  - `%pd`—Inserts the destination port in the HTTP header

The ACE supports the use of regular expressions (regexes) for matching data strings. Table 3-3 lists the supported characters that you can use in regular expressions. Use parenthesized expressions for dynamic replacement using `%1` and `%2` in the replacement pattern.

> **Note** When matching data strings, note that the period (.) and question mark (?) characters do not have a literal meaning in regular expressions. Use brackets ([ ]) to match these symbols (for example, enter `www[.]xyz[.]com` instead of `www.xyz.com`). You can also use a backslash (\) to escape a dot (.) or a question mark (?).

<table>
<thead>
<tr>
<th>Table 3-3</th>
<th>Special Characters for Matching String Expressions</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Convention</strong></td>
<td><strong>Description</strong></td>
</tr>
<tr>
<td>.</td>
<td>One of any character.</td>
</tr>
<tr>
<td>.*</td>
<td>Zero or more of any character.</td>
</tr>
</tbody>
</table>
### Table 3-3  Special Characters for Matching String Expressions (continued)

<table>
<thead>
<tr>
<th>Convention</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>.</td>
<td>Period (escaped).</td>
</tr>
<tr>
<td>[charset]</td>
<td>Match any single character from the range.</td>
</tr>
<tr>
<td>[^charset]</td>
<td>Do not match any character in the range. All other characters represent themselves.</td>
</tr>
<tr>
<td>()</td>
<td>Expression grouping.</td>
</tr>
<tr>
<td>(expr1 \ expr2)</td>
<td>OR of expressions.</td>
</tr>
<tr>
<td>(expr)*</td>
<td>0 or more of expression.</td>
</tr>
<tr>
<td>(expr)+</td>
<td>1 or more of expression.</td>
</tr>
<tr>
<td>expr{m,n}</td>
<td>Repeat the expression between m and n times, where m and n have a range of 0 to 255.</td>
</tr>
<tr>
<td>expr{m}</td>
<td>Match the expression exactly m times. The range for m is from 0 to 255.</td>
</tr>
<tr>
<td>expr{m,}</td>
<td>Match the expression m or more times. The range for m is from 0 to 255.</td>
</tr>
<tr>
<td>\a</td>
<td>Alert (ASCII 7).</td>
</tr>
<tr>
<td>\b</td>
<td>Backspace (ASCII 8).</td>
</tr>
<tr>
<td>\f</td>
<td>Form-feed (ASCII 12).</td>
</tr>
<tr>
<td>\n</td>
<td>New line (ASCII 10).</td>
</tr>
<tr>
<td>\r</td>
<td>Carriage return (ASCII 13).</td>
</tr>
<tr>
<td>\t</td>
<td>Tab (ASCII 9).</td>
</tr>
<tr>
<td>\v</td>
<td>Vertical tab (ASCII 11).</td>
</tr>
<tr>
<td>\0</td>
<td>Null (ASCII 0).</td>
</tr>
<tr>
<td>\</td>
<td>Backslash.</td>
</tr>
<tr>
<td>\x##</td>
<td>Any ASCII character as specified in two-digit hexadecimal notation.</td>
</tr>
<tr>
<td>\xST</td>
<td>Stop metacharacter. For information on the use of this metacharacter, see the “Using the “\xST” Metacharacter in Regular Expressions for Layer 4 Generic Data Parsing” section.</td>
</tr>
</tbody>
</table>
For example, to insert a Host: source_ip:source_port in both the client request and the server response headers, enter:

```plaintext
host1/Admin(config)# action-list type modify http HTTP_MODIFY_ACTLIST
host1/Admin(config-actlist-mod)# header insert both Host header-value %is:%ps
```

To associate the action list with a Layer 7 load-balancing policy map, enter:

```plaintext
host1/Admin(config)# policy-map type loadbalance http first-match L7_POLICY
host1/Admin(config-pmap-lb)# class L7_CLASS
host1/Admin(config-pmap-lb-c)# serverfarm sf1
host1/Admin(config-pmap-lb-c)# action HTTP_MODIFY_ACTLIST
```

To remove the `header insert` command from the action list, enter:

```plaintext
host1/Admin(config-actlist-mod)# no header insert both Host header-value %is:%ps
```

### Configuring HTTP Header Rewrite

You can rewrite an HTTP header in request packets from a client, response packets from a server, or both, by using the `header rewrite` command in action list modify configuration mode. The syntax of this command is as follows:

```
header rewrite {request | response | both} header_name header-value
expression replace pattern
```

The keywords and arguments are as follows:

- **request**—Specifies that the ACE rewrite an HTTP header string only in HTTP request packets from clients
- **response**—Specifies that the ACE rewrite an HTTP header string only in HTTP response packets from servers
- **both**—Specifies that the ACE rewrite an HTTP header string in both HTTP request packets and response packets
- **header_name**—Identifier of an HTTP header. Enter an unquoted text string with a maximum of 255 alphanumerical characters.
• **header-value expression**—Specifies the value of the HTTP header that you want to replace in request packets, response packets, or both. Enter a text string from 1 to 255 alphanumeric characters. The ACE supports the use of regexes for matching data strings. See Table 3-3 for a list of the supported characters that you can use in regular expressions. Use parenthesized expressions for dynamic replacement using %1 and %2 in the replacement pattern.

**Note** When matching data strings, note that the period (.) and question mark (?) characters do not have a literal meaning in regular expressions. Use brackets ([ ]) to match these symbols (for example, enter `www[.]xyz[.]com` instead of `www.xyz.com`). You can also use a backslash (\) to escape a dot (.) or a question mark (?).

• **replace pattern**—Specifies the pattern string that you want to substitute for the header value regular expression. For dynamic replacement of the first and second parenthesized expressions from the header value, use %1 and %2, respectively.

For example, to replace `www.cisco.com` with `www.cisco.net`, enter:

```
host1/Admin(config)# action-list type modify http HTTP_MODIFY_ACTLIST
host1/Admin(config-actlist-mod)# header rewrite request Host header-value www\.cisco\.com replace www.%1.net
```

To associate the action list with a Layer 7 load-balancing policy map, enter:

```
host1/Admin(config)# policy-map type loadbalance http first-match L7_POLICY
host1/Admin(config-pmap-lb)# class L7_CLASS
host1/Admin(config-pmap-lb-c)# serverfarm sf1
host1/Admin(config-pmap-lb-c)# action HTTP_MODIFY_ACTLIST
```

To remove the **header rewrite** command from the action list, enter:

```
host1/Admin(config-actlist-mod)# no header rewrite request Host header-value www\.cisco\.com replace www.%1.net
```
Configuring HTTP Header Deletion

You can delete an HTTP header in a request from a client, in a response from a server, or both, by using the header delete command in action list modify configuration mode. The syntax of this command is as follows:

```
header delete {request | response | both} header_name
```

The keywords and arguments are as follows:

- **request**—Specifies that the ACE delete the header only in HTTP request packets from clients
- **response**—Specifies that the ACE delete the header only in HTTP response packets from servers
- **both**—Specifies that the ACE delete the header in both HTTP request packets and response packets
- **header_name**—Identifier of an HTTP header that you want to delete. Enter an unquoted text string with a maximum of 255 alphanumeric characters.

For example, to delete the Host header from request packets only, enter:

```
host1/Admin(config)# action-list type modify http HTTP_MODIFY_ACTLIST
host1/Admin(config-actlist-mod)# header delete request Host
```

To associate the action list with a Layer 7 load-balancing policy map, enter:

```
host1/Admin(config)# policy-map type loadbalance http first-match L7_POLICY
host1/Admin(config-pmap-lb)# class L7_CLASS
host1/Admin(config-pmap-lb-c)# serverfarm sf1
host1/Admin(config-pmap-lb-c)# action HTTP_MODIFY_ACTLIST
```

To remove the header delete command from the action list, enter:

```
host1/Admin(config-actlist-mod)# no header delete request Host
```
Configuring a Layer 7 Class Map for Generic TCP and UDP Data Parsing

You can use generic TCP and UDP data parsing to perform regular expression (regex) matches on packets from protocols that the ACE does not explicitly support. Such regex matches can be based on a custom protocol configuration. To accomplish this task, you create a Layer 7 class map for generic TCP or UDP data parsing and then instruct the ACE to perform a policy-map action based on the payload of a TCP stream or UDP packet.

To avoid using a large amount of memory with regular expressions, we recommend the following guidelines when you configure generic data parsing:

- Use only one generic rule per VIP
- Use the same offset for all generic rules on the same VIP
- Use the smallest possible offset that will work for your application
- Avoid deploying Layer 4 payload stickiness (see Chapter 5, Configuring Stickiness) and Layer 4 payload matching simultaneously, when possible

Note

The persistence-rebalance command is not compatible with generic protocol parsing.

You can create a class map for generic TCP or UDP data parsing by using the class-map type generic command in configuration mode. The syntax of this command is as follows:

```
class-map type generic match-all | match-any name
```

The keywords and arguments are as follows:

- **generic**—Specifies nonprotocol-specific behavior for data parsing
- **match-all | match-any**—(Optional) Determines how the ACE evaluates Layer 3 and Layer 4 network traffic when multiple match criteria exist in a class map. The class map is considered a match if the match commands meet one of the following conditions.
  - **match-all**—(Default) Network traffic needs to satisfy all of the match criteria (implicit AND) to match the class map.
match-any—Network traffic needs to satisfy only one of the match criteria (implicit OR) to match the load-balancing class map.

- name—Name assigned to the class map. Enter an unquoted text string with no spaces and a maximum of 64 alphanumeric characters. The name is used for both the class map and to associate the class map with a policy map.

For example, enter:

```
host1/Admin(config)# class-map type generic match-any GENERIC_L7_CLASS
```

To remove the class map from the configuration, enter:

```
host1/Admin(config)# no class-map type generic match-any GENERIC_L7_CLASS
```

After you create a class map for generic protocol parsing, configure one or more match statements as described in the following sections:

- Defining Layer 4 Payload Match Criteria for Generic Data Parsing
- Using the “\xST” Metacharacter in Regular Expressions for Layer 4 Generic Data Parsing
- Defining Source IP Address Match Criteria

## Defining Layer 4 Payload Match Criteria for Generic Data Parsing

Generic data parsing begins at Layer 4 with the TCP or UDP payload, which allows you to match Layer 5 data (in the case of Lightweight Directory Access Protocol (LDAP) or DNS or any Layer 7 header or payload (for example, HTTP). You can define match criteria for Layer 4 payloads by using the `match layer4-payload` command in class-map generic configuration mode. The syntax of this command is as follows:

```
[line_number] match layer4-payload [offset number] | [regex expression]
```

**Note**

You cannot configure more than one `match layer4-payload` command in the same `match-all` class map.
The keywords, options, and arguments are as follows:

- **line_number**—(Optional) Line numbers that you can use for editing or deleting the individual **match** commands. For example, you can enter **no line_number** to delete long **match** commands instead of entering the entire line. The sequence numbers do not indicate any priority for the match statements. Enter a unique integer from 1 to 1024.

- **offset number**—(Optional) Specifies an absolute offset in the data where the Layer 4 payload expression search string starts. The offset starts at the first byte of the TCP or UDP body. Enter an integer from 0 to 999. The default is 0.

- **regex expression**—Specifies the Layer 4 payload expression that is contained within the TCP or UDP entity body. The ACE supports the use of regexes for matching data strings. See Table 3-3 for a list of the supported characters that you can use in regular expressions. Use parenthesized expressions for dynamic replacement using %1 and %2 in the replacement pattern.

For information on the use of the “\xST” (STop) metacharacter for regular expressions, see the “Using the “\xST” Metacharacter in Regular Expressions for Layer 4 Generic Data Parsing” section.

---

**Note**

When matching data strings, note that the period (.) and question mark (?) characters do not have a literal meaning in regular expressions. Use brackets ([]) to match these symbols (for example, enter www[.]xyz[.]com instead of www.xyz.com). You can also use a backslash (\) to escape a dot (.) or a question mark (?).

---

**Note**

You can associate a maximum of 1024 instances of the same type of regex with a Layer 4 policy map. This limit applies to all Layer 7 policy-map types, including generic, HTTP, RADIUS, RDP, RTSP, and SIP. You configure regexes in the following:

- Match statements in Layer 7 class maps
- Inline match statements in Layer 7 policy maps
- Layer 7 hash predictors for server farms
- Layer 7 sticky expressions in sticky groups
- Header insertion and rewrite (including SSL URL rewrite) expressions in Layer 7 action lists
For example, to create a class map for generic Layer 4 data parsing, enter:

```
host1/Admin(config)# class-map type generic match-any GENERIC_L7_CLASS
host1/Admin(config-cmap-generic)# 10 match layer4-payload offset 500
regex abc123
```

To remove the match statement from the class map, enter:

```
host1/Admin(config-cmap-generic)# no 10
```

### Using the “\xST” Metacharacter in Regular Expressions for Layer 4 Generic Data Parsing

This section describes the use of the “\xST” metacharacter for regular expressions that are used as part of Layer 4 generic data parsing. It includes the following topics:

- **Overview**
- “\xST” Metacharacter Regex Usage Considerations
- Configuration Examples

### Overview

The ACE supports the “\xST” (STop) metacharacter for all regular expressions (regexes) in specific cases that use the maximum parse length to terminate parsing. However, the “\xST” metacharacter is specifically for use by applications that involve the generic data parsing of a Layer 4 payload.

If you intend to use the “\xST” metacharacter for regex matches on packets from protocols, we recommend that you use this metacharacter only for the following protocols in the generic data parsing of a Layer 4 payload:

- **SSL session-ID stickiness**—To perform sticky hashing on the initial packets in an SSL handshake, allowing the ACE to stick the same client to the same SSL server based on the SSL session ID.
- **Financial Information eXchange (FIX) type ‘A’ Logon message**—To define load-balancing criteria while setting up the outbound path of a connection.
The inclusion of the “\xST” metacharacter aids the ACE in properly load-balancing SSL session-ID and FIX packets. Without the “\xST” metacharacter in regexes, certain SSL session-ID and FIX packets may get stuck in the ACE HTTP engine and eventually time out the connection.

“\xST” Metacharacter Regex Usage Considerations

The “\xST” metacharacter has the following usage guidelines related to its inclusion in regex matching:

- If the input matches a regex pattern that includes the “\xST” metacharacter, the regex engine halts upon finding the character directly next to the '\xST' in the regex string (2nd '\x01' in the match statement).

- Only use the “\xST” metacharacter once in the policy. The ACE does not consider any additional input data once it sees the matching pattern, which may affect other regexes that are configured elsewhere in the policy.

- Only use the “\xST” metacharacter at the end of a regex pattern; not at the beginning. Otherwise, the ACE will display the “Error: Invalid regular expression” error message.

- Do not add the “\xST” metacharacter directly after a * wildcard match. For example, “abc.*\xST” is not a recommended regex.

Configuration Examples

The following configuration examples show the use of the “\xST” metacharacter in two very specific regexes:

**SSL Session-ID Stickiness Configuration Example**

```
parameter-map type generic SESSID-PARAM

set max-parse-length 76

sticky layer4-payload SESSID-STICKY
    serverfarm SF1
    response sticky
    layer4-payload offset 43 length 32 begin-pattern
    "(\x20|\x00)\xST"
```
**FIX Protocol Configuration Example**

```plaintext
sticky layer4-payload FIX-STICKY
serverfarm FIX-SF1
layer4-payload begin-pattern "\x0149=" end-pattern "\x01"

class-map type generic match-all FIX-CM
  2 match layer4-payload regex ".*\x0110=...\x01\xST"
```

**Defining Source IP Address Match Criteria**

You can configure the class map to filter traffic based on a client source IP address by using the `match source-address` command in class map generic configuration mode. If this command is the only match criteria in the class map, it is considered to be a Layer 3 and Layer 4 class map.

The syntax of this command is as follows:

```
[line_number] match source-address ip_address [netmask]
```

The arguments and options are as follows:

- `line_number`—(Optional) Line numbers that you can use for editing or deleting the individual `match` commands. For example, you can enter `no line_number` to delete long `match` commands instead of entering the entire line. The line numbers do not indicate any priority for the match statements. Enter a unique integer from 1 to 1024.
- `ip_address`—Source IP address of the client. Enter the IP address in dotted-decimal notation (for example, 192.168.11.2).
- `netmask`—(Optional) Subnet mask of the IP address. Enter the netmask in dotted-decimal notation (for example, 255.255.255.0). The default is 255.255.255.255.

**Note**

You cannot configure more than one `match source-address` command in the same `match-all` class map.
For example, to specify that the class map match on source IP address 192.168.11.2 255.255.255.0, enter:

```bash
host1/Admin(config)# class-map type generic match-any GENERIC_L4_CLASS
host1/Admin(config-cmap-generic)# 50 match source-address 192.168.11.2 255.255.255.0
```

To remove the source IP address match statement from the class map, enter:

```bash
host1/Admin(config-cmap-generic)# no 50
```

### Nesting Layer 7 SLB Class Maps

The nesting of class maps allows you to achieve complex logical expressions for generic parsing. You can identify one generic class map that is to be used as a matching criterion for another generic class map by using the `match class-map` command in class-map generic configuration mode.

#### Note

The ACE restricts the nesting of class maps to two levels to prevent you from including a nested class map under another class map.

The syntax of this command is as follows:

```
[line_number] match class-map map_name
```

The keywords, arguments, and options are as follows:

- **line_number**—(Optional) Line numbers that you can use for editing or deleting the individual `match` commands. For example, you can enter `no line_number` to delete long `match` commands instead of entering the entire line.

- **map_name**—Name of an existing generic class map.

The `match class-map` command allows you to combine the use of the `match-any` and `match-all` keywords in the same class map. To combine `match-all` and `match-any` characteristics in a class map, create a class map that uses one `match` command (either `match any` or `match all`), and then use this class map as a match statement in a second class map that uses a different match type.
For example, assume that commands A, B, C, and D represent separate match criteria, and you want generic protocol traffic that matches A, B, or C and D (A or B or [C and D]) to satisfy the class map. Without the use of nested class maps, traffic would either have to match all four match criteria (A and B and C and D) or match any of the match criteria (A or B or C or D) to satisfy the class map. By creating a single class map that uses the \texttt{match-all} keyword for match criteria C and D (criteria E), you can then create a new \texttt{match-any} class map that uses match criteria A, B, and E. The new traffic class contains your desired classification sequence: A or B or E, which is equivalent to A or B or [C and D].

For example, to combine the characteristics of two class maps, one with \texttt{match-any} and one with \texttt{match-all} characteristics, into a single class map by using the \texttt{match class-map} command, enter:

```
host1/Admin(config)# class-map type generic match-any GENERIC_L4_CLASS
host1/Admin(config-cmap-generic)# 50 match source-address 192.168.11.2 255.255.255.0
host1/Admin(config-cmap-generic)# exit
```

```
host1/Admin(config)# class-map type generic match-all GENERIC_L4_CLASS
host1/Admin(config-cmap-generic)# 10 match class-map GENERIC_L4_CLASS2
host1/Admin(config-cmap-generic)# 20 match source-address 192.168.11.2
host1/Admin(config-cmap-generic)# exit
```

To remove the nested class map from the generic class map, enter:

```
host1/Admin(config-cmap-generic)# no 10
```

### Configuring a Layer 7 Class Map for SLB

A Layer 7 SLB class map contains match criteria that classify specific Layer 7 network traffic. This section describes how to create a class map for Layer 7 SLB based on HTTP cookies, HTTP headers, HTTP URLs, RADIUS attributes, RDP, RTSP headers or URLs, SIP headers, or source IP addresses.

You can create a Layer 7 class map for SLB and enter the class-map configuration mode by using the \texttt{class-map type} command in configuration mode. The syntax of this command is as follows:

```
class-map type \{ [http | radius | rtsp | sip] loadbalance \} [match-all | match-any] map_name
```
You can configure multiple **match** commands in a single class map to specify the matching criteria. For example, you can configure a Layer 7 load-balancing class map to define multiple URLs, cookies, and HTTP headers in a group that you then associate with a traffic policy. The **match-all** and **match-any** keywords determine how the ACE evaluates multiple match statement operations when multiple match criteria exist in a class map.

The keywords, arguments, and options are as follows:

- **http**—Specifies a Hypertext Transfer Protocol (HTTP) load-balancing class map. This is the default.
- **radius**—Specifies the Remote Access Dial-In User Service (RADIUS) protocol for load balancing.
- **rtsp**—Specifies the Real-Time Streaming Protocol (RTSP) for load balancing.
- **sip**—Specifies the Session Initiation Protocol (SIP) for load balancing.
- **loadbalance**—Specifies a load-balancing type class map.
- **match-all | match-any**—(Optional) Determines how the ACE evaluates Layer 7 HTTP SLB operations when multiple match criteria exist in a class map. The class map is considered a match if the match commands meet one of the following conditions:
  
  - **match-all**—(Default) Network traffic needs to satisfy all of the match criteria (implicit AND) to match the Layer 7 load-balancing class map. The **match-all** keyword is applicable only for match statements of different Layer 7 load-balancing types. For example, specifying a **match-all** condition for URL, HTTP header, and URL cookie statements in the same class map is valid. However, specifying a **match-all** condition for multiple HTTP headers or multiple cookies with the same names or multiple URLs in the same class map is invalid.
  
  - **match-any**—Network traffic needs to satisfy only one of the match criteria (implicit OR) to match the HTTP load-balancing class map. The **match-any** keyword is applicable only for match statements of the same Layer 7 load-balancing type. For example, the ACE does not allow you to specify a **match-any** condition for URL, HTTP header, and URL cookie statements in the same class map but does allow you to specify a **match-any** condition for multiple URLs, multiple HTTP headers, or multiple cookies with different names in the same class map.
map_name—Unique identifier assigned to the class map. Enter an unquoted text string with no spaces and a maximum of 64 alphanumeric characters. The class-map name is used for both the class map and to associate the class map with a policy map.

For example, to create a Layer 7 load-balancing class map named L7SLBCLASS, enter:

```
host1/Admin(config)# class-map type http loadbalance match-any L7SLBCLASS
```

To remove a Layer 7 load-balancing class map from the configuration, enter:

```
host1/Admin(config)# no class-map type http loadbalance match-any L7SLBCLASS
```

The following topics describe how to specify match criteria for the Layer 7 class map:

- **Configuration Considerations**
- **Defining an HTTP Content Match for Load Balancing**
- **Defining a Cookie for HTTP Load Balancing**
- **Defining an HTTP Header for Load Balancing**
- **Defining a URL for HTTP Load Balancing**
- **Defining an Attribute for RADIUS Load Balancing**
- **Defining a Header for RTSP Load Balancing**
- **Defining a URL for RTSP Load Balancing**
- **Defining a Header for SIP Load Balancing**
- **Defining Source IP Address Match Criteria**
- **Nesting Layer 7 SLB Class Maps**
Configuration Considerations

When you are creating a class map for SLB, note the following restrictions:

- You can associate a maximum of 10 cookie names and header names with each Layer 3 and Layer 4 policy map. You can allocate the number of cookie names and header names in any combination as long as you do not exceed the maximum of 10.
- You can associate a maximum of 1024 instances of the same type of regex with each Layer 3 and Layer 4 policy map. This limit applies to all Layer 7 policy-map types, including generic, HTTP, RADIUS, RDP, RTSP, and SIP. You configure regexes in the following:
  - Match statements in Layer 7 class maps
  - Inline match statements in Layer 7 policy maps
  - Layer 7 hash predictors for server farms
  - Layer 7 sticky expressions in sticky groups
  - Header insertion and rewrite (including SSL URL rewrite) expressions in Layer 7 action lists
- The ACE restricts the nesting of class maps to two levels to prevent you from including one nested class map in a different class map.
- The maximum number of class maps for each ACE is 8192.

Defining an HTTP Content Match for Load Balancing

The ACE performs regular expression matching against the received HTTP message body from a particular connection based on a regular expression string in the message body (not the header). To configure the class map to make Layer 7 SLB decisions based on the HTTP content, use the `match http content` command in class-map configuration mode. The syntax of this command is as follows:

```
[line_number] match http content expression [offset number]
```
The arguments and options are as follows:

- `line_number`—(Optional) Line numbers that you can use for editing or deleting the individual `match` commands. For example, you can enter `no line_number` to delete long `match` commands instead of entering the entire line. The line numbers do not indicate any priority for the match statements. Enter a unique integer from 1 to 1024.

- `expression`—The regular expression content to match. Enter a string from 1 to 255 alphanumeric characters. The ACE supports the use of regular expressions for matching data strings. See Table 3-3 for a list of the supported characters that you can use in regular expressions.

  **Note** When matching data strings, note that the period (.) and question mark (?) characters do not have a literal meaning in regular expressions. Use brackets ([ ]) to match these symbols (for example, enter `www[.].xyz[.].com` instead of `www.xyz.com`). You can also use a backslash (\) to escape a dot (.) or a question mark (?).

- `offset number`—(Optional) Specifies the byte at which the ACE begins parsing the message body. Enter an integer from 0 to 999. The default is 0.

For example, enter:

```plaintext
host1/Admin(config)# class-map type http loadbalance match-any L7_HTTP_CLASS
host1/Admin(config-cmap-http-lb)# 10 match http content abc*123 offset 50
```

### Defining a Cookie for HTTP Load Balancing

The ACE performs regular expression matching against the received packet data from a particular connection based on the cookie expression. You can configure a maximum of 10 cookie names and header names per class in any combination. You can configure the class map to make Layer 7 SLB decisions based on the name and string of a cookie by using the `match http cookie` command in class-map configuration mode. The syntax of this command is as follows:

```
[line_number] match http cookie {name | secondary name} cookie-value expression
```

```plaintext
[host1/][Admin(config-cmap-http-lb)# match http cookie abc*123 offset 50
```
The keywords, arguments, and options are as follows:

- **line_number**—(Optional) Line numbers that you can use for editing or deleting the individual match commands. For example, you can enter `no line_number` to delete long match commands instead of entering the entire line. The sequence numbers do not indicate any priority for the match statements. Enter a unique integer from 1 to 1024.

- **name**—Unique cookie name. Enter an unquoted text string with no spaces and a maximum of 64 alphanumeric characters.

**Note**

If certain characters are used in the cookie name, such as an underscore (_), hyphen (-), period (.), or semicolon (:), replace those characters with the equivalent percent encoding (% HEX HEX) characters. For example, to configure the cookie name Regex_MatchCookie, replace the underscore (_) character with the equivalent %5F percent encoding character and enter the cookie name as `Regex%5FMatchCookie` in the CLI.

- **secondary name**—Specifies a cookie in a URL string. You can specify the delimiters for cookies in a URL string using a command in an HTTP parameter map. For more information, see the “Defining URL Delimiters” section.

- **cookie-value expression**—Specifies a unique cookie value regular expression. Enter an unquoted text string with no spaces and a maximum of 255 alphanumeric characters. Alternatively, you can enter a text string with spaces provided that you enclose the entire string in quotation marks ("). The ACE supports the use of regular expressions for matching string expressions. See Table 3-3 for a list of the supported characters that you can use for matching string expressions.

**Note**

When matching data strings, note that the period (.) and question mark (?) characters do not have a literal meaning in regular expressions. Use brackets ([ ]) to match these symbols (for example, enter `www[.]xyz[.]com` instead of `www.xyz.com`). You can also use a backslash (\) to escape a dot (.) or a question mark (?).
For example, to specify that the Layer 7 class map load balance on a cookie with the name of testcookie1, enter:

```
host1/Admin(config)# class-map type http loadbalance match-any
L7SLBCLASS
host1/Admin(config-cmap-http-lb)# 100 match http cookie testcookie1
cookie-value 123456
```

To remove an HTTP cookie match statement from the class map, enter:

```
host1/Admin(config-cmap-http-lb)# no 100
```

### Defining an HTTP Header for Load Balancing

The ACE performs regular expression matching against the received packet data from a particular connection based on the HTTP header expression. You can configure a maximum of 10 HTTP header names and cookie names per class in any combination. To configure a class map to make Layer 7 SLB decisions based on the name and value of an HTTP header, use the `match http header` command in class-map HTTP load balance configuration mode.

The syntax of this command is as follows:

```
[line_number] match http header name header-value expression
```

The keywords, arguments, and options are as follows:

- **line_number**—(Optional) Line numbers that you can use for editing or deleting the individual `match` commands. For example, you can enter `no line_number` to delete long `match` commands instead of entering the entire line. The sequence numbers do not indicate any priority for the match statements. Enter a unique integer from 1 to 1024.

- **name**—Name of the field in the HTTP header. Enter an unquoted text string with no spaces and a maximum of 64 alphanumeric characters. Alternatively, you can enter a text string with spaces if you enclose the entire string in quotation marks ("”). You can enter any header field name, including a standard HTTP header field name or any user-defined header field name. For a list of standard HTTP header field names, see **Table 3-4**.
- **header-value expression**—Specifies the header value regular expression string to compare against the value in the specified field in the HTTP header. Enter a text string with a maximum of 255 alphanumeric characters. The ACE supports the use of regular expressions for header matching. Expressions are stored in a header map in the form `header-name: expression`. Header expressions allow spaces, provided that the entire string that contains spaces is quoted. If you use a **match-all** class map, all headers in the header map must be matched. See Table 3-3 for a list of the supported characters that you can use in regular expressions.

**Note**
When matching data strings, note that the period (.) and question mark (?) characters do not have a literal meaning in regular expressions. Use brackets ([[]]) to match these symbols (for example, enter `www[.]xyz[.]com` instead of `www.xyz.com`). You can also use a backslash (\) to escape a dot (.) or a question mark (?).

### Table 3-4 Standard HTTP Header Fields

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accept</td>
<td>Semicolon-separated list of representation schemes (content type metainformation values) that will be accepted in the response to the request.</td>
</tr>
<tr>
<td>Accept-Charset</td>
<td>Character sets that are acceptable for the response. This field allows clients capable of understanding more comprehensive or special-purpose character sets to signal that capability to a server that can represent documents in those character sets.</td>
</tr>
<tr>
<td>Accept-Encoding</td>
<td>Restricts the content encoding that a user will accept from the server.</td>
</tr>
<tr>
<td>Accept-Language</td>
<td>ISO code for the language in which the document is written. The language code is an ISO 3316 language code with an optional ISO639 country code to specify a national variant.</td>
</tr>
<tr>
<td>Authorization</td>
<td>Specifies that the user agent wants to authenticate itself with a server, usually after receiving a 401 response.</td>
</tr>
</tbody>
</table>
### Table 3-4 Standard HTTP Header Fields (continued)

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cache-Control</td>
<td>Directives that must be obeyed by all caching mechanisms along the request/response chain. The directives specify behavior intended to prevent caches from adversely interfering with the request or response.</td>
</tr>
<tr>
<td>Connection</td>
<td>Allows the sender to specify connection options.</td>
</tr>
<tr>
<td>Content-MD5</td>
<td>MD5 digest of the entity-body that provides an end-to-end integrity check. Only a client or an origin server can generate this header field.</td>
</tr>
<tr>
<td>Expect</td>
<td>Used by a client to inform the server about what behaviors the client requires.</td>
</tr>
<tr>
<td>From</td>
<td>E-mail address of the person that controls the requesting user agent.</td>
</tr>
<tr>
<td>Host</td>
<td>Internet host and port number of the resource being requested, as obtained from the original URI given by the user or referring resource. The Host field value must represent the naming authority of the origin server or gateway given by the original URL.</td>
</tr>
<tr>
<td>If-Match</td>
<td>Used with a method to make it conditional. A client that has one or more entities previously obtained from the resource can verify that one of those entities is current by including a list of their associated entity tags in the If-Match header field. This feature allows efficient updates of cached information with a minimum amount of transaction overhead. It is also used on updating requests to prevent inadvertent modification of the wrong version of a resource. As a special case, the value “*” matches any current entity of the resource.</td>
</tr>
</tbody>
</table>
Table 3-4  Standard HTTP Header Fields (continued)

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pragma</td>
<td>Pragma directives understood by servers to whom the directives are relevant. The syntax is the same as for other multiple-value fields in HTTP, for example, the <code>accept</code> field, a comma-separated list of entries, for which the optional parameters are separated by semicolons.</td>
</tr>
<tr>
<td>Referer</td>
<td>Address (URI) of the resource from which the URI in the request was obtained.</td>
</tr>
<tr>
<td>Transfer-Encoding</td>
<td>What (if any) type of transformation has been applied to the message body in order to safely transfer it between the sender and the recipient.</td>
</tr>
<tr>
<td>User-Agent</td>
<td>Information about the user agent, for example, a software program originating the request. This information is for statistical purposes, the tracing of protocol violations, and automated recognition of user agents to customize responses to avoid particular user agent limitations.</td>
</tr>
<tr>
<td>Via</td>
<td>Used by gateways and proxies to indicate the intermediate protocols and recipients between the user agent and the server on requests and between the origin server and the client on responses.</td>
</tr>
</tbody>
</table>

For example, to specify that the Layer 7 class map load balance on an HTTP header named Host, enter:

```
host1/Admin(config)# class-map type http loadbalance match-any L7SLBCLASS
host1/Admin(config-cmap-http-lb)# 100 match http header Host header-value .*cisco.com
```
For example, to use regular expressions in a class map to emulate a wildcard search to match the header value expression string, enter:

```
host1/Admin(config)# class-map type http loadbalance match-any L7SLBCLASS
host1/Admin(config-cmap-http-lb)# 10 match http header Host header-value .*cisco.com
host1/Admin(config-cmap-http-lb)# 20 match http header Host header-value .*yahoo.com
```

For example, to specify that the Layer 7 class map load balance on an HTTP header named Via, enter:

```
host1/Admin(config)# class-map type http loadbalance match-any L7SLBCLASS
host1/Admin(config-cmap-http-lb)# 200 match http header Via header-value 192.*
```

To remove HTTP header match criteria from the L7SLBCLASS class map, enter:

```
host1/Admin(config-cmap-http-lb)# no 10
host1/Admin(config-cmap-http-lb)# no 20
```

**Defining a URL for HTTP Load Balancing**

The ACE performs regular expression matching against the received packet data from a particular connection based on the HTTP URL string. To configure a class map to make Layer 7 SLB decisions based on the URL name and, optionally, the HTTP method, use the `match http url` command in class-map HTTP load balance configuration mode. The syntax of this command is as follows:

```
[line_number] match http url expression [method name]
```

The keywords, arguments, and options are as follows:

- **line_number**—(Optional) Line numbers that you can use for editing or deleting the individual `match` commands. For example, you can enter `no line_number` to delete long `match` commands instead of entering the entire line. The `line_number` line numbers do not indicate any priority for the match statements. Enter a unique integer from 1 to 1024.
- **expression**—URL, or portion of a URL, to match. Enter a URL string from 1 to 255 alphanumeric characters. The ACE performs matching on whatever URL string appears after the HTTP method, regardless of whether the URL includes the hostname. The ACE supports the use of regular expressions for matching URL strings. See Table 3-3 for a list of the supported characters that you can use in regular expressions.

**Note** When matching URLs, note that the period (.) and question mark (?) characters do not have a literal meaning in regular expressions. Use brackets ([ ]) to match these symbols (for example, enter www.[.]xyz[.]com instead of www.xyz.com). You can also use a backslash (\) to escape a dot (.) or a question mark (?).

- **method name**—(Optional) Specifies the HTTP method to match. Enter a method name as an unquoted text string with no spaces and a maximum of 64 alphanumeric characters. The method can either be one of the standard HTTP 1.1 method names (OPTIONS, GET, HEAD, POST, PUT, DELETE, TRACE, or CONNECT) or a text string that must be matched exactly (for example, CORVETTE).

We recommend that you use the “/.*” regex to match HTTP URLs. If you use the “.*” regex only, the ACE may pass requests that do not conform to RFC syntax (see RFC 2396). Web servers typically respond to such invalid requests with a 400 error. An example of an invalid request is “GET index.html HTTP/1.1”, whereas “GET /index.html HTTP/1.1” is valid. Using the “/.*” regex will match all valid URLs that do not have a host name in the URI, which is rare.

Configuring the ”/.*” regex does exclude matches for URIs that begin with “http://”, which, in some scenarios, may not be desirable. However, such requests are not expected to be seen in a production environment because only some web proxies exhibit this behavior and not clients. To match generic requests with a hostname in the URI, include a statement such as `match http url http://.*` with the `match http url /.*` statement in a match-any type class map. This combination of regular expressions in different match statements will rule out a URI that does not include a preceding “/” (forward slash) character.
Using the configured **match http url expression** command, the ACE attempts to match a URL in an HTTP request following the first space after the request method. For example, if the request were “GET /index.html HTTP/1.1”, the ACE tries to match starting with the “/” character. Therefore, the class-map match statement is conformant with the RFCs that cover HTTP URI syntax. This syntax includes request URIs that start either with “/” or with "scheme://authority/” (for example, http://www.cisco.com/).

To specify that the Layer 7 class map load balance on a specific URL, enter:

```
host1/Admin(config)# class-map type http loadbalance L7SLBCLASS
host1/Admin(config-cmap-http-lb)# 10 match http url /whatsnew/latest.*
```

To use regular expressions to emulate a wildcard search to match on any .gif or .html file, enter:

```
host1/Admin(config)# class-map type http loadbalance match-any L7SLBCLASS
host1/Admin(config-cmap-http-lb)# 100 match http url .*\..*\.(gif|html)
```

To remove a URL match statement from the L7SLBCLASS class map, enter `no` and the line number. For example, to remove line 100, enter:

```
host1/Admin(config-cmap-http-lb)# no 100
```

**Note**
If you did not use line numbers to enter the original URL match statement, you can obtain the line number from your running configuration. To display the running configuration, enter `show running-config`.

## Defining an Attribute for RADIUS Load Balancing

The ACE performs Layer 7 RADIUS load balancing based on the calling-station-ID or username RADIUS attributes. After you configure a Layer 7 class map (see the “Configuring a Layer 7 Class Map for SLB” section), you can specify Layer 7 RADIUS match criteria by using the **match radius attribute** command in class map RADIUS load balance configuration mode. The syntax of this command is as follows:

```
match radius attribute {calling-station-id | username} expression
```
The keywords and arguments are as follows:

- **calling-station-id**—Specifies the unique identifier of the calling station.
- **username**—Specifies the name of the RADIUS user who initiated the connection.
- **expression**—The calling station ID or username to match. Enter a string from 1 to 64 alphanumeric characters. The ACE supports the use of regular expressions for matching strings. See Table 3-3 for a list of the supported characters that you can use in regular expressions.

---

**Note**

A match-all class map cannot have more than one same type match, while a match-any class map cannot have more than one different type match.

For example, to configure RADIUS match criteria based on the calling station ID, enter:

```
host1/Admin(config)# class-map type radius loadbalance match-any RADIUS_L7_CLASS
host1/Admin(config-cmap-radius-lb)# 10 match radius attribute calling-station-id 122*
```

To remove the RADIUS attribute match statement from the RADIUS_L7_CLASS class map, enter `no` and the line number. For example, to remove line 10, enter:

```
host1/Admin(config-cmap-radius-lb)# no 10
```

---

## Defining a Header for RTSP Load Balancing

The ACE performs regular expression matching against the received packet data from a particular connection based on the RTSP header expression. You can configure a maximum of 10 RTSP header names per class.

---

**Note**

When the ACE receives an RTSP session request, the load-balancing decision is based on the first request message. All subsequent request and response message exchanges are forwarded to the same server. When you configure header match criteria, ensure that the header is included in the first request message by a media player.
You can configure a class map to make Layer 7 SLB decisions based on the name and value of an RTSP header by using the `match rtsp header` command in class-map RTSP load balance configuration mode.

The syntax of this command is as follows:

```
[line_number] match rtsp header name header-value expression
```

The keywords, arguments, and options are as follows:

- `line_number`—(Optional) Line numbers that you can use for editing or deleting the individual `match` commands. For example, you can enter `no line_number` to delete long `match` commands instead of entering the entire line. The sequence numbers do not indicate any priority for the match statements. Enter a unique integer from 1 to 1024.

- `name`—Name of the field in the RTSP header. Enter an unquoted text string with no spaces and a maximum of 64 alphanumeric characters. Alternatively, you can enter a text string with spaces if you enclose the entire string in quotation marks ("). You can enter any header field name, including a standard RTSP header field name or any user-defined header field name.

```
Note
RTSP is intentionally similar in syntax and operation to HTTP/1.1, so you can use any HTTP header defined in Table 3-4 if the RTSP server supports it. For a complete list of RTSP headers, see RFC 2326.
```

- `header-value expression`—Specifies the header value expression string to compare against the value in the specified field in the RTSP header. Enter a text string with a maximum of 255 alphanumeric characters. The ACE supports the use of regular expressions for header matching. Expressions are stored in a header map in the form `header-name: expression`. Header expressions allow spaces if the entire string that contains spaces is quoted. If you use a `match-all` class map, all headers in the header map must be matched. See Table 3-3 for a list of the supported characters that you can use in regular expressions.

For example, to specify that the Layer 7 class map load balance on an RTSP header named Session, enter:

```
host1/Admin(config)# class-map type rtsp loadbalance match-any L7SLBCLASS
host1/Admin(config-cmap-rtsp-lb)# 100 match rtsp header Session header-value abc123
```
To use regular expressions in a class map to emulate a wildcard search to match the header value expression string, enter:

```
host1/Admin(config)# class-map type rtsp loadbalance match-any  
    L7SLBCLASS
host1/Admin(config-cmap-rtsp-lb)# 10 match rtsp header Require header-value feature1
host1/Admin(config-cmap-rtsp-lb)# 20 match rtsp header Require header-value feature2
```

To specify that the Layer 7 class map load balance on an RTSP header named Via, enter:

```
host1/Admin(config)# class-map type rtsp loadbalance match-any  
    L7SLBCLASS
host1/Admin(config-cmap-rtsp-lb)# 30 match rtsp header Via header-value 192.*
```

To remove all RTSP header match criteria from the L7SLBCLASS class map, enter:

```
host1/Admin(config-cmap-rtsp-lb)# no 10
host1/Admin(config-cmap-rtsp-lb)# no 20
host1/Admin(config-cmap-rtsp-lb)# no 30
```

### Defining a URL for RTSP Load Balancing

The ACE performs regular expression matching against the received packet data from a particular connection based on the RTSP URL string. You can configure a class map to make Layer 7 SLB decisions based on the URL name and optionally, the RTSP method, by using the `match rtsp url` command in class-map RTSP load balance configuration mode. The syntax of this command is as follows:

```
[line_number] match rtsp url expression
```

The keywords, arguments, and options are as follows:

- `line_number`—(Optional) Line numbers that you can use for editing or deleting the individual `match` commands. For example, you can enter `no line_number` to delete long `match` commands instead of entering the entire line. The `line_number` line numbers do not indicate any priority for the match statements. Enter a unique integer from 1 to 1024.
expression—URL, or portion of a URL, to match. Enter a URL string from 1 to 255 alphanumeric characters. The ACE performs matching on whatever URL string appears after the RTSP method, regardless of whether the URL includes the hostname. The ACE supports the use of regular expressions for matching URL strings. See Table 3-3 for a list of the supported characters that you can use in regular expressions.

Note When matching URLs, note that the period (.) and question mark (?) characters do not have a literal meaning in regular expressions. Use brackets ([ ]) to match these symbols (for example, enter www.[.]xyz[.]com instead of www.xyz.com). You can also use a backslash (\) to escape a dot (.) or a question mark (?).

To specify that the Layer 7 class map load balance on a specific URL, enter:

```
host1/Admin(config)# class-map type rtsp loadbalance L7SLBCLASS
host1/Admin(config-cmap-rtsp-lb)# 10 match rtsp url /whatsnew/latest.*
```

To use regular expressions to emulate a wildcard search to match on any .wav or .mpg file, enter:

```
host1/Admin(config)# class-map type rtsp loadbalance match-any L7SLBCLASS
host1/Admin(config-cmap-rtsp-lb)# 100 match rtsp url .*\.wmv
host1/Admin(config-cmap-rtsp-lb)# 200 match rtsp url .*\.mpg
```

To remove a URL match statement from the L7SLBCLASS class map, enter no and the line number. For example, to remove line 100, enter:

```
host1/Admin(config-cmap-rtsp-lb)# no 100
```

Note If you did not use line numbers to enter the original URL match statement, you can obtain the line number from your running configuration. To display the running configuration, enter show running-config.
Defining a Header for SIP Load Balancing

The ACE performs regular expression matching against the received packet data from a particular connection based on the SIP header expression. You can configure a maximum of nine SIP header field names per class (the ACE always parses Call-ID).

Note

When the ACE receives a SIP session, the load-balance decision is based on the first request message. All subsequent request and response message exchanges (with the same Call-ID) are forwarded to the same server. As a result, when configuring header match criteria, ensure that the header is included in the first request message.

You can configure a class map to make Layer 7 SLB decisions based on the name and value of a SIP header by using the `match sip header` command in class-map SIP load balance configuration mode.

The syntax of this command is as follows:

```
[line_number] match sip header name header-value expression
```

The keywords, arguments, and options are as follows:

- `line_number`—(Optional) Line numbers that you can use for editing or deleting the individual `match` commands. For example, you can enter `no line_number` to delete long `match` commands instead of entering the entire line. The sequence numbers do not indicate any priority for the match statements. Enter a unique integer from 1 to 1024.

- `name`—Name of the field in the SIP header. Enter an unquoted text string with no spaces and a maximum of 64 alphanumeric characters. Alternatively, you can enter a text string with spaces if you enclose the entire string in quotation marks (" "). You can enter any header field name, including a standard SIP header field name or any user-defined header field name. For a list of standard SIP header field names, see Table 3-5.

Note

SIP is similar to HTTP, so you can use any HTTP header defined in Table 3-4 if the SIP server supports it. For a complete list of SIP headers, see RFC 3261.
header-value expression—Specifies the header value expression string to compare against the value in the specified field in the SIP header. Enter a text string with a maximum of 255 alphanumeric characters. The ACE supports the use of regular expressions for header matching. Expressions are stored in a header map in the form `header-name: expression`. Header expressions allow spaces if the entire string that contains the spaces is quoted. If you use a match-all class map, all headers in the header map must be matched. See Table 3-3 for a list of the supported characters that you can use in regular expressions.

**Table 3-5 Standard SIP Header Fields**

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Call-ID</td>
<td>Unique identifier that groups a series of messages in a call.</td>
</tr>
<tr>
<td>Contact</td>
<td>SIP URI that can be used to contact the user agent.</td>
</tr>
<tr>
<td>From</td>
<td>Initiator of the SIP request, the source.</td>
</tr>
<tr>
<td>To</td>
<td>Desired recipient of the SIP request; the destination.</td>
</tr>
<tr>
<td>Via</td>
<td>Transport used for the transaction and where the response should be sent.</td>
</tr>
</tbody>
</table>

For example, to specify that the Layer 7 class map load balance on an SIP header named Session, enter:

```
host1/Admin(config)# class-map type sip loadbalance match-any L7SLBCLASS
host1/Admin(config-cmap-sip-lb)# 100 match sip header Session header-value abc123
```

To use regular expressions in a class map to emulate a wildcard search to match the header value expression string, enter:

```
host1/Admin(config)# class-map type sip loadbalance match-any L7SLBCLASS
host1/Admin(config-cmap-sip-lb)# 10 match sip header To header-value .*@cisco.com
host1/Admin(config-cmap-sip-lb)# 20 match sip header To header-value .*@linksys.com
```
To specify that the Layer 7 class map load balance on an SIP header named Via, enter:

```
host1/Admin(config)# class-map type sip loadbalance match-any L7SLBCLASS
host1/Admin(config-cmap-sip-lb)# 30 match sip header Via header-value 192.*
```

To remove all SIP header match criteria from the L7SLBCLASS class map, enter:

```
host1/Admin(config-cmap-sip-lb)# no 10
host1/Admin(config-cmap-sip-lb)# no 20
host1/Admin(config-cmap-sip-lb)# no 30
```

### Defining Source IP Address Match Criteria

You can configure the class map to make Layer 7 SLB decisions based on a client source IP address by using the `match source-address` command in class-map load balance configuration mode. If this command is the only match criteria in the class map, the ACE considers it to be a Layer 3 and Layer 4 class map.

The syntax of this command is as follows:

```
[line_number] match source-address ip_address [netmask]
```

The arguments and options are as follows:

- `line_number`—(Optional) Line numbers that you can use for editing or deleting the individual `match` commands. For example, you can enter `no line_number` to delete long `match` commands instead of entering the entire line. The line numbers do not indicate any priority for the match statements. Enter a unique integer from 1 to 1024.

- `ip_address`—Source IP address of the client. Enter the IP address in dotted-decimal notation (for example, 192.168.11.2).

- `netmask`—(Optional) Subnet mask of the IP address. Enter the netmask in dotted-decimal notation (for example, 255.255.255.0). The default is 255.255.255.255.
For best results, do not configure multiple class maps with overlapping subnets in the `match source-address` statements. For example:

```
class-map type http loadbalance match-any LB_CLASS
  2 match source-address 192.168.40.0 255.255.255.0
  3 match source-address 192.168.41.0 255.255.255.0

class-map type http loadbalance match-all WIDE_SUBNET_CLASS
  2 match source-address 192.168.0.0 255.255.0.0
```

```
policy-map type loadbalance http first-match HTTP_POLICY
  class LB_CLASS
    drop
  class WIDE_SUBNET_CLASS
    sticky-serverfarm SF2
```

In this case, the ACE may not match the client source addresses properly. To ensure proper operation of the ACE, configure individual source IP addresses in one of the class maps as a workaround.

For example, to apply the workaround to the WIDE_SUBNET_CLASS class map, configure the following IP addresses:

```
class-map type http loadbalance match-any WIDE_SUBNET
  2 match source-address 192.168.40.0 255.255.255.0
  3 match source-address 192.168.41.0 255.255.255.0
  4 match source-address 192.168.0.0 255.255.0.0
```

For example, to specify that the class map match on source IP address 192.168.11.2 255.255.255.0, enter:

```
host1/Admin(config)# class-map http type loadbalance match-any L7SLBCLASS
host1/Admin(config-cmap-http-lb)# 50 match source-address 192.168.11.2 255.255.255.0
```

To remove the source IP address match statement from the class map, enter:

```
host1/Admin(config-cmap-http-lb)# no 50
```
Nesting Layer 7 SLB Class Maps

The nesting of class maps allows you to achieve complex logical expressions for Layer 7 SLB. You can identify one Layer 7 SLB class map that is to be used as a matching criterion for another Layer 7 class map by using the `match class-map` command in class-map load balance configuration mode.

**Note**
The ACE restricts the nesting of class maps to two levels to prevent you from including a nested class map under another class map.

The syntax of this command is as follows:

```
[line_number] match class-map map_name
```

The keywords, arguments, and options are as follows:

- **line_number**—(Optional) Line numbers that you can use for editing or deleting the individual `match` commands. For example, you can enter `no line_number` to delete long `match` commands instead of entering the entire line.

- **map_name**—Name of an existing Layer 7 load-balancing class map.

The `match class-map` command allows you to combine the use of the `match-any` and `match-all` keywords in the same class map. To combine `match-all` and `match-any` characteristics in a class map, create a class map that uses one `match` command (either `match any` or `match all`), and then use this class map as a match statement in a second class map that uses a different match type.

For example, assume that commands A, B, C, and D represent separate match criteria, and you want Layer 7 traffic that matches A, B, or C and D (A or B or [C and D]) to satisfy the class map. Without the use of nested class maps, traffic would either have to match all four match criteria (A and B and C and D) or match any of the match criteria (A or B or C or D) to satisfy the class map. By creating a single class map that uses the `match-all` keyword for match criteria C and D (criteria E), you can then create a new `match-any` class map that uses match criteria A, B, and E. The new traffic class contains your desired classification sequence: A or B or E, which is equivalent to A or B or [C and D].
For example, to combine the characteristics of two class maps, one with match-any and one with match-all characteristics, into a single class map by using the match class-map command, enter:

```
host1/Admin(config)# class-map type http loadbalance match-any CLASS3
host1/Admin(config-cmap-http-lb)# 100 match http url .*.gif
host1/Admin(config-cmap-http-lb)# 200 match http url .*.html
host1/Admin(config-cmap-http-lb)# exit
```

```
host1/Admin(config)# class-map type http loadbalance match-all CLASS4
host1/Admin(config-cmap-http-lb)# 10 match class-map CLASS3
host1/Admin(config-cmap-http-lb)# 20 match source-address 192.168.11.2
host1/Admin(config-cmap-http-lb)# exit
```

To remove the nested class map from the Layer 7 class map, enter:

```
host1/Admin(config-cmap-http-lb)# no 10
```

### Configuring a Layer 7 Policy Map for SLB

To use a Layer 7 SLB policy map, first create the policy map and define match statements and policy actions. Because Layer 7 policy maps are child policies, you must then associate a Layer 7 policy map with the appropriate Layer 3 and Layer 4 policy map to provide an entry point for Layer 7 SLB traffic classification. You cannot directly apply a Layer 7 policy map on an interface; you can activate only a Layer 3 and Layer 4 policy map on an interface or globally on all interfaces in a context.

For background information about the role of policy maps in the ACE, see the *Cisco Application Control Engine Module Administration Guide*.

**Note**

The ACE treats as a Layer 3 and Layer 4 policy any policy map that has only source IP configured as the match criteria in the class map or inline match (except SIP LB) or the default class configured as the class map, and there are no configured Layer 7 policy actions.

You can create a Layer 7 SLB policy map and enter policy-map configuration mode by using the `policy-map type` command in configuration mode. The syntax of this command is as follows:

```
policy-map type loadbalance [generic | http | radius | rdp | rtsp | sip]
    first-match map_name
```
The keywords and arguments are as follows:

- **loadbalance**—Specifies a policy map that defines Layer 7 SLB decisions.
- **generic**—(Optional) Specifies a generic protocol policy map for load balancing. Use this keyword to provide support for protocols that the ACE does not explicitly support. If you do not configure Layer 4 payload match criteria (see the “Defining Layer 4 Payload Match Criteria for Generic Data Parsing” section) or the UDP fast-age feature (see the “Enabling Per-Packet Load Balancing for UDP Traffic” section), the ACE treats the generic policy as a Layer 3 and Layer 4 policy.

Note: The **persistence-rebalance** command is not compatible with generic protocol parsing.

When configuring the generic protocol policy map, you can also enable per-packet load balancing on UDP traffic, also known as the UDP fast-age feature. For more information on this feature, see the “Enabling Per-Packet Load Balancing for UDP Traffic” section.

- **http**—(Optional) Specifies the Hypertext Transfer Protocol (HTTP) for load balancing. This is the default.
- **radius**—(Optional) Specifies the Remote Authentication Dial-In User Service (RADIUS) for load balancing.
- **rdp**—(Optional) Specifies the Microsoft Remote Desktop Protocol (RDP) for load balancing.
- **rtsp**—(Optional) Specifies the Real-Time Streaming Protocol (RTSP) for load balancing.
- **sip**—(Optional) Specifies the Session Initiation Protocol (SIP) for load balancing.
- **first-match**—Defines the execution for the Layer 7 load-balancing policy map. The ACE executes only the action specified against the first-matching classification.
- **map_name**—Identifier assigned to the policy map. Enter an unquoted text string with no spaces and a maximum of 64 alphanumeric characters.
For example, to create a Layer 7 policy map for SIP load balancing, enter:

```
host1/Admin(config)# policy-map type loadbalance sip first-match
SIP_L7_POLICY
host1/Admin(config-pmap-lb-sip)#
```

To remove a policy map from the ACE, enter:

```
host1/Admin(config)# no policy-map type loadbalance sip first-match
SIP_L7_POLICY
```

This section contains the following topics that describe how to use sequence numbers, define inline match statements, and define policy-map actions:

- Adding a Layer 7 Policy Map Description
- Defining Inline Match Statements in a Layer 7 Policy Map
- Associating a Layer 7 Class Map with a Layer 7 Policy Map
- Specifying Layer 7 SLB Policy Actions
- Associating a Layer 7 Policy Map with a Layer 3 and Layer 4 Policy Map

### Adding a Layer 7 Policy Map Description

You can use the `description` command to provide a brief summary about the Layer 7 policy map.

You must access the policy map configuration mode to specify the `description` command.

The syntax of this command is as follows:

```
description text
```

Use the `text` argument to enter an unquoted text string with a maximum of 240 alphanumeric characters.

For example, to add a description that the policy map is to insert HTTP headers, enter:

```
host1/Admin(config-pmap-lb)# description insert HTTP headers
```

To remove the description from the policy map, enter:

```
host1/Admin(config-pmap-lb)# no description
```
Defining Inline Match Statements in a Layer 7 Policy Map

Layer 7 SLB policy maps allow you to enter a single inline SLB match criteria in the policy map without specifying a traffic class. The inline Layer 7 SLB policy map `match` commands function similarly to the Layer 7 SLB class map `match` commands. However, when you use an inline `match` command, you can specify an action for only a single match statement in the Layer 7 policy.

Note

To specify actions for multiple match statements, use a class map as described in the “Configuring a Layer 7 Class Map for Generic TCP and UDP Data Parsing” and “Configuring a Layer 7 Class Map for SLB” sections.

The syntax for an inline `match` command is as follows:

```
match name1 match_statement [insert-before name2]
```

The arguments and options are as follows:

- `name1`—Name assigned to the inline `match` command. Enter an unquoted text string with no spaces. The length of the inline match statement name plus the length of the policy map name with which it is associated cannot exceed a total maximum of 64 alphanumeric characters. For example, if the policy map name is `L7_POLICY` (nine characters), an inline match statement name under this policy cannot exceed 55 alphanumeric characters (64 - 9 = 55).

- `match_statement`—Individual Layer 7 SLB match criteria.

- `insert-before name2`—(Optional) Places the current match statement ahead of an existing class map or other match statement specified by the `name2` argument in the policy-map configuration. The ACE does not save the sequence reordering as part of the configuration.
You can associate a maximum of 1024 instances of the same type of regex with a Layer 4 policy map. This limit applies to all Layer 7 policy-map types, including generic, HTTP, RADIUS, RDP, RTSP, and SIP. You configure regexes in the following:

- Match statements in Layer 7 class maps
- Inline match statements in Layer 7 policy maps
- Layer 7 hash predictors for server farms
- Layer 7 sticky expressions in sticky groups
- Header insertion and rewrite (including SSL URL rewrite) expressions in Layer 7 action lists

For information about the inline match statements that you can configure in a Layer 7 SLB policy map, see the “Configuring a Layer 7 Class Map for Generic TCP and UDP Data Parsing” and the “Configuring a Layer 7 Class Map for SLB” sections.

The `line_number` argument described in the above-referenced sections is only for use with match statements in class maps. Otherwise, the descriptions of match statements in Layer 7 class maps and inline match statements in Layer 7 policy maps are the same.

### Associating a Layer 7 Class Map with a Layer 7 Policy Map

You can associate an existing Layer 7 class map with a Layer 7 policy map by using the `class` command. The syntax of this command is as follows:

```
class {name1 [insert-before name2] | class-default}
```

The keywords, arguments, and options are as follows:

- `name1`—Name of a previously defined traffic class, configured with the `class-map` command, to associate traffic to the traffic policy. Enter an unquoted text string with no spaces and a maximum of 64 alphanumeric characters.
- **insert-before** name2—(Optional) Places the current class map ahead of an existing class map or match statement specified by the name2 argument in the policy-map configuration. The ACE does not save the sequence reordering as part of the configuration.

- **class-default**—Specifies the reserved, well-known class map created by the ACE. You cannot delete or modify this class. All traffic that fails to meet the other matching criteria in the named class map belongs to the default traffic class. If none of the specified classifications match the traffic, then the ACE performs the action specified under the **class class-default** command. The **class-default** class map has an implicit match any statement in it that enables it to match all traffic.

For example, to use the **insert-before** option to define the position of a class map in the policy map, enter:

```
host1/Admin(config-pmap-lb)# class L7SLBCLASS insert-before http_class
host1/Admin(config-pmap-lb-c)#
```

To remove a class map from a Layer 7 policy map, enter:

```
host1/Admin(config-pmap-lb)# no class L7SLBCLASS
```

The following example shows the use of the **class class-default** command:

```
host1/Admin(config-pmap-lb)# class L7SLBCLASS insert-before http_class
host1/Admin(config-pmap-lb-c)# exit
host1/Admin(config-pmap-lb)# class class-default
host1/Admin(config-pmap-lb-c)#
```

### Specifying Layer 7 SLB Policy Actions

After you associate a Layer 7 SLB class map with a Layer 7 SLB policy map or specify inline **match** commands, you must specify the actions that the ACE should take when network traffic matches a class map or inline **match** command. You can specify the Layer 7 SLB policy actions by using the commands described in the following topics:

- **Associating an Action List with a Layer 7 Policy Map**
- **Discarding Requests**
- **Forwarding Requests Without Load Balancing**
- **Configuring HTTP Header Insertion**
- **Enabling Load Balancing to a Server Farm**
- Configuring a Sticky Server Farm
- Specifying the IP Differentiated Services Code Point of Packets
- Specifying an SSL Proxy Service

**Associating an Action List with a Layer 7 Policy Map**

You use action lists to group several ACE actions (for example, HTTP header insert, rewrite, or delete) together in a named list under a Layer 7 policy map. For details about configuring an action list, see the “Configuring HTTP Header Insertion, Deletion, and Rewrite” section.

You can associate an action list with a Layer 7 policy map by using the `action` command in policy map load-balancing class configuration mode. The syntax of this command is as follows:

```
    action name
```

The `name` argument is the identifier of an existing action list. Enter an unquoted text string with a maximum of 64 alphanumeric characters.

For example, to associate an action list for SSL URL rewrite with a policy map, enter:

```
host1/Admin(config)# policy-map multi-match L4POLICY
host1/Admin(config-pmap)# class L4VIPCLASS
host1/Admin(config-pmap-c)# action SSL_ACTLIST
```

To disassociate the action list from the policy map, enter:

```
host1/Admin(config-pmap-c)# no action SSL_ACTLIST
```

For example, to associate an action list for HTTP header rewrite, enter:

```
host1/Admin(config-pmap-lb-c)# action HTTP_MODIFY_ACTLIST
```

To disassociate the action list from the policy map, enter:

```
host1/Admin(config-pmap-lb-c)# no action HTTP_MODIFY_ACTLIST
```
Discarding Requests

You can instruct the ACE to discard packets that match a particular policy map by using the `drop` command in policy map load-balancing class configuration mode. The syntax of this command is as follows:

```
drop
```

For example, enter:
```
host1/Admin(config-pmap-lb-c)# drop
```

To reset the behavior of the ACE to the default of accepting packets that match a policy map, enter:
```
host1/Admin(config-pmap-lb-c)# no drop
```

Forwarding Requests Without Load Balancing

You can instruct the ACE to forward requests that match a particular policy map without performing load balancing on the request by using the `forward` command in policy map load-balancing class configuration mode. The syntax of this command is as follows:

```
forward
```

For example, enter:
```
host1/Admin(config-pmap-lb-c)# forward
```

To reset the ACE to the default of load balancing packets that match a policy map, enter:
```
host1/Admin(config-pmap-lb-c)# no forward
```

Configuring HTTP Header Insertion

When the ACE uses Network Address Translation (NAT) to translate the source IP address of a client to a VIP, servers need a way to identify that client for the TCP and IP return traffic. To identify a client whose source IP address has been translated using NAT, you can instruct the ACE to insert a generic header and string value of your choice in the client HTTP request. (For information about NAT, see the Cisco Application Control Engine Module Security Configuration Guide.)
With either TCP server reuse or persistence rebalance enabled, the ACE inserts a header in every client request. For information about TCP server reuse, see the “Configuring TCP Server Reuse” section. For information about persistence rebalance, see the “Configuring HTTP Persistence Rebalance” section.

You can insert a generic header and value in an HTTP request by using the `insert-http` command in policy map load-balancing class configuration mode. You can enter multiple `insert-http` commands for each class. The syntax of this command is as follows:

```
insert-http name header-value expression
```

The keywords and arguments are as follows:

- **name**—Name of the HTTP header to insert in the client HTTP request. Enter an unquoted text string with no spaces and a maximum of 64 alphanumeric characters. You can specify any custom header name that you want, subject to the maximum character length. You can also enter any of the predefined header names in Table 3-4, regardless of whether that header name already exists in the client request header. The ACE does not overwrite any existing header information in the client request.

- **header-value expression**—Specifies the header-value expression string to insert in the HTTP header. Enter a text string with a maximum of 512 alphanumeric characters. If you configure more than 512 bytes of data to be inserted into the HTTP header, the ACE does not insert any data in the header. You can also specify the following special **header-value** expressions using the following dynamic replacement strings:

  - `%is`—Inserts the source IP address in the HTTP header
  - `%id`—Inserts the destination IP address in the HTTP header
  - `%ps`—Inserts the source port in the HTTP header
  - `%pd`—Inserts the destination port in the HTTP header

For Microsoft Outlook Web Access (OWA), specify the field name as `HTTP_FRONT_END_HTTPS` with a value of ON.
You can associate a maximum of 1024 instances of the same type of regex with a Layer 4 policy map. This limit applies to all Layer 7 policy-map types, including generic, HTTP, RADIUS, RDP, RTSP, and SIP. You configure regexes in the following:

- Match statements in Layer 7 class maps
- Inline match statements in Layer 7 policy maps
- Layer 7 hash predictors for server farms
- Layer 7 sticky expressions in sticky groups
- Header insertion and rewrite (including SSL URL rewrite) expressions in Layer 7 action lists

For example, in an SSL configuration, you could insert a generic field called ClientCert, and the header value could be the client certificate or a portion thereof.

For example, to insert the header name Host with a header value of www.cisco.com in an HTTP client request header, enter:

```
host1/Admin(config)# policy-map type loadbalance first-match L7SLBPOLICY
host1/Admin(config-pmap-lb)# class L7SLBCLASS
host1/Admin(config-pmap-lb-c)# insert-http Host header-value www.cisco.com
```

The header name and value will appear in the HTTP header as follows:

```
Host: www.cisco.com
```

To remove the HTTP header name and value from the policy map, enter:

```
host1/Admin(config-pmap-lb-c)# no insert-http Host header-value www.cisco.com
```

### Enabling Load Balancing to a Server Farm

You can load balance a client request for content to a server farm by using the `serverfarm` command in policy map load-balancing class configuration mode. Server farms are groups of networked real servers that contain the same content and that typically reside in the same physical location.
The syntax of this command is as follows:

```
serverfarm name1 [backup name2 [aggregate-state]]
```

The keywords, arguments, and options are as follows:

- **name1**—Unique identifier of the server farm. Enter an unquoted text string with no spaces and a maximum of 64 alphanumeric characters.

- **backup name2**—(Optional) Designates an existing host (with valid content) or a redirect (sorry) server farm as a backup server farm in case all the real servers in the primary server farm become unavailable. You can configure one backup server farm for each existing primary server farm. When at least one server in the primary server farm becomes available again, the ACE sends all new connections back to the primary server farm. The ACE allows existing connections to the backup server farm to complete. You can fine-tune the conditions under which the primary server farm fails over and returns to service by configuring a partial server farm failover. For details about partial server farm failover, see the “Configuring a Partial Server Farm Failover” section in Chapter 2, Configuring Real Servers and Server Farms. Enter the name of an existing server farm that you want to specify as a backup server farm as an unquoted text string with no spaces and a maximum of 64 alphanumeric characters.

**Note**
If all servers in the server farm fail and you did not configure a backup server farm, the ACE sends a reset (RST) to a client in response to a connection request.

- **aggregate-state**—This option has been deprecated and no longer has an effect on the state of the VIP. By default, the ACE takes into account the state of all real servers in the backup server farm before taking the VIP out of service. If all real servers in the primary server farm fail, but there is at least one real server in the backup server farm that is operational, the ACE keeps the VIP in service.

The following example specifies the `serverfarm` command as an action in a Layer 7 load-balancing policy map:

```
host1/Admin(config)# policy-map type loadbalance first-match
L7SLBPOLICY
host1/Admin(config-pmap-lb)# class L7SLBCLASS
host1/Admin(config-pmap-lb-c)# serverfarm SFARM1 backup SFARM2
```
To remove the server-farm action from the Layer 7 load-balancing policy map, enter:

```
host1/Admin(config-pmap-lb-c)# no serverfarm FARM2
```

### Configuring a Sorry Server Farm

When the primary server farm is unavailable, you can instruct the ACE to send client requests to a sorry server farm. A sorry server is a redirect server in a backup server farm with content stating that the web page, resource, or service that a client requested is temporarily unavailable. When at least one server in the primary server farm returns to service, the ACE directs clients back to the primary server farm. You can fine-tune the conditions under which the primary server farm fails over and returns to service by configuring a partial server farm failover. For details about partial server farm failover, see the “Configuring a Partial Server Farm Failover” section in Chapter 2, Configuring Real Servers and Server Farms.

To configure a sorry server, use the `serverfarm` command in policy map class configuration mode as described in the “Enabling Load Balancing to a Server Farm” section and configure an existing redirect server farm as the backup server farm. If you specifically want client connections to return to the same server in the primary server farm that they were connected to before the primary went down based on a client source IP addresses, configure the `predictor hash address source command` as the load-balancing method for the primary server farm. If the primary server farm is down and you want all requests from a particular subnet to be redirected to a particular site and requests from a different subnet to be redirected to a different site, use the `predictor hash address source` command as the load-balancing method for the sorry server farm. For equal load balancing across the sites in either server farm, use the `roundrobin` predictor method. Otherwise, you can configure any supported predictor method that works for your application on either server farm. For information about configuring the server farm predictor, see Chapter 2, Configuring Real Servers and Server Farms.
For example, to configure a primary server farm and a sorry server farm, enter the following commands:

```
host1/Admin(config)# rserver SERVER1
host1/Admin(config-rserver-host)# ip address 192.168.12.4
host1/Admin(config-rserver-host)# inservice
host1/Admin(config)# rserver SERVER2
host1/Admin(config-rserver-host)# ip address 192.168.12.5
host1/Admin(config-rserver-host)# inservice
host1/Admin(config)# rserver redirect SERVER3
host1/Admin(config-rserver-redir)# webhost-redirection www.cisco.com 301
host1/Admin(config-rserver-redir)# inservice
host1/Admin(config-rserver-redir)# exit
host1/Admin(config)# serverfarm SFARM1
host1/Admin(config-sfarm-host)# predictor roundrobin
host1/Admin(config-sfarm-host)# rserver SERVER1
host1/Admin(config-sfarm-host-rs)# inservice
host1/Admin(config-sfarm-host-rs)# exit
host1/Admin(config-sfarm-host)# rserver SERVER2
host1/Admin(config-sfarm-host-rs)# inservice
host1/Admin(config-sfarm-host-rs)# exit
host1/Admin(config-sfarm-host)# exit
host1/Admin(config)# serverfarm redirect SFARM2
host1/Admin(config-sfarm-redirect)# predictor roundrobin
host1/Admin(config-sfarm-redirect)# rserver SERVER3
host1/Admin(config-sfarm-redirect-rs)# inservice
host1/Admin(config-sfarm-redirect-rs)# exit
host1/Admin(config-sfarm-redirect)# rserver SERVER4
host1/Admin(config-sfarm-redirect-rs)# inservice
host1/Admin(config-sfarm-redirect-rs)# exit
host1/Admin(config-sfarm-redirect)# exit
host1/Admin(config)# class-map type http loadbalance match-any L7SLBCLASS
host1/Admin(config-cmap-http-lb)# 100 match http header Host header-value .*cisco.com
host1/Admin(config-cmap-http-lb)# exit
```
Configuring a Layer 7 Policy Map for SLB

host1/Admin(config)# policy-map type loadbalance first-match L7SLBPOLICY
host1/Admin(config-pmap-lb)# class L7SLBCLASS
host1/Admin(config-pmap-lb-c)# serverfarm SFARM1 backup SFARM2

Configuring a Sticky Server Farm

You can specify that requests matching a Layer 7 policy map be load balanced to a sticky server farm by using the `sticky-serverfarm` command in policy map load-balancing class configuration mode. The syntax of this command is as follows:

```
sticky-serverfarm name
```

The `name` argument is the identifier of an existing sticky group. Enter an unquoted text string with no spaces and a maximum of 64 alphanumeric characters. For information about sticky groups, see Chapter 5, Configuring Stickiness.

For example, enter:

```
host1/Admin(config-pmap-lb-c)# sticky-serverfarm STICKY_GROUP1
```

To remove the sticky server farm from the policy map, enter:

```
host1/Admin(config-pmap-lb-c)# no sticky-serverfarm STICKY_GROUP1
```

Specifying the IP Differentiated Services Code Point of Packets

You can specify the IP differentiated services code point (DSCP) of packets in a policy map by using the `set ip tos` command in policy map load-balancing class configuration mode. This command marks a packet by setting the IP DSCP bit in the Type of Service (ToS) byte. Once the IP DSCP bit is set, other Quality of Service (QoS) services can operate on the bit settings.

The syntax of this command is as follows:

```
set ip tos value
```

The `value` argument is the IP DSCP value. Enter an integer from 0 to 255. The default is to not modify the ToS field.
The following example specifies the set ip tos command as a QoS action in the Layer 7 load-balancing policy map. All packets that satisfy the match criteria of L7SLBCLASS are marked with the IP DSCP value of 8. How packets marked with the IP DSCP value of 8 are treated is determined by the network configuration.

```
host1/Admin(config)# policy-map type loadbalance first-match L7SLBPOLICY
host1/Admin(config-pmap)# class L7SLBCLASS
host1/Admin(config-pmap-lb-c)# set ip tos 8
```

To reset the ACE to the default of not modifying the ToS byte value, enter:

```
host1/Admin(config-pmap-lb-c)# no set ip tos 8
```

### Specifying an SSL Proxy Service

The ACE uses an SSL proxy service in a Layer 7 policy map to load balance outbound SSL initiation requests to SSL servers. The ACE acts as an SSL client sending an encrypted request to an SSL server. For more information about SSL initiation, see the *Cisco Application Control Engine Module SSL Configuration Guide*.

To specify an SSL proxy service in a policy map, use the `ssl-proxy` command in policy map load-balancing class configuration mode. The syntax of this command is as follows:

```
ssl-proxy client name
```

The `name` argument is the identifier of an existing SSL proxy service. Enter an unquoted text string with no spaces and a maximum of 64 alphanumeric characters.

For example, enter:

```
host1/Admin(config-pmap-lb-c)# ssl-proxy client PROXY_SERVICE1
```

To remove the SSL proxy service from the policy map, enter:

```
host1/Admin(config-pmap-lb-c)# no ssl-proxy client PROXY_SERVICE1
```
Associating a Layer 7 Policy Map with a Layer 3 and Layer 4 Policy Map

You can associate a Layer 7 SLB policy with a Layer 3 and Layer 4 SLB policy by using the `service-policy type loadbalance` command in policy-map class configuration mode. For details, see the “Associating a Layer 7 SLB Policy Map with a Layer 3 and Layer 4 SLB Policy Map” section.

Configuring a Generic Protocol Parameter Map

You can use a parameter map to combine related actions for generic protocol parsing. You reference this parameter map in the policy map by using the `appl-parameter generic advanced-options` command. See the “Associating a Generic, HTTP, or RTSP Parameter Map with a Layer 3 and Layer 4 Policy Map” section.

You can configure generic protocol actions for SLB connections by using the `parameter-map type generic` command in configuration mode. The syntax of this command is as follows:

```
parameter-map type generic name
```

The `name` argument is the identifier assigned to the parameter map. Enter an unquoted text string with no spaces and a maximum of 64 alphanumeric characters.

For example, enter:

```
host1/Admin(config)# parameter-map type generic GEN_PARAMETER_MAP
```

To remove a generic parameter map from the configuration, enter:

```
host1/Admin(config)# no parameter-map type generic GEN_PARAMETER_MAP
```

The following topics describe how to use the commands to define the generic protocol parameter map:

- Disabling Case-Sensitivity Matching for Generic Protocols
- Setting the Maximum Number of Bytes to Parse for Generic Protocols
Disabling Case-Sensitivity Matching for Generic Protocols

By default, the ACE CLI is case sensitive. To disable case-sensitivity matching for generic protocols only, use the `case-insensitive` command in generic parameter-map configuration mode. With case-insensitive matching enabled, uppercase and lowercase letters are considered the same.

The syntax of this command is as follows:

```
    case-insensitive
```

For example, to disable case sensitivity, enter:

```
host1/Admin(config-parammap-generi)# case-insensitive
```

To reenable case-sensitive matching after it has been disabled, enter:

```
host1/Admin(config-parammap-generi)# no case-insensitive
```

Setting the Maximum Number of Bytes to Parse for Generic Protocols

You can set the maximum number of bytes to parse for generic protocols by using the `set max-parse-length` command in generic parameter-map configuration mode. The syntax of this command is as follows:

```
    set max-parse-length bytes
```

The `bytes` argument is the maximum number of bytes to parse. Enter an integer from 1 to 65535. The default is 2048 bytes.

For example, to set the maximum parse length to 8192, enter:

```
host1/Admin(config-parammap-generi)# set max-parse-length 8192
```

To reset the maximum parse length to the default of 2048 bytes, enter:

```
host1/Admin(config-parammap-generi)# no set max-parse-length
```
Configuring an HTTP Parameter Map

You can use a parameter map to combine related HTTP actions for a Layer 3 and Layer 4 policy map. You reference this parameter map in the policy map by using the `appl-parameter http advanced-options` command. See the “Associating a Generic, HTTP, or RTSP Parameter Map with a Layer 3 and Layer 4 Policy Map” section.

You can configure advanced HTTP behavior for SLB connections by using the `parameter-map type http` command in configuration mode. The syntax of this command is as follows:

```
parameter-map type http name
```

The `name` argument is the identifier assigned to the parameter map. Enter an unquoted text string with no spaces and a maximum of 64 alphanumeric characters.

For example, enter:

```
host1/Admin(config)# parameter-map type http HTTP_PARAMETER_MAP
```

To remove an HTTP parameter map from the configuration, enter:

```
host1/Admin(config)# no parameter-map type http HTTP_PARAMETER_MAP
```

The following topics describe how to use the commands to define the advanced HTTP parameter map:

- Disabling Case-Sensitivity Matching for HTTP
- Configuring the ACE to Modify Headers on Every HTTP Request or Response
- Defining URL Delimiters
- Setting the Maximum Number of Bytes to Parse for Content
- Setting the Maximum Number of Bytes to Parse for Cookies, HTTP Headers, and URLs
- Configuring the ACE Behavior when a URL or Cookie Exceeds the Maximum Parse Length
- Configuring HTTP Persistence Rebalance
- Configuring TCP Server Reuse
Disabling Case-Sensitivity Matching for HTTP

By default, the ACE CLI is case sensitive. To disable case-sensitivity matching for HTTP only, use the `case-insensitive` command in HTTP parameter-map configuration mode. With case-insensitive matching enabled, uppercase and lowercase letters are considered the same. When case sensitivity is disabled, it applies to the following:

- HTTP header names and values
- HTTP cookie names and values
- URL strings
- HTTP deep inspection (for details, see the *Cisco Application Control Engine Module Security Configuration Guide* )

The syntax of this command is as follows:

```
case-insensitive
```

For example, to disable case sensitivity, enter:

```
host1/Admin(config-parammap-http)# case-insensitive
```

To reenable case-sensitive matching after it has been disabled, enter:

```
host1/Admin(config-parammap-http)# no case-insensitive
```

Configuring the ACE to Modify Headers on Every HTTP Request or Response

By default, when the `persistence-rebalance` command is disabled and you configure the ACE to modify HTTP headers (insert, delete, or rewrite), the ACE performs the operation only on the first HTTP request or response. The `persistence-rebalance` command causes the ACE to perform header modifications on each request or response, but it also instructs the ACE to load balance each new request to a potentially new real server. For more information about the `persistence-rebalance` command, see the “Configuring HTTP Persistence Rebalance” section. For information about header insertion, deletion, and rewrite, see the “Configuring HTTP Header Insertion, Deletion, and Rewrite” section.
To instruct the ACE to modify headers (insert, delete, or rewrite) on every HTTP request or response without the additional effect of performing load balancing on each new HTTP request caused by the `persistence-rebalance` command, use the `header modify per-request` command in parameter map HTTP configuration mode. This command has an effect only when `persistence-rebalance` is disabled. The syntax of this command is:

```
header modify per-request
```

The `header modify per-request` command also causes the ACE to perform URL location header rewrite on every HTTP response if the `ssl url rewrite location` command is enabled. For more information about SSL URL rewrite, see the Cisco Application Control Engine Module SSL Configuration Guide.

For example, to instruct the ACE to perform header modification on every HTTP request or response, enter the following command:

```
host1/Admin(config-parammap-http)# header modify per-request
```

To return the ACE behavior to the default of modifying headers only on the first HTTP request or response, enter the following command:

```
host1/Admin(config-parammap-http)# no header modify per-request
```

### Defining URL Delimiters

You can define a list of ASCII-character delimiters that you can use to separate the cookies in a URL string by using the `set secondary-cookie-delimiters` command in parameter map HTTP configuration mode. The syntax of this command is as follows:

```
set secondary-cookie-delimiters text
```

The `text` argument identifies the list of delimiters. Enter an unquoted text string with no spaces and a maximum of four characters. The order of the delimiters in the list does not matter. The default list of delimiters is `/&#+`.

Cookies and their delimiters appear in GET request lines. In the following example of a GET request line, the ampersand (`&`) that appears between name-value pairs is the secondary cookie delimiter. The question mark (`?`) begins the URL query and is not configurable.

```
GET /default.cgi?user=me&hello=world&id=2 HTTP/1.1
```
For example, to specify the secondary cookie delimiters as !@#$, enter:

```
host1/Admin(config-parammap-http)# set secondary-cookie-delimiters !@#$
```

To reset the delimiter list to the default of /&#+, enter:

```
host1/Admin(config-parammap-http)# no set secondary-cookie-delimiters !@#$
```

### Defining the Secondary Cookie Start

You can define the ASCII-character string at the start of a secondary cookie in a URL or ignore any start string of a secondary cookie in the URL and consider the secondary cookie part of the URL by using the `set secondary-cookie-start` command in parameter map HTTP configuration mode. The syntax of this command is as follows:

```
set secondary-cookie-start { none | text }
```

The keyword and argument are as follows:

- **none**—The secondary cookie start is not configured or the ACE ignores any start string of a secondary cookie in the URL and considers the secondary cookie as part of the URL.

  When you configure the `none` keyword to consider the entire URL query string as part of a URL, the commands that rely on the URL query, such as the `match cookie secondary` and `predictor hash cookie secondary` commands, do not work. Do not configure these commands under the same real server.

- **text**—The start string of the secondary cookie. Enter a maximum of two characters. The default start character is `?`.

For example, to define the secondary cookie start string, enter:

```
host1/Admin(config-parammap-http)# set secondary-cookie-start ?!
```

To reset the secondary cookie start string to the default setting of `?`, enter:

```
host1/Admin(config-parammap-http)# no set secondary-cookie-start
```
Setting the Maximum Number of Bytes to Parse for Content

By default, the maximum number of bytes that the ACE parses to check for content is 4096. If a content string exceeds the default value, the ACE drops the packet and sends a RST (reset) to the client browser. You can increase the number of bytes that the ACE parses using the `set content-maxparse-length` command in HTTP parameter map configuration mode. The syntax of this command is as follows:

```
set content-maxparse-length bytes
```

The `bytes` argument is the maximum number of bytes to parse for the total length of a content string. Enter an integer from 1 to 65535. The default is 4096 bytes.

*Note*  
If you set the `bytes` argument to a value that is greater than 32 KB, you must configure the `set tcp buffer-share` command in a connection parameter map to a value that is greater than the `bytes` value. If you do not, even if you configure the `length-exceed continue` command, the ACE may not completely parse a content string packet that is greater than 32 KB. The reason is that the default value of the `set tcp buffer-share` command buffer size (32 KB) will not accommodate the larger content string size.

For example, to set the content maximum parse length to 8192, enter:

```
host1/Admin(config-parammap-http)# set content-maxparse-length 8192
```

To reset the HTTP content maximum parse length to the default of 4096 bytes, enter:

```
host1/Admin(config-parammap-http)# no set content-maxparse-length
```
Setting the Maximum Number of Bytes to Parse for Cookies, HTTP Headers, and URLs

By default, the maximum number of bytes that the ACE parses to check for a cookie, HTTP header, or URL is 4096. If a cookie, HTTP header, or URL exceeds the default value, the ACE drops the packet and sends a RST (reset) to the client browser. You can increase the number of bytes that the ACE parses using the `set header-maxparse-length` command in HTTP parameter-map configuration mode. The syntax of this command is as follows:

```
set header-maxparse-length bytes
```

The `bytes` argument is the maximum number of bytes to parse for the total length of all cookies, HTTP headers, and URLs. Enter an integer from 1 to 65535. The default is 4096 bytes.

**Note**

If you set the `bytes` argument to a value that is greater than 32 KB, you must configure the `set tcp buffer-share` command in a connection parameter map to a value that is greater than the `bytes` value. If you do not, even if you configure the `length-exceed continue` command, the ACE may not completely parse a header packet that is greater than 32 KB. The reason is that the default value of the `set tcp buffer-share` buffer size (32 KB) will not accommodate the larger header size.

For example, to set the HTTP header maximum parse length to 8192, enter:

```
host1/Admin(config-parammap-http)# set header-maxparse-length 8192
```

To reset the HTTP header maximum parse length to the default of 4096 bytes, enter:

```
host1/Admin(config-parammap-http)# no set header-maxparse-length
```
Configuring the ACE Behavior when a URL or Cookie Exceeds the Maximum Parse Length

You can configure how the ACE handles cookies, HTTP headers, and URLs that exceed the maximum parse length by using the `length-exceed` command in HTTP parameter-map configuration mode. The syntax of this command is as follows:

```
length-exceed { continue | drop }
```

The keywords are as follows:

- **continue**—Specifies how to continue load balancing. When you specify this keyword, the `persistence-rebalance` command is disabled if the total length of all cookies, HTTP headers, and URLs exceeds the maximum parse length value. For details on setting the maximum parse length, see the “Setting the Maximum Number of Bytes to Parse for Cookies, HTTP Headers, and URLs” section.

- **drop**—(Default) Specifies how to stop load balancing and discard the packet.

For example, enter:

```
host1/Admin(config-parammap-http)# length-exceed continue
```

To reset the ACE to the default of stopping load balancing and discarding a packet when its URL or cookie exceeds the maximum parse length, enter:

```
host1/Admin(config-parammap-http)# no length-exceed continue
```

Configuring HTTP Persistence Rebalance

When persistence rebalance is disabled (the default for the module), the ACE matches an HTTP request to a Layer 7 class map in a Layer 7 load-balancing policy map and load balances the request to one of the servers in the serverfarm associated with that class map. The ACE sends all subsequent requests on the same TCP connection to the same server regardless of whether or not they match the same Layer 7 class map.
You can enable the persistence rebalance feature by using the **persistence-rebalance** command in HTTP parameter-map configuration mode. (Be sure to apply the HTTP parameter map to a Layer 4 multi-match policy map.) The syntax of this command is as follows:

```
persistence rebalance
```

With persistence rebalance enabled, when the first HTTP request arrives, the ACE matches the request to a Layer 7 class map in a Layer 7 policy map and load balances the request to one of the servers in the serverfarm associated with that class map. The ACE matches all subsequent requests on the same TCP connection to a Layer 7 class map. If the subsequent request matches the same Layer 7 class map as the previous request, then the ACE sends the request to the same server as the previous request. This behavior produces less overhead and better performance.

If the request matches a different Layer 7 class map, then the ACE load balances the request to one of the servers in the serverfarm associated with the newly matched Layer 7 class map according to the serverfarm predictor.

When persistence rebalance is enabled, header insertion and cookie insertion, if enabled, occur for every request instead of only the first request. For information about header insertion, see the “Configuring HTTP Header Insertion” section in this chapter. For information about cookie insertion, see the “Enabling Cookie Insertion” section in Chapter 5, Configuring Stickiness.

---

**Note**

If a real server is enabled with the NTLM Microsoft authentication protocol, we recommend that you leave persistence rebalance disabled. NTLM is a security measure that is used to perform authentication with Microsoft remote access protocols. When a real server is enabled with NTLM, every connection to the real server must be authenticated; typically, each client user will see a pop-up window prompting for a username and password. Once the connection is authenticated, all subsequent requests on the same connection will not be challenged. However, when the server load balancing function is enabled and configured with persistence rebalance, a subsequent request may point to a different real server causing a new authentication handshake.
The following example specifies the `parameter-map type http` command to configure URL cookie delimiter strings, set the maximum number of bytes to parse for URLs and cookies, and enable HTTP persistence:

```
host1/Admin(config)# parameter-map type http http_parameter_map
host1/Admin(config-parammap-http)# secondary-cookie-delimiters !@#$
host1/Admin(config-parammap-http)# header-maxparse-length 8192
length-exceed continue
host1/Admin(config-parammap-http)# persistence-rebalance
```

To reset persistence to the default setting of disabled, enter:

```
host1/Admin(config-parammap-http)# no persistence-rebalance
```

### Configuring Persistence with Load Balancing on Each HTTP Request

When the persistence-rebalance feature is enabled on the ACE, it does not load balance successive GET requests on the same TCP connection unless it matches a load-balancing class map that is different from the load-balancing class map matched by the previous request (see the “Configuring HTTP Persistence Rebalance” section). To configure the ACE to load balance each subsequent GET request on the same TCP connection independently, use the `persistence-rebalance strict` command in HTTP parameter-map configuration mode.

The syntax of this command is as follows:

```
persistence-rebalance strict
```

This command allows the ACE to load balance each HTTP request to a potentially different Layer 7 class or real server.

For example, to enable the strict persistence rebalance feature, enter:

```
host1/Admin(config)# parameter-map type http http_parameter_map
host1/Admin(config-parammap-http)# persistence-rebalance strict
```

To reset persistence to the default setting of disabled, enter:

```
host1/Admin(config-parammap-http)# no persistence-rebalance
```

To change to persistence rebalance behavior that does not load balance successive requests to the same connection, use the `persistence-rebalance` command.
Configuring TCP Server Reuse

TCP server reuse allows the ACE to reduce the number of open connections on a server by allowing connections to persist and be reused by multiple client connections. The ACE maintains a pool of TCP connections based on TCP options. New client connections can reuse those connections in the pool if the new client connections and prior server connections share the same TCP options. For information about configuring how the ACE handles TCP options, see the Cisco Application Control Engine Module Security Configuration Guide.

To ensure proper operation of this feature, follow these TCP server reuse recommendations and restrictions:

- Ensure that the ACE maximum segment size (MSS) is the same as the server MSS.
- Configure port address translation (PAT) on the interface that is connected to the real server. PAT prevents collisions when a client stops using a server connection, and then that connection is reused by another client. Without PAT, if the original client tries to reuse the original server connection, it is no longer available. For details about configuring PAT, see the Cisco Application Control Engine Module Security Configuration Guide.
- Configure the ACE with the same TCP options that exist on the TCP server.
- Ensure that each server farm is homogeneous (all real servers within a server farm have identical configurations).

Note

Always configure PAT with TCP reuse. If you configure TCP reuse and NAT only, unexpected results may occur.

Another effect of TCP server reuse is that header insertion and cookie insertion, if enabled, occur for every request instead of only the first request. For information about header insertion, see the “Configuring HTTP Header Insertion” section in this chapter. For information about cookie insertion, see the “Enabling Cookie Insertion” section in Chapter 5, Configuring Stickiness.

You can configure TCP server reuse by using the server-conn reuse command in HTTP parameter-map configuration mode. The syntax of this command is as follows:

```
server-conn reuse
```
For example, to enable TCP server reuse, enter:

```
host1/Admin(config-parammap-http)# server-conn reuse
```

To disable TCP server reuse, enter:

```
host1/Admin(config-parammap-http)# no server-conn reuse
```

## Configuring an RTSP Parameter Map

You can use a parameter map to combine related RTSP actions for a Layer 3 and Layer 4 policy map. You reference this parameter map in the policy map using the `appl-parameter rtsp advanced-options` command. See the “Associating a Generic, HTTP, or RTSP Parameter Map with a Layer 3 and Layer 4 Policy Map” section.

You can configure advanced RTSP behavior for SLB connections by using the `parameter-map type rtsp` command in configuration mode. The syntax of this command is as follows:

```
parameter-map type rtsp name
```

The `name` argument is the identifier assigned to the parameter map. Enter an unquoted text string with no spaces and a maximum of 64 alphanumeric characters.

For example, enter:

```
host1/Admin(config)# parameter-map type rtsp RTSP_PARAMETER_MAP
```

To remove an RTSP parameter map from the configuration, enter:

```
host1/Admin(config)# no parameter-map type rtsp RTSP_PARAMETER_MAP
```

You can define the advanced RTSP parameter map by using the commands in the following topics:

- **Disabling Case-Sensitivity Matching for RTSP**
- **Setting the Maximum Number of Bytes to Parse for RTSP Headers**
Disabling Case-Sensitivity Matching for RTSP

By default, the ACE CLI is case sensitive. To disable case-sensitivity matching for RTSP, use the case-insensitive command in RTSP parameter-map configuration mode. With case-insensitive matching enabled, uppercase and lowercase letters are considered the same. When case sensitivity is disabled, it applies to the following:

- RTSP header names and values
- RTSP URL strings
- RTSP inspection (for details, see the Cisco Application Control Engine Module Security Configuration Guide)

The syntax is as follows:

```
case-insensitive
```

For example, to disable case sensitivity, enter:

```
host1/Admin(config-parammap-rtsp)# case-insensitive
```

To reenable case-sensitive matching after it has been disabled, enter:

```
host1/Admin(config-parammap-rtsp)# no case-insensitive
```

Setting the Maximum Number of Bytes to Parse for RTSP Headers

You can set the maximum number of bytes to parse for RTSP headers by using the set header-maxparse-length command in RTSP parameter map configuration mode. The syntax of this command is as follows:

```
set header-maxparse-length bytes
```

The `bytes` argument is the maximum number of bytes to parse for the total length of all RTSP headers. Enter an integer from 1 to 65535. The default is 2048 bytes.

For example, to set the RTSP header maximum parse length to 16384, enter:

```
host1/Admin(config-parammap-rtsp)# set header-maxparse-length 16384
```
To reset the RTSP header maximum parse length to the default of 2048 bytes, enter:

```
host1/Admin(config-parammap-rtsp)# no set header-maxparse-length
```

## Configuring a Layer 3 and Layer 4 Class Map for SLB

A Layer 3 and Layer 4 class map contains match criteria to classify network traffic that can pass through the ACE. The ACE uses these Layer 3 and Layer 4 traffic classes to perform server load balancing (SLB). For a Layer 3 and Layer 4 traffic classification, the match criteria in a class map include the VIP address, protocol, and port of the ACE. You can configure multiple commands in a single class map to specify the match criteria in a group that you then associate with a traffic policy.

You can create a Layer 3 and Layer 4 class map to classify network traffic passing through the ACE and enter class-map configuration mode by using the `class-map` command in configuration mode. The syntax of this command is as follows:

```
class-map [match-all | match-any] map_name
```

The arguments and options are as follows:

- **match-all | match-any**—(Optional) Determines how the ACE evaluates Layer 3 and Layer 4 network traffic when multiple match criteria exist in a class map. The class map is considered a match if the `match` commands meet one of the following conditions:
  - **match-all**—(Default) Network traffic needs to satisfy all of the match criteria (implicit AND) to match the class map.
  - **match-any**—Network traffic needs to satisfy only one of the match criteria (implicit OR) to match the load-balancing class map.

- **map_name**—Name assigned to the class map. Enter an unquoted text string with no spaces and a maximum of 64 alphanumeric characters. The class name is used for both the class map and to configure policy for the class in the policy map.

For example, to create a class map named L4VIPCLASS that specifies the network traffic must satisfy all the match criteria (match-all is the default), enter:

```
host1/Admin(config)# class-map L4VIPCLASS
```
To remove the class map from the configuration, enter:

```
host1/Admin(config)# no class-map L4VIPCLASS
```

After you have created a Layer 3 and Layer 4 class map for SLB, use the commands in the following topics to configure a description and the VIP match criteria for the class map:

- **Defining a Class Map Description**
- **Defining VIP Address Match Criteria**

### Defining a Class Map Description

You can provide a brief summary about the Layer 3 and Layer 4 class map by using the `description` command in class-map configuration mode. The syntax of this command is as follows:

```
description text
```

For the `text` argument, enter an unquoted text string with no spaces and a maximum of 240 alphanumeric characters.

For example, to specify a description where the class map filters network traffic to the server, enter:

```
host1/Admin(config)# class-map match-any L4VIPCLASS
host1/Admin(config-cmap)# description filter server traffic
```

To remove the description from the class map, enter:

```
host1/Admin(config-cmap)# no description
```

### Defining VIP Address Match Criteria

You can define a 3-tuple flow of VIP address, protocol, and port as matching criteria for SLB by using the `match virtual-address` command in class-map configuration mode. You can configure multiple match criteria statements to define the VIPs for SLB.
Previous to release A2(2.0), the ACE allowed you to configure a class-map VIP address that overlaps with an ACE interface IP address. The ACE no longer ignores this configuration and displays the following warning:

```
Error: Entered VIP address is not the first address in the VIP range
```

The syntax of this command is as follows:

```
[line_number] match virtual-address vip_address { [mask] | any | { tcp | udp { any | eq port_number | range port1 port2 } } | protocol_number }
```

The keywords, arguments, and options are as follows:

- **line_number**—(Optional) Line numbers that you use for editing or deleting individual match commands. For example, you can enter `no line_number` to delete long match commands instead of entering the entire line.

- **vip_address**—Virtual IP (VIP) address of the virtual server in the ACE specified in dotted-decimal notation (192.168.1.2). VIPs are public addresses owned by the customer.

- **mask**—(Optional) Mask for the VIP address of the ACE to allow connections to an entire network specified in dotted decimal format (255.255.255.0).

- **any**—Specifies the wildcard value for the IP protocol value.

- **tcp | udp**—Specifies the protocol, TCP or UDP.
  
  - **any**—Specifies the wildcard value for the TCP or UDP port number. With `any` used in place of either the `eq` or `range` values, packets from any incoming port match.
  
  - **eq port_number**—Specifies that the TCP or UDP port number must match the specified value. Enter an integer from 0 to 65535. A value of 0 instructs the ACE to include all ports. Alternatively, you can enter the keyword name of a well-known TCP port from Table 3-6 or a well-known UDP port from Table 3-7.
Table 3-6  **Well-Known TCP Port Numbers and Keywords**

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Port Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>domain</td>
<td>53</td>
<td>Domain Name System (DNS)</td>
</tr>
<tr>
<td>ftp</td>
<td>21</td>
<td>File Transfer Protocol (FTP)</td>
</tr>
<tr>
<td>ftp-data</td>
<td>20</td>
<td>FTP data connections</td>
</tr>
<tr>
<td>http</td>
<td>80</td>
<td>Hypertext Transfer Protocol (HTTP)</td>
</tr>
<tr>
<td>https</td>
<td>443</td>
<td>HTTP over TLS or SSL (HTTPS)</td>
</tr>
<tr>
<td>irc</td>
<td>194</td>
<td>Internet Relay Chat (IRC)</td>
</tr>
<tr>
<td>matip-a</td>
<td>350</td>
<td>Mapping of Airline Traffic over Internet Protocol (MATIP) Type A</td>
</tr>
<tr>
<td>nntp</td>
<td>119</td>
<td>Network News Transport Protocol (NNTP)</td>
</tr>
<tr>
<td>pop2</td>
<td>109</td>
<td>Post Office Protocol (POP) v2</td>
</tr>
<tr>
<td>pop3</td>
<td>110</td>
<td>Post Office Protocol (POP) v3</td>
</tr>
<tr>
<td>rdp</td>
<td>3389</td>
<td>Remote Desktop Protocol (RDP)</td>
</tr>
<tr>
<td>rtsp</td>
<td>554</td>
<td>Real-Time Streaming Protocol (RTSP)</td>
</tr>
<tr>
<td>sip</td>
<td>5060</td>
<td>Session Initiation Protocol (SIP)</td>
</tr>
<tr>
<td>skinny</td>
<td>2000</td>
<td>Skinny Client Control Protocol (SCCP)</td>
</tr>
<tr>
<td>smtp</td>
<td>25</td>
<td>Simple Mail Transfer Protocol (SMTP)</td>
</tr>
<tr>
<td>telnet</td>
<td>23</td>
<td>Telnet</td>
</tr>
<tr>
<td>www</td>
<td>80</td>
<td>World Wide Web (WWW)</td>
</tr>
<tr>
<td>xot</td>
<td>1998</td>
<td>X.25 over TCP</td>
</tr>
</tbody>
</table>

**Note**

The ACE supports both the `http` and the `www` literal keywords for port 80 as well as the port number `80` itself. Regardless of whether you configure a literal keyword or port 80, the “www” literal appears in the running configuration.
Chapter 3  Configuring Traffic Policies for Server Load Balancing

Configuring a Layer 3 and Layer 4 Class Map for SLB

Configuring a Layer 3 and Layer 4 Class Map for SLB

---

### Table 3-7 Well-Known UDP Port Numbers and Keywords

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Port Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>domain</td>
<td>53</td>
<td>Domain Name System</td>
</tr>
<tr>
<td>radius-acct</td>
<td>1813</td>
<td>Remote Authentication Dial-In User Service (accounting)</td>
</tr>
<tr>
<td>radius-auth</td>
<td>1812</td>
<td>Remote Authentication Dial-In User Service (server)</td>
</tr>
<tr>
<td>sip</td>
<td>5060</td>
<td>Session Initiation Protocol (SIP)</td>
</tr>
<tr>
<td>wsp</td>
<td>9200</td>
<td>Connectionless Wireless Session Protocol (WSP)</td>
</tr>
<tr>
<td>wsp-wtls</td>
<td>9202</td>
<td>Secure Connectionless WSP</td>
</tr>
<tr>
<td>wsp-wtp</td>
<td>9201</td>
<td>Connection-based WSP</td>
</tr>
<tr>
<td>wsp-wtp-wtls</td>
<td>9203</td>
<td>Secure Connection-based WSP</td>
</tr>
</tbody>
</table>

- **range** `port1 port2`—Specifies a port range to use for the TCP or UDP port. Valid port ranges are from 0 to 65535. A value of 0 instructs the ACE to match all ports.

- **protocol_number**—Number of an IP protocol. Enter an integer from 1 to 254 that represents an IP protocol number.

**Note**

The ACE always attempts to match incoming traffic to the configured classes in a Layer 4 multi-match policy on a first-match basis. When you configure two or more class maps with the same VIP address match criteria and you configure the protocol as `any` in the first class map, the ACE will not match incoming traffic to a more specific class map (one with a specified protocol and port) that follows the non-specific class map in a policy map. For example, if you configure `match virtual-address 192.168.12.15 any` in the first class map and `match virtual-address 192.168.12.15 tcp eq https` in the second class map and associate the classes in that order in a policy map, any HTTPS traffic received by the ACE will never match the intended class. Therefore, the ACE will loadbalance the traffic to the wrong server and you will receive The Page Cannot Be Displayed error in your browser. Always configure the more specific class first, or else you may experience unexpected results. If you configure the two classes in separate Layer 4 policy maps, be sure to apply the policies in the correct order on the interface using a service policy.
The following example specifies that the class map L4VIPCLASS matches traffic destined to VIP address 192.168.1.10 and TCP port number 80:

```
host1/Admin(config)# class-map L4VIPCLASS
host1/Admin(config-cmap)# match virtual-address 192.168.1.10 tcp port eq 80
```

To remove the VIP match statement from the class map, enter:

```
host1/Admin(config-cmap)# no match virtual-address 192.168.1.10 tcp port eq 80
```

### Configuring a Layer 3 and Layer 4 Policy Map for SLB

For a Layer 3 and Layer 4 SLB traffic classification, you create an SLB policy map that contains actions that are related to a VIP. A policy map associates a predefined traffic class (class map) with a series of actions to be performed on the traffic that matches the classifications defined in the traffic class. At the Layer 3 and Layer 4 network traffic level, there is a single policy map for each network traffic feature. The Layer 3 and Layer 4 policy maps are typed accordingly and, through the `service-policy` command, applied to a single interface or globally to all interfaces in a context.

The sequence in which the ACE applies actions for a specific policy are independent of the actions configured for a class inside a policy. The ACE follows an implicit feature lookup order that is dictated by the policy map actions and features, which can mean that the user-configured order of class maps does not necessarily have an effect on the lookup order. For example, if you configure one or more security ACLs in a policy map, an ACL may not allow a certain flow through the ACE even if you want to perform operations such as SLB on that flow. For details on the lookup order that the ACE uses, see the *Cisco Application Control Engine Module Administration Guide*.

To create an SLB policy map, use the `policy-map` command in configuration mode. The syntax of this command is as follows:

```
policy-map multi-match map_name
```
Defining a Layer 3 and Layer 4 Policy Map Description

You can use the `description` command to provide a brief summary about the Layer 3 and Layer 4 policy map.

You must access the policy map configuration mode to specify the `description` command.

The syntax of this command is as follows:

```
description text
```

Use the `text` argument to enter an unquoted text string with a maximum of 240 alphanumeric characters.
For example, to specify a description that the policy map is to filter network traffic to a VIP, enter:

```bash
host1/Admin(config-pmap)# description filter traffic matching a VIP
```

To remove the description from the class map, enter:

```bash
host1/Admin(config-pmap)# no description
```

## Associating a Layer 3 and Layer 4 Class Map with a Policy Map

You can associate a Layer 3 and Layer 4 SLB class map with a Layer 3 and Layer 4 SLB policy map by using the `class` command in policy-map configuration mode. The syntax of this command is as follows:

```
class {name1 [insert-before name2] | class-default}
```

The keywords, arguments, and options are as follows:

- `name1`—Name of a previously defined traffic class configured with the `class-map` command. Enter an unquoted text string with no spaces and a maximum of 64 alphanumeric characters.

- `class-default`—Specifies the reserved, well-known class map created by the ACE. You cannot delete or modify this class. All traffic that fails to meet the other matching criteria in the named class map belongs to the default traffic class. If none of the specified classifications match the traffic, then the ACE performs the action specified under the `class class-default` command. The `class-default` class map has an implicit match any statement in it enabling it to match all traffic.

- `insert-before name2`—(Optional) Places the current class map ahead of an existing class map specified by the `name2` argument in the policy-map configuration. The ACE does not preserve the command in the running configuration but does retain the configured order of class maps in the policy map.

For example, to use the `insert-before` command to define the sequential order of two class maps in the policy map, enter:

```bash
host1/Admin(config-pmap)# class L4VIPCLASS insert-before FILTERHTML
```

To remove a class map from a Layer 3 and Layer 4 policy map, enter:

```bash
host1/Admin(config-pmap)# no class L4VIPCLASS
```
Specifying Layer 3 and Layer 4 SLB Policy Actions

After you associate a Layer 3 and Layer 4 class map with an SLB policy map, you need to specify the actions that the ACE should take when network traffic matches one or more match statements in a class map. To specify the Layer 3 and Layer 4 SLB policy actions, use the commands described in the following topics:

- Associating a Layer 7 SLB Policy Map with a Layer 3 and Layer 4 SLB Policy Map
- Associating a Generic, HTTP, or RTSP Parameter Map with a Layer 3 and Layer 4 Policy Map
- Associating a Connection Parameter Map with a Layer 3 and Layer 4 Policy Map
- Enabling the Advertising of a Virtual Server IP Address
- Enabling a VIP to Reply to ICMP Requests
- Enabling Per-Packet Load Balancing for UDP Traffic
- Enabling a VIP

Associating a Layer 7 SLB Policy Map with a Layer 3 and Layer 4 SLB Policy Map

The ACE treats all Layer 7 policy maps as child policies, so you must always associate a Layer 7 SLB policy map with a Layer 3 and Layer 4 SLB policy map. You can apply on an interface or globally on all interfaces in a context only a Layer 3 and Layer 4 policy map and not a Layer 7 policy map. For details on creating a Layer 7 load-balancing policy map, see the “Configuring a Layer 7 Policy Map for SLB” section.

You can associate a Layer 7 SLB policy map with a Layer 3 and Layer 4 SLB policy map by using the `loadbalance` command in policy-map class configuration mode.

The syntax of this command is as follows:

```
loadbalance policy name
```

The `policy name` keyword and argument are the identifiers of an existing Layer 7 SLB policy map. Enter the name as an unquoted text string with no spaces and a maximum of 64 alphanumeric characters.
For example, to reference the Layer 7 L7SLBPOLICY policy map within the Layer 3 and Layer 4 L4SLBPOLICY policy map, enter:

```
host1/Admin(config)# policy-map type loadbalance first-match L7SLBPOLICY
host1/Admin(config-pmap-lb)# class L7SLBCLASS
host1/Admin(config-pmap-lb-c)# serverfarm FARM2
```

```
host1/Admin(config)# policy-map multi-match L4SLBPOLICY
host1/Admin(config-pmap)# class L4SLBCLASS
host1/Admin(config-pmap-c)# loadbalance policy L7SLBPOLICY
```

To dissociate the Layer 7 SLB policy from the Layer 3 and Layer 4 SLB policy, enter:

```
host1/Admin(config-pmap-c)# no loadbalance policy L7LBPOICLY
```

### Associating a Generic, HTTP, or RTSP Parameter Map with a Layer 3 and Layer 4 Policy Map

You can configure generic, HTTP, or RTSP parameters by creating a parameter map to define related actions. See the “Configuring a Generic Protocol Parameter Map”, “Configuring an HTTP Parameter Map”, or “Configuring an RTSP Parameter Map” section.

You can associate a parameter map with a Layer 3 and Layer 4 policy map by using the `appl-parameter advanced-options` command in policy-map class configuration mode.

The syntax of this command is as follows:

```
appl-parameter {generic | http | rtsp} advanced-options name
```

The keywords and arguments are as follows:

- `generic`—Specifies a generic protocol parameter map.
- `http`—Specifies an HTTP parameter map.
- `rtsp`—Specifies an RTSP parameter map.
- `name`—Identifier of an existing generic, HTTP, or RTSP parameter map. Parameter maps aggregate related traffic actions together.
For example, to specify the `appl-parameter http advanced-options` command as an action for the SLB policy map, enter:

```
host1/Admin(config)# policy-map multi-match L4SLBPOLICY
host1/Admin(config-pmap)# class FILTERHTTP
host1/Admin(config-pmap-c)# appl-parameter http advanced-options HTTP_PARAM_MAP1
```

To disassociate the HTTP parameter map as an action from the SLB policy map, enter:

```
host1/Admin(config-pmap-c)# no appl-parameter http advanced-options HTTP_PARAM_MAP1
```

**Associating a Connection Parameter Map with a Layer 3 and Layer 4 Policy Map**

You can configure TCP/IP normalization and connection parameters by creating a connection parameter map to define related actions. See the *Cisco Application Control Engine Module Security Configuration Guide*.

You can associate a connection parameter map with a Layer 3 and Layer 4 policy map by using the `connection advanced-options` command in policy-map class configuration mode. The syntax of this command is as follows:

```
connection advanced-options name
```

For the `name` argument, enter the name of an existing parameter map as an unquoted text string with no spaces and a maximum of 64 alphanumeric characters.

For example, enter:

```
host1/Admin(config-pmap-c)# connection advanced-options TCP_PARAM_MAP
```

To dissociate the TCP parameter map from a policy map, enter:

```
host1/Admin(config-pmap-c)# no connection advanced-options TCP_PARAM_MAP
```

**Enabling the Advertising of a Virtual Server IP Address**

You can allow the ACE to advertise the IP address of the virtual server as the host route by using the `loadbalance vip advertise` command in policy-map class configuration mode. This function is used with route health injection (RHI) to allow the ACE to advertise the availability of a VIP address throughout the network.
The syntax of this command is as follows:

```
loadbalance vip advertise [active] [metric number]
```

The keywords, arguments, and options are as follows:

- **active**—(Optional) Allows the ACE to advertise the IP address of the virtual server (VIP) as the host route only if there is at least one active real server in the server farm. Without the `active` option, the ACE always advertises the VIP whether or not there is any active real server associated with this VIP.

- **metric number**—(Optional) Specifies the distance metric for the route that needs to be entered in the routing table. Enter an integer from 1 to 254. The default is 77.

---

**Note**  You must enable the advertising of a VIP using the `loadbalance vip advertise` command before you can enter a distance `metric` value for the route. Otherwise, the ACE returns an error message.

---

**Note**  If you configured the `loadbalance vip advertise metric` command and then you enter the `no loadbalance vip advertise [active]` command, the ACE resets the `metric` value to the default of 77.

For example, to specify the `loadbalance vip advertise` command as an action for the SLB policy map, enter:

```
host1/Admin(config)# policy-map multi-match L4SLBPOLICY
host1/Admin(config-pmap)# class FILTERHTTP
host1/Admin(config-pmap-c)# loadbalance vip advertise active
```

To stop advertising the host route as an action in the SLB policy map and to reset the value of the `loadbalance vip advertise metric` command (if configured) to the default of 77, enter:

```
host1/Admin(config-pmap-c)# no loadbalance vip advertise active
```

To specify a distance metric for the route, enter:

```
host1/Admin(config-pmap-c)# loadbalance vip advertise metric 30
```

To remove the distance metric from the advertised VIP, enter:

```
host1/Admin(config-pmap-c)# no loadbalance vip advertise metric 30
```
Enabling a VIP to Reply to ICMP Requests

You can enable a VIP to reply to ICMP ECHO requests by using the `loadbalance vip icmp-reply` command in policy-map class configuration mode. For example, if a user sends an ICMP ECHO request to a VIP, this command instructs the VIP to send an ICMP ECHO-REPLY. The syntax of this command is as follows:

```
loadbalance vip icmp-reply [active [primary-inservice]]
```

The options are:

- **active**—(Optional) Instructs the ACE to reply to an ICMP request only if the configured VIP is active. If the VIP is not active and the `active` option is specified, the ACE discards the ICMP request and the request times out.

- **primary-inservice**—Instructs the ACE to reply to an ICMP ping only if the primary server farm state is UP, regardless of the state of the backup server farm. If this option is enabled and the primary server farm state is DOWN, the ACE discards the ICMP request and the request times out.

---

**Note**

The `loadbalance vip icmp-reply` command controls a ping to a VIP on the ACE. This command implicitly downloads an ICMP access control list entry for the VIP. When you configure this command on the ACE, any configured ACLs that deny ICMP traffic have no effect on a client’s ability to ping the VIP.

To complete the configuration when you configure the `active` option of this command, be sure to configure a Telnet probe and associate it with the server farm. The probe monitors the health of all the real servers in the server farm and ensures that the VIP responds with an ICMP ECHO REPLY only if the server port is active. If the server port is down or unreachable, the probe fails and the VIP stops responding to the ECHO request. For details about configuring probes, see Chapter 4, Configuring Health Monitoring.

---

**Note**

For security reasons, the ACE does not allow pings from an interface on a VLAN on one side of the ACE through the module to an interface on a different VLAN on the other side of the module. For example, you cannot ping a VIP from a server if the VIP is on a VLAN that is different from the server VLAN.
For example, enter:

```
host1/Admin(config)# policy-map multi-match L4SLBPOLICY
host1/Admin(config-pmap)# class FILTERHTTP
host1/Admin(config-pmap-c)# loadbalance vip icmp-reply active
```

To view the current states of the `loadbalance vip icmp-reply` command, the primary server farm, and the backup server farm, use the `show service-policy policy-name detail` command. For details, see the “Displaying Service-Policy Statistics” section.

### Enabling Per-Packet Load Balancing for UDP Traffic

By default, the ACE can load balance UDP DNS address (A-record) request packets using the same tuple to the same real server on an existing connection. You can close the connection immediately after a DNS response is sent back to the client, enabling per-packet load balancing for this traffic, by using the `loadbalance vip udp-fast-age` command in policy-map class configuration mode. The syntax of this command is as follows:

```
loadbalance vip udp-fast-age
```

You must configure this command under a VIP class under a Layer-4 policy. When you use this command, the ACE load balances all new DNS A-record requests to a new real server in the server farm according to the predictor algorithm. All retransmitted UDP DNS A-record packets from the client go to the same real server. This command is only applicable for UDP DNS A-record flows. It also works with fragmented packets of this type. The ACE performs the reassembly and then forwards the response, and deletes the connection record.

**Note**

You can use the `loadbalance vip udp-fast-age` command with a generic Layer-7 policy only. For more information on configuring the generic Layer-7 policy, see the “Configuring a Layer 7 Policy Map for SLB” section.

For example, enter:

```
host1/Admin(config-pmap-c)# loadbalance vip udp-fast-age
```

To reset the default behavior, use the `no loadbalance vip udp-fast-age` command. For example, enter:

```
host1/Admin(config-pmap-c)# no loadbalance vip udp-fast-age
```
The following example provides a running configuration using the `loadbalance vip udp-fast-age` command:

```
access-list ACL1 line 10 extended permit ip any any

rserver host RS1
  ip address 10.6.252.245
  inservice
rserver host RS2
  ip address 10.6.252.246
  inservice
serverfarm host SF1
  rserver RS1
    inservice
  rserver RS2
    inservice

class-map match-any GENERIC_VIP
  2 match virtual-address 10.6.252.19 udp any

policy-map type loadbalance generic first-match GENERIC_LB1
  class class-default
    serverfarm SF1

policy-map multi-match LB
  class GENERIC_VIP
    loadbalance vip udp-fast-age
    loadbalance vip inservice
    loadbalance policy GENERIC_LB1

interface vlan 10
  ip address 10.6.252.12 255.255.255.0
  access-group input TEST
  service-policy input LB
  no shutdown
```

### Enabling a VIP

You can enable a VIP for SLB operations by using the `loadbalance vip inservice` command in policy-map class configuration mode. The syntax of this command is as follows:

```
loadbalance vip inservice
```
The following example specifies the `loadbalance vip inservice` command as an action for the SLB policy map:

```
host1/Admin(config)# policy-map multi-match L4SLBPOLICY
host1/Admin(config-pmap)# class FILTERHTTP
host1/Admin(config-pmap-c)# loadbalance vip inservice
```

To disable a VIP, enter:

```
host1/Admin(config-pmap-c)# no loadbalance vip inservice
```

### Applying a Layer 3 and Layer 4 Policy to an Interface

Use the `service-policy` command to do the following:

- Apply a previously created policy map.
- Attach the traffic policy to a specific VLAN interface or globally to all VLAN interfaces in the same context.

The `service-policy` command is available at both the interface configuration mode and at the configuration mode. Specifying a policy map in the configuration mode applies the policy to all of the VLAN interfaces associated with a context.

The syntax of this command is as follows:

```
service-policy input policy_name
```

The keywords, arguments, and options are as follows:

- **input**—Specifies that the traffic policy is to be attached to the input direction of an interface. The traffic policy evaluates all traffic received by that interface.
- **policy_name**—Name of a previously defined policy map, configured with a previously created `policy-map` command. The name can be a maximum of 64 alphanumeric characters.

Follow these guidelines when creating a service policy:

- Policy maps, applied globally in a context, are internally applied on all interfaces associated with the context.
- You can apply the policy in an input direction only.
- A policy activated on an interface overwrites any specified global policies for overlapping classification and actions.
• The ACE allows only one policy of a specific feature type to be activated on a given interface.

• You can apply a maximum of 128 service policies on each interface.

When you configure multiple VIPs on an interface, the match criteria for incoming traffic follow the order in which you configure the service-policy statements on that interface. Each service policy that you configure on an interface applies a Layer 3 and Layer 4 multi-match policy map to the interface. Each multi-match policy map may contain one or more features as defined in the class maps associated with the policy map.

Because service policies do not have line numbers, the order in which you configure them on an interface is extremely important. The reason is that the ACE has an implicit feature lookup order as follows:

1. Access control (permit or deny a packet)
2. Management traffic
3. TCP normalization and connection parameters
4. Server load balancing
5. Application protocol inspection
6. Source NAT
7. Destination NAT

When you apply multiple service policies to an interface, the ACE appends the last service policy at the end of the list. If you need to change the order of the service policies on an interface, you must first remove the service policies and then add them back in the appropriate order. This process is disruptive to the network.

As an alternative to reordering the service policies, you can configure multiple class maps in the same multi-match policy map, where you can define the order of the class maps. This process is not disruptive to the network. When you add new class maps to an existing policy, use the `insert-before` command to place the new class map in the desired order.
The following example shows how to configure two class maps in a policy map, where the VIP-ACCESS-MANAGER-80 is the more specific class map. To ensure that the ACE matches traffic to the more specific classification, configure VIP-ACCESS-MANAGER-80 class map first under the LB-TRAFFIC policy map. This example and the one that follows include only the Layer 3 and Layer 4 traffic classification portions of the configuration.

class-map match-all VIP-ACCESS-MANAGER-ANY
  2 match virtual-address 10.238.45.200 tcp eq any

class-map match-all VIP-ACCESS-MANAGER-80
  2 match virtual-address 10.238.45.200 tcp eq www

policy-map multi-match LB-TRAFFIC
  class VIP-ACCESS-MANAGER-80
    loadbalance vip inservice
    loadbalance policy POLICY-ACCESS-MANAGER-80
    loadbalance vip icmp-reply active
  class VIP-ACCESS-MANAGER-ANY
    loadbalance vip inservice
    loadbalance policy POLICY-ACCESS-MANAGER-ANY
    loadbalance vip icmp-reply active

interface vlan 758
  description CLIENT-SIDE-VLAN
  bridge-group 100
  access-group input ALL
  service-policy input LB-TRAFFIC
  no shutdown

The following example shows how to use two policy maps, one for each class map, and achieve the same results as in the previous example. In this example, you configure the more specific policy map first under the interface using a service policy.

policy-map multi-match LB-TRAFFIC-80
  class VIP-ACCESS-MANAGER-80
    loadbalance vip inservice
    loadbalance policy POLICY-ACCESS-MANAGER-80
    loadbalance vip icmp-reply active

policy-map multi-match LB-TRAFFIC-ANY
  class VIP-ACCESS-MANAGER-ANY
    loadbalance vip inservice
    loadbalance policy POLICY-ACCESS-MANAGER-ANY
    loadbalance vip icmp-reply active
interface vlan 758
  description CLIENT-SIDE-VLAN
  bridge-group 100
  access-group input ALL
  service-policy input LB-TRAFFIC-80
  service-policy input LB-TRAFFIC-ANY
  no shutdown

The following example specifies an interface VLAN and applies a Layer 3 and Layer 4 SLB policy map to the VLAN:

host1/Admin(config)# interface vlan 50
host1/Admin(config-if)# mtu 1500
host1/Admin(config-if)# ip address 172.20.1.100 255.255.0.0
host1/Admin(config-if)# service-policy input L4SLBPOLICY

To apply the Layer 3 and Layer 4 SLB policy map to all interfaces in the context, enter:

host1/Admin(config)# service-policy input L4SLBPOLICY

To detach a traffic policy from an interface, enter:

host1/Admin(config-if)# no service-policy input L4SLBPOLICY

To globally detach a traffic policy from a context, enter:

host1/Admin(config)# no service-policy input L4SLBPOLICY

When you detach a traffic policy either individually from the last VLAN interface on which you applied the service policy or globally from all VLAN interfaces in the same context, the ACE automatically resets the associated service-policy statistics. The ACE performs this action to provide a new starting point for the service-policy statistics the next time that you attach a traffic policy to a specific VLAN interface or globally to all VLAN interfaces in the same context.
Configuring UDP Booster

If you have a network application that requires very high UDP connection rates, configure the UDP booster feature. This feature is well suited to applications where statistical load-balancing algorithms are adequate. The most common application of this feature is DNS load balancing, but you can use it wherever you require very high UDP connection rates.

To achieve the very high connection rates with this feature, the ACE load balances new UDP requests. Then, it attempts to match subsequent UDP packets that hit the appropriate interface to a previous load-balancing result. Subsequent packets that match previous load-balancing results likely will have source IP addresses different from the loadbalanced packets. The ACE load balances UDP misses normally using the configured load balancing predictor for the server farm.

The ACE keeps track of hash hits in a special-purpose connection table. The hash range is 1 to 16384 (16 K). A full table contains 65,536 (64 K) entries, based on one client and one server for each entry, and each of the two network processors has its own independent table.

For client-side interface hits, the ACE NATs the VIP to the associated server IP address, while preserving the source IP address. For return traffic from the server, the ACE NATs the server IP address to the VIP, while preserving the destination address.

For requests where the source and destination ports are the same (common for DNS applications with a source and a destination port number of 53), the ACE translates the source port using implicit PAT. The ACE reserves the port range of 1 to 1023 for the source port hash. Therefore, incoming UDP traffic must either have identical source and destination ports or the source port must be greater than 1023. Otherwise, the feature does not work properly.

---

**Note**

Do not configure UDP booster with NAT or with any Layer 7 feature such as per-packet UDP load balancing (also called UDP fast-age). Otherwise, unexpected results may occur. For details about per-packet UDP load balancing, see the “Enabling Per-Packet Load Balancing for UDP Traffic” section. For details about NAT, see the Cisco Application Control Engine Module Security Configuration Guide.
To configure the UDP booster feature, use the `udp` command in interface configuration mode. The syntax of this command is as follows:

```
udp {ip-source-hash | ip-destination-hash}
```

The keywords are as follows:

- **ip-source-hash**—Instructs the ACE to hash the source IP address of UDP packets that hit a source-hash VLAN interface before performing a connection match. Configure this keyword on a client-side interface.

- **ip-destination-hash**—Instructs the ACE to hash the destination IP address of UDP packets that hit a destination-hash VLAN interface before performing a connection match. Configure this keyword on a server-side interface.

---

**Note**

To use this feature, you must configure both keywords on the appropriate interfaces, and configure standard load balancing on the ACE. Be aware that when you configure the UDP booster feature, the ACE drops all traffic in the opposite direction (for example, UDP connections initiated from real servers).

For details on configuring load balancing, see the chapters in this guide.

For example, to configure this feature on client-side VLAN interface 101 and on server-side VLAN interface 201, enter:

```
host1/Admin(config)# interface vlan 101
host1/Admin(config-if)# udp ip-source-hash
host1/Admin(config-if)# exit
host1/Admin(config)# interface vlan 201
host1/Admin(config-if)# udp ip-destination-hash
```

To remove the UDP booster feature from the above-mentioned interfaces, enter:

```
host1/Admin(config)# interface vlan 101
host1/Admin(config-if)# no udp ip-source-hash
host1/Admin(config-if)# exit
host1/Admin(config)# interface vlan 201
host1/Admin(config-if)# no udp ip-destination-hash
```
Configuring the ACE to Perform Hashing When the Source and Destination Ports Are Equal

By default, when the source and destination ports of a TCP or UDP packet are equal, the classification and distribution engine (CDE) uses the source IP address and destination IP address to perform the hash function. When they are not equal, the CDE only uses the ports.

To configure the CDE to perform the hash function using the source and destination ports when the TCP or UDP packets are equal, use the `hw-module cde-same-port-hash` command. When you configure this command, the ACE also disables implicit PAT on packets so that the source port does not change. This command is available only in the Admin context.

When this command is configured and the ports are equal, the CDE uses a slightly different hash method from the default method.

For example, enter:

```
switch/Admin(config)# hw-module cde-same-port-hash
```

To reset the default behavior, enter:

```
switch/Admin(config)# no hw-module cde-same-port-hash
```

Configuring RDP Load Balancing

The Microsoft Remote Desktop Protocol (RDP) provides users with remote display and input capabilities over network connections for Windows-based applications running on a terminal server. One of the key features supported by RDP is called roaming disconnect. With roaming disconnect, you can manually disconnect from a terminal server session without logging off the session. When you later log on to the system using either the same device or a different device, you are automatically reconnected to your disconnected session. Also, when your session is unexpectedly terminated by a network or client failure, you are disconnected but not logged off.
In a load-balancing configuration, the ACE distributes incoming session connections across the terminal servers in a server farm according to the load-balancing method configured on the server farm. The session directory (SD) keeps a list of user sessions indexed by username and allows you to reconnect to the terminal server where your disconnected session resides and to resume that session. When you authenticate yourself with a terminal server in the server farm, the SD is queried with your username. If a session with your username exists on one of the terminal servers, the SD redirects you to that terminal server. This feature allows you to disconnect a session with applications running, whether intentionally or because of a network failure, and then reconnect at a later time to the same session with the same applications running. The SD passes to the client a routing token with login information and the server IP address embedded in it.

To parse the routing token and to use the IP address embedded inside it to load balance client sessions to terminal servers, the ACE needs to look inside the RDP packet. The ACE makes the load-balancing decision based on the presence or absence of the routing token in the RDP packet. If the token is present, the ACE forwards the packet to the server that has an address and port embedded in the token. If the token is not present, the ACE uses the Layer 3 and Layer 4 load-balancing configuration to determine the real server.

---

**Note**

The ACE supports RDP load balancing based on routing tokens. If the client does not send the routing token in the request to the ACE, the ACE load balances the client request to one of the available servers in the server farm using the configured predictor.
This section contains the following topics on configuring the ACE for RDP load balancing:

- Configuring Real Servers and a Server Farm
- Configuring a Layer 7 RDP Load-Balancing Policy
- Configuring a Layer 3 and Layer 4 RDP Policy
- Applying the Layer 3 and Layer 4 RDP Policy to an Interface
- Example of an RDP Load-Balancing Configuration

## Configuring Real Servers and a Server Farm

For information about configuring real servers and server farms, see Chapter 2, Configuring Real Servers and Server Farms.

## Configuring a Layer 7 RDP Load-Balancing Policy

To configure a Layer 7 RDP load-balancing policy, perform the following steps:

1. Create an RDP Layer 7 load-balancing policy map.
   ```
   (config)# policy-map type loadbalance rdp first-match RDP_L7_POLICY
   host1/Admin(config-pmap-lb-rdp)#
   ```

2. Associate the default class map with the policy map. The default class map is the only class map that you can associate with the Layer 7 RDP policy map.
   ```
   host1/Admin(config-pmap-lb-rdp)# class class-default
   host1/Admin(config-pmap-lb-rdp-C)#
   ```

3. Associate the server farm with the policy map.
   ```
   host1/Admin(config-pmap-lb-rdp-C)# serverfarm sf1
   ```

For more details about configuring a Layer 7 load-balancing policy, see the “Configuring a Layer 7 Class Map for SLB” section and the “Configuring a Layer 7 Policy Map for SLB” section.
Configuring a Layer 3 and Layer 4 RDP Policy

To configure a Layer 3 and Layer 4 RDP policy, perform the following steps:

**Step 1** Create a Layer 3 and Layer 4 class map for RDP and specify a VIP match statement. The `rdp` keyword in the match statement sets the port to the default RDP port of 3389. You can also enter the port number directly.

```plaintext
host1/Admin(config)# class-map match-any RDP_L4_CLASS
host1/Admin(config-cmap)# match virtual-address 192.168.12.15 tcp port eq rdp
```

**Step 2** Create a Layer 3 and Layer 4 policy map and associate the Layer 3 and Layer 4 class map with it.

```plaintext
host1/Admin(config)# policy-map multi-match RDP_L4_POLICY
host1/Admin(config-pmap)# class RDP_L4_CLASS
host1/Admin(config-pmap-c)#
```

**Step 3** Place the VIP inservice.

```plaintext
host1/Admin(config-pmap-c)# loadbalance vip inservice
```

**Step 4** Associate the Layer 7 policy map that you created in the “Configuring a Layer 7 RDP Load-Balancing Policy” section with the Layer 3 and Layer 4 policy map.

```plaintext
host1/Admin(config-pmap-c)# loadbalance policy RDP_L7_POLICY
```

For more details about configuring a Layer 3 and Layer 4 LB policy, see the “Configuring a Layer 3 and Layer 4 Class Map for SLB” and the “Configuring a Layer 3 and Layer 4 Policy Map for SLB” sections.

Applying the Layer 3 and Layer 4 RDP Policy to an Interface

For information about applying a Layer 3 and Layer 4 load-balancing policy to an interface, see the “Applying a Layer 3 and Layer 4 Policy to an Interface” section.
Example of an RDP Load-Balancing Configuration

The following example provides a running configuration for RDP load balancing:

```
access-list ACL1 line 10 extended permit ip any any

rserver host RS1
   ip address 10.6.252.245
   inservice
rserver host RS2
   ip address 10.6.252.246
   inservice

serverfarm host SF1
   rserver RS1
       inservice
   rserver RS2
       inservice

class-map match-any RDP_L4_CLASS
   2 match virtual-address 10.6.252.19 tcp eq rdp

policy-map type loadbalance rdp first-match RDP_L7_POLICY
   class class-default
      serverfarm SF1

policy-map multi-match RDP_L4_POLICY
   class RDP_L4_CLASS
      loadbalance vip inservice
      loadbalance RDP_L7_POLICY

interface vlan 10
   ip address 10.6.252.12 255.255.255.0
   access-group input ACL1
   service-policy input RDP_L4_POLICY
   no shutdown
```
Configuring RADIUS Load Balancing

The ACE uses the RADIUS protocol to load balance client requests. The ACE processes RADIUS client requests for accounting and server access by using the following:

- Layer 3 and Layer 4 load balancing to determine which server will process the first packet
- Stickiness to load balance to the same server subsequent packets from the same client based on a specific RADIUS field in those packets

If the ACE receives any retransmitted requests, it sends them to the same server where it sent the original request. The module uses the source and destination addresses and ports and a header field called “identifier” in the RADIUS header to identify a retransmission.

To ensure that the ACE forwards client data frames to the same RADIUS server where the RADIUS requests were sent, you need to configure a Layer 3 and Layer 4 load-balancing policy with a catch-all VIP and a RADIUS sticky group as the action. Upon receiving the framed IP address in the Access-Accept or the Accounting-Request message, the ACE forwards the data frames to the appropriate server.

In a load-balanced service gateway environment, the ACE supports partitioned subscriber databases across data centers by mapping the defined calling station ID or username ranges to a specific server farm. This feature allows you to partition your subscriber database so that each server farm will have a limited set of subscribers to service. You can specify subscriber attributes by configuring a RADIUS class map with match criteria set to the subscriber attributes.

The ACE does not load balance RADIUS accounting on/off messages. Instead, it replicates those messages to each real server in the server farm that is configured in the RADIUS LB policy.

This section contains the following topics on configuring RADIUS load balancing:

- Configuring Real Servers and a Server Farm
- Configuring a RADIUS Sticky Group
- Configuring a Layer 7 RADIUS Load-Balancing Policy
- Configuring a Layer 3 and Layer 4 RADIUS Load-Balancing Policy
- Configuring a Traffic Policy for Non-RADIUS Data Forwarding
• Applying a Layer 3 and Layer 4 RADIUS Policy to an Interface
• Examples of RADIUS Load-Balancing Configurations

Configuring Real Servers and a Server Farm

For information about configuring real servers and server farms, see Chapter 2, Configuring Real Servers and Server Farms.

Configuring a RADIUS Sticky Group

You can configure a RADIUS sticky group based on one of the following criteria:

• Framed IP only
• Framed IP and calling station ID
• Framed IP and username

To configure a RADIUS sticky group, perform the following steps:

Step 1  Create a RADIUS sticky group.

```bash
host1/admin(config)# sticky radius framed-ip calling-station-id
RADIUS_GROUP
host1/admin(config-sticky-radius)#
```

The RADIUS sticky group ensures that the ACE forwards subsequent messages that belong to the same user session (identified by username, calling ID, or framed IP) to the same server as the first message in that user session. Multiple sessions can come from the same client.

Step 2  Associate a server farm with the sticky group.

```bash
host1/admin(config-sticky-radius)# serverfarm sf2
```

For more information about configuring RADIUS stickiness, see Chapter 5, Configuring Stickiness.
Configuring a Layer 7 RADIUS Load-Balancing Policy

To configure a Layer 7 RADIUS load-balancing policy, perform the following steps:

**Step 1** Create a Layer 7 load-balancing class map.

```bash
host1/Admin(config)# class-map type radius loadbalance match-any RADIUS_L7_CLASS
host1/Admin(config-cmap-radius-lb)#
```

You can also omit the first two steps and use the default class map (class-default) instead. If you use the default class map, you cannot specify match criteria for the RADIUS calling station ID or username.

**Note** A match-all class map cannot have more than one of the same type of match. A match-any class map cannot have more than one of a different type of match.

**Step 2** Define match criteria for the class map.

```bash
host1/Admin(config-cmap-radius-lb)# match radius attribute calling-station-id 122
host1/Admin(config-cmap-radius-lb)# match radius attribute username JSMITH
```

**Step 3** Create a Layer 7 RADIUS load-balancing policy map.

```bash
host1/Admin(config)# policy-map type loadbalance radius first-match RADIUS_L7_POLICY
host1/Admin(config-pmap-lb-radius)#
```

**Step 4** Associate the class map that you created in Step 1 with the policy map that you created in Step 3.

```bash
host1/Admin(config-pmap-lb-radius)# class RADIUS_L7_CLASS
host1/Admin(config-pmap-lb-radius-c)#
```

**Step 5** Associate the sticky group with the policy map.

```bash
host1/Admin(config-pmap-lb-radius-c)# sticky-serverfarm RADIUS_GROUP
```
You can specify a simple (nonsticky) server farm in the Layer 7 policy instead of a sticky server farm. If you do, the only features available would be the forwarding of retransmissions to the same real server and replication of accounting on or off to all real servers.

For more details about configuring a Layer 7 load-balancing policy, see the “Configuring a Layer 7 Class Map for SLB” section and the “Configuring a Layer 7 Policy Map for SLB” section.

### Configuring a Layer 3 and Layer 4 RADIUS Load-Balancing Policy

To configure a Layer 3 and Layer 4 RADIUS policy, perform the following steps:

**Step 1** Configure a Layer 3 and Layer 4 class map for RADIUS load balancing and specify a VIP match statement. The `radius-auth` keyword in the match statement sets the port to the default RADIUS port of 1812 and the `radius-acct` keyword sets the port to the default RADIUS port or 1813. You can also enter the port numbers directly.

```
host1/Admin(config)# class-map match-any RADIUS_L4_CLASS
host1/Admin(config-cmap)# match virtual-address 192.168.12.15 udp eq radius-auth
host1/Admin(config-cmap)# match virtual-address 192.168.12.15 udp eq radius-acct
```

You can also use a single match statement with a range of ports as follows:

```
host1/Admin(config-cmap)# match virtual-address 192.168.12.15 udp range 1812 1813
```

**Step 2** Configure a Layer 3 and Layer 4 RADIUS policy map and associate the Layer 3 and Layer 4 RADIUS class map that you created in Step 1 with it.

```
host1/Admin(config)# policy-map multi-match RADIUS_L4_POLICY
host1/Admin(config-pmap)# class RADIUS_L4_CLASS
host1/Admin(config-pmap-c)#
```

**Step 3** Place the VIP inservice.

```
host1/Admin(config-pmap-c)# loadbalance vip inservice
```
Configuring Traffic Policies for Server Load Balancing

Configuring RADIUS Load Balancing

Step 4
Associate the Layer 7 policy map that you created in the “Configuring a Layer 7 RADIUS Load-Balancing Policy” section with the Layer 3 and Layer 4 policy map.

```
host1/Admin(config-pmap-c)# loadbalance policy RADIUS_L7_POLICY
```

For more details about configuring a Layer 3 and Layer 4 LB policy, see the “Configuring a Layer 3 and Layer 4 Class Map for SLB” and the “Configuring a Layer 3 and Layer 4 Policy Map for SLB” sections.

For examples of RADIUS load-balancing policies, see the “Examples of RADIUS Load-Balancing Configurations” section.

Configuring a Traffic Policy for Non-RADIUS Data Forwarding

You can configure the ACE to forward non-RADIUS data packets from a client to the same server where the ACE sent the RADIUS packets from the same client. This feature is used in mobile wireless implementations with Cisco Services Selection Gateways (SSGs). The end user traffic passes through the load balancer and must be forwarded to the same real server (SSG) that authenticated or accounted for the user identified by the framed IP.

When the ACE needs to forward a non-RADIUS packet using framed-IP stickiness, it uses the source IP address of the packet to look up the framed IP address in the sticky database. If the lookup succeeds, the ACE sends the packet to the appropriate real server IP address as the next hop (the destination IP address of the packet does not change). If the lookup fails, the ACE routes the packet.

**Note**
This procedure includes the use of a catch-all VIP (IP address and netmask of 0.0.0.0 0.0.0.0). Use this type of VIP very carefully. Whenever possible, use a VIP address and netmask that are as close as possible to the actual traffic that the ACE needs to forward.
The following procedure contains the steps necessary to configure those parts of the configuration that pertain to the data forwarding process. Use this procedure with the previous three configuration procedures.

**Step 1** Create a Layer 4 class map with a catch-all VIP.

```
host1/Admin(config)# class-map match-any LAYER4_CLASS
host1/Admin(config-cmap)# 4 match virtual-address 0.0.0.0 0.0.0.0 any
host1/Admin(config-cmap)# exit
```

**Step 2** Create a simple Layer 7 load-balancing policy map.

```
host1/Admin(config)# policy-map type loadbalance first-match DATA_FORWARD_L7POLICY
host1/Admin(config-pmap-lb)#
```

**Step 3** Associate the default class map with the Layer 7 policy map to match any non-RADIUS incoming traffic. The default class map is the only class map that you can configure under this policy map.

```
host1/Admin(config-pmap-lb)# class class-default
host1/Admin(config-pmap-lb-c)#
```

**Step 4** Associate the RADIUS sticky group that you created in the “Configuring a RADIUS Sticky Group” section with the Layer 7 policy map.

```
host1/Admin(config-pmap-lb-c)# sticky-serverfarm RADIUS_GROUP
```

**Step 5** Associate the Layer 4 class map with the Layer 4 multimatch policy map that you created in the “Configuring a Layer 3 and Layer 4 RADIUS Load-Balancing Policy” section.

```
host1/Admin(config)# policy-map multi-match RADIUS_L4_POLICY
host1/Admin(config-pmap)# class LAYER4_CLASS
```

**Step 6** Enable the catch-all VIP that you configured in Step 1.

```
host1/Admin(config-pmap-c)# loadbalance vip inservice
```

**Step 7** Associate the Layer 7 load-balancing policy that you created in Step 2 with the Layer 4 multimatch policy.

```
host1/Admin(config-pmap-c)# loadbalance policy DATA_FORWARD_L7POLICY
```
For an example of the entire configuration for a data forwarding RADIUS traffic policy, see the “End User Data Forwarding Policy” section.

### Applying a Layer 3 and Layer 4 RADIUS Policy to an Interface

For information about applying a Layer 3 and Layer 4 load balancing policy to an interface, see the “Applying a Layer 3 and Layer 4 Policy to an Interface” section.

### Examples of RADIUS Load-Balancing Configurations

The following configuration examples provide RADIUS load-balancing configurations with and without a Layer 7 RADIUS class map:

- **Without a Layer 7 RADIUS Class Map**
- **With a Layer 7 RADIUS Class Map**
- **End User Data Forwarding Policy**

#### Without a Layer 7 RADIUS Class Map

The following configuration example shows the running configuration for RADIUS load-balancing without a Layer 7 RADIUS class map. The parts of the configuration that pertain strictly to the RADIUS load-balancing configuration are shown in bold text.

```plaintext
access-list ACL1 line 10 extended permit ip any any
rserver host RS1
  ip address 10.6.252.245
  inservice
rserver host RS2
  ip address 10.6.252.246
  inservice
serverfarm host SF1
  rserver RS1
    inservice
  rserver RS2
    inservice

sticky radius framed-ip calling-station-id RADIUS_GROUP
  serverfarm SF1
```

With a Layer 7 RADIUS Class Map

The following configuration example shows the running configuration for RADIUS load-balancing with a Layer 7 RADIUS class map. The parts of the configuration that pertain strictly to the RADIUS load-balancing configuration are shown in bold text.

```
access-list ACL1 line 10 extended permit ip any any

rserver host RS1
  ip address 10.6.252.245
  inservice
rserver host RS2
  ip address 10.6.252.246
  inservice

serverfarm host SF1
  rserver RS1
    inservice
  rserver RS2
    inservice

sticky radius framed-ip calling-station-id RADIUS_GROUP1
  serverfarm SF1

sticky radius framed-ip calling-station-id RADIUS_GROUP2
  serverfarm SF2

class-map match-any RADIUS_L4_CLASS
  2 match virtual-address 192.168.12.15 udp range 1812 1813
```

```
class-map type radius loadbalance match-any RADIUS_L7_CLASS1
  2 match radius attribute calling-station-id 122*
  3 match radius attribute calling-station-id 133*

class-map type radius loadbalance match-any RADIUS_L7_CLASS2
  2 match radius attribute calling-station-id 144*

policy-map type loadbalance radius first-match RADIUS_L7_POLICY
  class RADIUS_L7_CLASS1
    sticky-serverfarm RADIUS_GROUP1
  class RADIUS_L7_CLASS2
    sticky-serverfarm RADIUS_GROUP2

policy-map multi-match RADIUS_L4_POLICY
  class RADIUS_L4_CLASS
    loadbalance vip inservice
    loadbalance RADIUS_L7_POLICY

interface vlan 10
  ip address 192.168.12.12 255.255.255.0
  service-policy input RADIUS_L4_POLICY
  no shutdown

**End User Data Forwarding Policy**

The following configuration example provides the commands necessary to instruct the ACE to forward non-RADIUS data packets to the same RADIUS server where the ACE sent the first RADIUS packet. The parts of the configuration that pertain strictly to the data forwarding policy are shown in bold text.

access-list ACL1 line 10 extended permit ip any any

rserver host RS1
  ip address 10.6.252.245
  inservice
rserver host RS2
  ip address 10.6.252.246
  inservice

serverfarm host SF1
  rserver RS1
    inservice
  rserver RS2
    inservice
sticky radius framed-ip RADIUS_GROUP1
   serverfarm SF1

class-map type radius loadbalance match-any RADIUS_L7_CLASS
   2 match radius attribute calling-station-id 133*

class-map match-any RADIUS_L4_CLASS
   3 match virtual-address 192.168.12.15 udp range 1812 1813

class-map match-any LAYER4_CLASS
   4 match virtual-address 0.0.0.0 0.0.0.0 any

policy-map type loadbalance radius first-match RADIUS_L7_POLICY
   class RADIUS_L7_CLASS
      sticky-serverfarm RADIUS_GROUP1

policy-map type loadbalance first-match DATA_FORWARD_L7POLICY
   class class-default
      sticky-serverfarm RADIUS_GROUP1

policy-map multi-match RADIUS_L4_POLICY
   class RADIUS_L4_CLASS
      loadbalance vip inservice
      loadbalance policy RADIUS_L7_POLICY
      loadbalance vip icmp-reply
      loadbalance vip advertise
   class LAYER4_CLASS
      loadbalance vip inservice
      loadbalance policy LAYER7_POLICY

interface vlan 10
   ip address 192.168.12.12 255.255.255.0
   service-policy input RADIUS_L4_POLICY
   no shutdown
Configuring RTSP Load Balancing

The Real-Time Streaming Protocol (RTSP) is used for streaming audio, video, and simulation data from a server to a client by such applications as the following:

- Cisco IP/TV
- Apple QuickTime 4
- RealAudio
- RealNetworks
- RealPlayer

The ACE supports RTSP over TCP. The ACE bases Layer 3 and Layer 4 RTSP load-balancing decisions on the configured VIP and RTSP port. For Layer 7 load balancing, the ACE uses the RTSP URL or RTSP header. The format of the RTSP URL is rtsp://cisco.com/video.

An RTSP session can have multiple request and response message exchanges over the same or different connections. For example, when a media player tries to play a media file, the following message exchange may occur:

- OPTIONS query the server for the supported features (some players skip this).
- DESCRIBE retrieves the description of a media resource.
- SETUP specifies the transport mechanism to be used for the media.
- PLAY tells the server to start sending streamed data via the mechanism specified in SETUP.
- PAUSE causes the stream delivery to be interrupted temporarily.
- TEARDOWN stops the stream delivery and frees the resources.

When the ACE receives the first request message (usually OPTIONS or DESCRIBE) for a new RTSP session, it makes a load-balancing decision based on the configured policy and forwards the message to the selected server. The ACE forwards all subsequent messages in this session to the same server. RTSP messages are text based and similar to HTTP.
RTSP sticky uses the Session header to stick a client to a server. The Session header contains a session ID assigned by the server in the SETUP response. When the server responds to the SETUP request, the ACE creates an entry in the sticky database with the session ID from the server. All subsequent message exchanges will contain the same Session header.

**Note**
If you expect all RTSP requests for the same session to arrive over a single connection, the sticky part of the configuration is optional.

The inspection part of the configuration is optional. However, to support RTSP data traffic running on a separate connection over Real-Time Transport Protocol (RTP), you must configure RTSP inspection. If you configure inspection, the ACE performs an inspection and fixes the RTSP packets, including any necessary rewriting of the packet and opening of the restricted ports. For information about RTSP inspection, see the *Cisco Application Control Engine Module Security Configuration Guide*.

This section contains the following topics on configuring RTSP load balancing:
- Configuring Real Servers and a Server Farm
- Configuring an RTSP Sticky Group
- Configuring a Layer 7 RTSP Load-Balancing Policy
- Configuring a Layer 3 and Layer 4 RTSP Load-Balancing Policy
- Applying a Layer 3 and Layer 4 RTSP Policy to an Interface
- Example of an RTSP Load-Balancing Configuration

### Configuring Real Servers and a Server Farm

For information about configuring real servers and server farms, see Chapter 2, *Configuring Real Servers and Server Farms*. 
Configuring an RTSP Sticky Group

This section describes how to configure RTSP stickiness. It is an optional procedure and is not required for RTSP load balancing.

To configure an RTSP sticky group, perform the following steps:

**Step 1** Create an RTSP sticky group. The RTSP sticky group ensures that the ACE sends subsequent requests for the same session to the same server as the first request for that session.

```
host1/admin(config)# sticky rtsp-header Session RTSP_GROUP
host1/admin(config-sticky-rtsp)#
```

**Note** If you expect all RTSP requests for the same URL to arrive over a single connection, the sticky part of the configuration is optional.

```
host1/admin(config)# sticky rtsp-header Session RTSP_GROUP
host1/admin(config-sticky-rtsp)#
```

**Step 2** Associate a server farm with the sticky group.

```
host1/admin(config-sticky-rtsp)# serverfarm SF4
```

For more information about configuring stickiness, see Chapter 3, Configuring Stickiness.

Configuring a Layer 7 RTSP Load-Balancing Policy

To configure a Layer 7 RTSP load-balancing policy, perform the following steps:

**Step 1** Create a Layer 7 RTSP load-balancing class map. Y

```
host1/Admin(config)# class-map type rtsp loadbalance match-any RTSP_L7_CLASS
host1/Admin(config-cmap-rtsp-lb)#
```
Step 2  Define match criteria for the class map.

```bash
host1/Admin(config-cmap-rtsp-lb)# match rtsp url rtsp://cisco.com/movie
host1/Admin(config-cmap-rtsp-lb)# match rtsp header Accept header-value application/sdp
host1/Admin(config-cmap-rtsp-lb)# match source-address 192.168.12.15 255.255.255.0
```

Step 3  Create a Layer 7 RTSP load-balancing policy map.

```bash
host1/Admin(config)# policy-map type loadbalance rtsp first-match RTSP_L7_POLICY
host1/Admin(config-pmap-lb-rtsp)#
```

Step 4  Associate the class map that you created in Step 1 (or the default class) with the policy map that you created in Step 3.

```bash
host1/Admin(config-pmap-lb-rtsp)# class RTSP_L7_CLASS
host1/Admin(config-pmap-lb-rtsp-c)#
```

Step 5  Associate the sticky group with the policy map.

```bash
host1/Admin(config-pmap-lb-rtsp-c)# sticky-serverfarm RTSP_GROUP
```

For more details about configuring a Layer 7 load balancing policy, see the “Configuring a Layer 7 Class Map for SLB” section and the “Configuring a Layer 7 Policy Map for SLB” section.

**Configuring a Layer 3 and Layer 4 RTSP Load-Balancing Policy**

To configure a Layer 3 and Layer 4 RTSP load-balancing policy, perform the following steps:

Step 1  Configure a Layer 3 and Layer 4 class map for RTSP load balancing and specify a VIP match statement. The `rtsp` keyword in the match statement sets the port to the default RTSP port.

```bash
host1/Admin(config)# class-map match-all RTSP_L4_CLASS
host1/Admin(config-cmap)# match virtual-address 192.168.12.15 tcp eq rtsp
```
Step 2 Configure a Layer 3 and Layer 4 RTSP policy map and associate the Layer 3 and Layer 4 RTSP class map that you created in Step 1 with it.

```
host1/Admin(config)# policy-map multi-match RTSP_L4_POLICY
host1/Admin(config-pmap)# class RTSP_L4_CLASS
host1/Admin(config-pmap-c)#
```

Step 3 Place the VIP inservice.

```
host1/Admin(config-pmap-c)# loadbalance vip inservice
```

Step 4 Associate the Layer 7 policy map that you created in the “Configuring a Layer 7 RADIUS Load-Balancing Policy” section with the Layer 3 and Layer 4 policy map.

```
host1/Admin(config-pmap-c)# loadbalance policy RTSP_L7_POLICY
```

For more details about configuring a Layer 3 and Layer 4 load balancing policy, see the “Configuring a Layer 3 and Layer 4 Class Map for SLB” and the “Configuring a Layer 3 and Layer 4 Policy Map for SLB” sections.

### Applying a Layer 3 and Layer 4 RTSP Policy to an Interface

For information about applying a Layer 3 and Layer 4 load-balancing policy to an interface, see the “Applying a Layer 3 and Layer 4 Policy to an Interface” section.

### Example of an RTSP Load-Balancing Configuration

The following is a sample RTSP configuration. The sticky group and RTSP inspection portions are optional. If a client uses different connections for multiple requests in one session, you must configure the sticky group. If you use RTP for data traffic that runs on separate connections, an inspection is needed to open the proper pinholes.

```
access-list ACL1 line 10 extended permit ip any any
rserver host RS1
  ip address 10.6.252.245
  inservice
```
rserver host RS2
  ip address 10.6.252.246
  inservice

serverfarm host SF4
  rserver RS1
    inservice
  rserver RS2
    inservice

**sticky rtsp-header Session RTSP_GROUP**
  serverfarm SF4

class-map type rtsp loadbalance match-any RTSP_L7_CLASS
  match rtsp url rtsp://cisco.com/movie
  match rtsp header Accept header-value application/sdp

class-map match-all RTSP_L4_CLASS
  match virtual-address 192.168.12.15 tcp eq rtsp

policy-map type loadbalance rtsp first-match RTSP_L7_POLICY
  class RTSP_L7_CLASS
    sticky-serverfarm RTSP_GROUP

policy-map multi-match RTSP_L4_POLICY
  class RTSP_L4_CLASS
    loadbalance vip inservice
    loadbalance policy RTSP_L7_POLICY
    inspect rtsp

interface vlan 10
  ip address 192.168.12.12 255.255.255.0
  service-policy input RTSP_L4_POLICY
  no shutdown
Configuring SIP Load Balancing

The Session Initiation Protocol (SIP) functions as a signaling mechanism between user devices and media servers. SIP is a peer-to-peer protocol where end-devices (the user agent clients) initiate interactive communications sessions with SIP servers. These sessions can include Internet multimedia conferences, Internet telephone calls (Voice-over-IP), and multimedia distribution. Examples of client devices include hardware, software, handheld IP telephones, and personal digital assistants (PDAs). The ACE supports SIP over TCP or UDP.

The session Call-ID is a unique call identifier that resides in the text-based SIP messages sent from the client to the SIP proxy server (for example, a Cisco Softswitch placed between the clients and the ACE). A proxy server can reuse the same connection for multiple calls to the ACE simultaneously. The ACE independently load balances each call that has a different Call-ID, but keeps the back-end connections open at the same time. This behavior is different from HTTP persistence-rebalance where the ACE closes the existing connection after it makes the new load-balancing decision. For more information about persistence-rebalance, see the “Configuring HTTP Persistence Rebalance” section.

During a SIP session, the client and the server may exchange many messages. See Figure 3-2.
Figure 3-2  Example of a SIP Call Setup and Call Dialog

When the ACE receives the first request message (usually INVITE) from a client for a new SIP session, it makes a load-balancing decision based on the configured policy and forwards the message to the selected server. The ACE forwards all subsequent messages that have the same Call-ID in this session to the same server. Call-ID stickiness ensures that messages for a particular Call-ID from different TCP or UDP connections reach the correct servers. Stickiness by Call-ID is particularly important for stateful call services that use the Call-ID to identify current SIP sessions and make decisions based on the content of a message.

Note

If you expect all SIP requests for the same Call-ID to arrive over a single connection, then the sticky part of the configuration is optional.
If SIP stickiness is configured and the ACE finds the Call-ID in the header of the SIP messages sent from the client to the server, the ACE generates a key (hash value) based on the Call-ID. The ACE uses the key to look up an entry in the sticky table. If the entry exists, the ACE sends the client to the sticky server indicated by the table entry. If the entry does not exist, the ACE creates a new sticky entry, hashes the SIP Call-ID value into a key, and saves the key in the entry.

In most cases, you need to enable at least basic SIP inspection (configuring the `inspect sip` command without an inspection policy) for SIP load balancing to work. SIP inspection fixes the SIP packets, including any necessary rewriting of the packet and opening of restricted ports. These actions ensure that SIP traffic is routed properly through the ACE. For information about SIP inspection, see the Cisco Application Control Engine Module Security Configuration Guide.

This section contains the following topics on configuring SIP load balancing:

- Configuring Real Servers and a Server Farm
- Configuring a SIP Sticky Group
- Configuring a Layer 7 SIP Load-Balancing Policy
- Configuring a Layer 3 and Layer 4 SIP Load-Balancing Policy
- Applying a Layer 3 and Layer 4 SIP Policy to an Interface
- Example of a SIP Load-Balancing Configuration

**Configuring Real Servers and a Server Farm**

For information about configuring real servers and server farms, see Chapter 2, Configuring Real Servers and Server Farms.
Configuring a SIP Sticky Group

This section describes how to configure SIP stickiness. It is an optional procedure and is not required for SIP load balancing.

To configure a SIP sticky group, perform the following steps:

**Step 1** Create a SIP sticky group.

```
host1/admin(config)# sticky sip-header Call-ID SIP_GROUP
host1/admin(config-sticky-sip)#
```

The SIP sticky group ensures that the ACE sends subsequent packets from the same client to the same server as the first packet from that client.

**Note** If you expect all SIP requests for the same Call-ID to arrive over a single connection, the sticky part of the configuration is optional.

**Step 2** Associate a server farm with the sticky group.

```
host1/admin(config-sticky-sip)# serverfarm sf3
```

For more information about configuring SIP stickiness, see Chapter 3, Configuring Stickiness.

Configuring a Layer 7 SIP Load-Balancing Policy

To configure a Layer 7 SIP load-balancing policy, perform the following steps:

**Step 1** Create a Layer 7 SIP LB class map.

```
host1/Admin(config)# class-map type sip loadbalance match-any SIP_L7_CLASS
host1/Admin(config-cmap-sip-lb)#
```
You can also omit the first two steps and use the default class map (class-default) instead. If you use the default class map, you cannot specify match criteria for other SIP headers. Instead, the ACE performs the action specified under the class class-default command.

The class-default class map contains an implicit match-any statement that enables it to match all traffic. Additionally, the ACE ensures that messages with the same Call-ID are load balanced to the same server. For an example configuration, see the “SIP Load Balancing Without Match Criteria” section.

**Step 2** Define match criteria for the class map.

```
host1/Admin(config-cmap-sip-lb)# match sip header To header-value .*@cisco.com
```

**Step 3** Create a Layer 7 SIP load-balancing policy map.

```
host1/Admin(config)# policy-map type loadbalance sip first-match SIP_L7_POLICY
host1/Admin(config-pmap-lb-sip)#
```

**Step 4** Associate the class map that you created in Step 1 (or the default class) with the policy map that you created in Step 3.

```
host1/Admin(config-pmap-lb-sip)# class SIP_L7_CLASS
host1/Admin(config-pmap-lb-sip-c)#
```

**Step 5** Associate the sticky group with the policy map.

```
host1/Admin(config-pmap-lb-sip-C)# sticky-serverfarm SIP_GROUP
```

For more details about configuring a Layer 7 load-balancing policy, see the “Configuring a Layer 7 Class Map for SLB” section and the “Configuring a Layer 7 Policy Map for SLB” section.
Configuring a Layer 3 and Layer 4 SIP Load-Balancing Policy

To configure a Layer 3 and Layer 4 SIP load-balancing policy, perform the following steps:

**Step 1** Configure a Layer 3 and Layer 4 class map for SIP load balancing and specify a VIP match statement. The *sip* keyword in the match statement sets the port to the default SIP port of 5060.

```
host1/Adm(config)# class-map match-all SIP_L4_CLASS
host1/Adm(config-cmap)# match virtual-address 192.168.12.15 udp eq sip
```

**Step 2** Configure a Layer 3 and Layer 4 SIP policy map and associate the Layer 3 and Layer 4 SIP class map that you created in Step 1 with it.

```
host1/Adm(config)# policy-map multi-match SIP_L4_POLICY
host1/Adm(config-pmap)# class SIP_L4_CLASS
```

**Step 3** Place the VIP inservice.

```
host1/Adm(config-pmap-c)# loadbalance vip inservice
```

**Step 4** Associate the Layer 7 policy map that you created in the “Configuring a Layer 7 RADIUS Load-Balancing Policy” section with the Layer 3 and Layer 4 policy map.

```
host1/Adm(config-pmap-c)# loadbalance policy SIP_L7_POLICY
```

For more details about configuring a Layer 3 and Layer 4 load-balancing policy, see the “Configuring a Layer 3 and Layer 4 Class Map for SLB” and the “Configuring a Layer 3 and Layer 4 Policy Map for SLB” sections.

Applying a Layer 3 and Layer 4 SIP Policy to an Interface

For information about applying a Layer 3 and Layer 4 LB policy to an interface, see the “Applying a Layer 3 and Layer 4 Policy to an Interface” section.
Example of a SIP Load-Balancing Configuration

The following examples are from a sample SIP configuration:

- SIP Load Balancing Without Match Criteria
- SIP Load Balancing Based on SIP headers and SIP Inspection

The sticky group and SIP inspection are optional. If you do not configure class-map match criteria, the ACE performs the action specified under the `class class-default` command. The class-default class map contains an implicit match any statement that enables it to match all traffic. Additionally, the ACE ensures that messages with the same Call-ID are load balanced to the same server.

SIP Load Balancing Without Match Criteria

The following configuration example shows the running configuration for SIP load-balancing with match criteria. The parts of the configuration that pertain strictly to the SIP load-balancing configuration are shown in bold text.

access-list ACL1 line 10 extended permit ip any any

rserver host RS1
  ip address 10.6.252.245
  inservice
rserver host RS2
  ip address 10.6.252.246
  inservice
serverfarm host SF1
  rserver RS1
    inservice
  rserver RS2
    inservice

**sticky sip-header Call-ID SIP_GROUP**
  serverfarm SF3

class-map match-all SIP_L4_CLASS
  match virtual-address 192.168.12.15 udp eq sip

policy-map type loadbalance sip first-match SIP_L7_POLICY
  class class-default
    sticky-serverfarm SIP_GROUP
policy-map multi-match SIP_L4_POLICY
  class SIP_L4_CLASS
    loadbalance vip inservice
    loadbalance policy SIP_L7_POLICY
  inspect sip

interface vlan 10
  ip address 192.168.12.12 255.255.255.0
  service-policy input SIP_L4_POLICY
  no shutdown

SIP Load Balancing Based on SIP headers and SIP Inspection

The following configuration example shows the running configuration for SIP load-balancing on SIP headers and SIP inspection. The parts of the configuration that pertain strictly to the SIP load-balancing configuration are shown in bold text.

access-list ACL1 line 10 extended permit ip any any

rserver host RS1
  ip address 10.6.252.245
  inservice
rserver host RS2
  ip address 10.6.252.246
  inservice

serverfarm host SF3
  rserver RS1
    inservice
  rserver RS2
    inservice

sticky sip-header Call-ID SIP_GROUP
  serverfarm SF3

class-map type sip loadbalance match-any SIP_L7_CLASS
  match sip header Call_ID header-value sip:

class-map match-all SIP_L4_CLASS
  match virtual-address 192.168.12.15 tcp eq sip

policy-map type loadbalance sip first-match SIP_L7_POLICY
  class SIP_L7_CLASS
    sticky-serverfarm SIP_GROUP
policy-map multi-match SIP_L4_POLICY
  class SIP_L4_CLASS
    loadbalance vip inservice
    loadbalance policy SIP_L7_POLICY
      inspect sip

interface vlan 10
  ip address 192.168.12.12 255.255.255.0
  service-policy input SIP_L4_POLICY
  no shutdown

Example of a Server Load-Balancing Policy Configuration

The following example shows a running configuration that includes multiple class maps and policy maps that define a traffic policy for SLB. The class map and policy map configuration appears in bold in the example.

In this configuration, when a server farm is chosen for a connection, the connection is sent to a real server based on one of several load-balancing predictors. The leastconns predictor method load balances connections to the server that has the lowest number of open connections.

access-list ACL1 line 10 extended permit ip any any

probe tcp TCP
  interval 5
  faildetect 2
  passdetect interval 10
  open 3

parameter-map type http PERSIST-REBALANCE
  persistence-rebalance
parameter-map type connection PRED-CONNS-UDP_CONN
  set timeout inactivity 300
serverfarm host PRED-CONNS
  predictor leastconns
  rserver SERVER1
    inservice
  rserver SERVER2
    inservice
  rserver SERVER3
    inservice
Example of a Server Load-Balancing Policy Configuration

```
rserver SERVER4
  inservice
rserver SERVER5
  inservice
rserver SERVER6
  inservice
rserver SERVER7
  inservice
rserver SERVER8
  inservice
serverfarm host PRED-CONNS-UDP
  failaction purge
  predictor leastconns
  rserver SERVER1
    inservice
  rserver SERVER2
    inservice
  rserver SERVER3
    probe ICMP
    inservice
  rserver SERVER5
    inservice
  rserver SERVER6
    inservice
  rserver SERVER7
    inservice
serverfarm host PREDICTOR
  probe TCP
  rserver SERVER1
    inservice
  rserver SERVER2
    inservice
  rserver SERVER6
    inservice
  rserver SERVER7
    inservice
sticky http-cookie COOKIE_TEST STKY-GRP-43
  cookie offset 1 length 999
  timeout 30
  replicate sticky
serverfarm PREDICTOR

class-map match-all L4PRED-CONNS-UDP-VIP_128:2222_CLASS
  2 match virtual-address 192.168.120.128 udp eq 0
class-map match-all L4PRED-CONNS-VIP_128:80_CLASS
  2 match virtual-address 192.168.120.128 tcp eq www
```
Example of a Server Load-Balancing Policy Configuration

```plaintext
class-map match-all L4PREDICTOR_117:80_CLASS
  2 match virtual-address 192.168.120.117 tcp eq www
policy-map type loadbalance first-match L7PLBSF_PRED-CONNS_POLICY
  class class-default
    serverfarm PRED-CONNS
policy-map type loadbalance first-match L7PLBSF_PRED-CONNS-UDP_POLICY
  class class-default
    serverfarm PRED-CONNS-UDP
policy-map type loadbalance first-match L7PLBSF_PREDICTOR_POLICY
  class class-default
    sticky-serverfarm STKY-GRP-43
policy-map multi-match L4SH-Gold-VIPS_POLICY
  class L4PREDICTOR_117:80_CLASS
    loadbalance vip inservice
    loadbalance policy L7PLBSF_PREDICTOR_POLICY
    loadbalance vip icmp-reply active
    nat dynamic 1 vlan 120
    appl-parameter http advanced-options PERSIST-REBALANCE
  class L4PRED-CONNS-VIP_128:80_CLASS
    loadbalance vip inservice
    loadbalance policy L7PLBSF_PRED-CONNS_POLICY
    loadbalance vip icmp-reply active
    nat dynamic 1 vlan 120
    appl-parameter http advanced-options PERSIST-REBALANCE
  class L4PRED-CONNS-UDP-VIP_128:2222_CLASS
    loadbalance vip inservice
    loadbalance policy L7PLBSF_PRED-CONNS-UDP_POLICY
    loadbalance vip icmp-reply active
    nat dynamic 1 vlan 120
    appl-parameter http advanced-options PERSIST-REBALANCE
    connection advanced-options PRED-CONNS-UDP_CONN

interface vlan 120
  description Upstream VLAN_120 - Clients and VIPs
  ip address 192.168.120.1 255.255.255.0
  fragment chain 20
  fragment min-mtu 68
  access-group input ACL1
  nat-pool 1 192.168.120.70 192.168.120.70 netmask 255.255.255.0 pat
  service-policy input L4SH-Gold-VIPS_POLICY
  no shutdown
  ip route 10.1.0.0 255.255.255.0 192.168.120.254
```
Displaying Load-Balancing Configuration Information and Statistics

Use the commands in the following sections to display configuration information and statistics for Layer 3 and Layer 4, and Layer 7 class maps and policy maps. This section contains the following topics:

- Displaying Class-Map Configuration Information
- Displaying Policy-Map Configuration Information
- Displaying Parameter Map Configuration Information
- Displaying Load-Balancing Statistics
- Displaying HTTP Parameter Map Statistics
- Displaying Service-Policy Statistics
- Displaying the Layer 7 Match HTTP URL Statement Hit Counts
- Displaying HTTP Statistics

Displaying Class-Map Configuration Information

You can display class-map configuration information by using the `show running-config class-map` command in Exec mode. This command displays the names of all the class maps configured in the contexts to which you have access and the match statements configured in each class map. The syntax of this command is as follows:

```
show running-config class-map
```

Displaying Policy-Map Configuration Information

You can display policy-map configuration information by using the `show running-config policy-map` command in Exec mode. This command displays the names of all the policy maps configured in the contexts to which you have access and the class maps and actions configured in each policy map. The syntax of this command is as follows:

```
show running-config policy-map
```
Displaying Parameter Map Configuration Information

You can display a list of parameter maps and their configurations by using the `show running-config parameter-map` command in Exec mode. The syntax of this command is as follows:

```
show running-config parameter-map
```

Displaying Load-Balancing Statistics

You can display load-balancing statistics by using the `show stats loadbalance` command in Exec mode. The syntax of this command is as follows:

```
show stats loadbalance [radius | rdp]
```

The keywords and options are as follows:

- **radius**—(Optional) Displays Remote Authentication Dial-In User Service (RADIUS) load-balancing statistics associated with the current context.
- **rdp**—(Optional) Displays Reliable Datagram Protocol (RDP) load-balancing statistics associated with the current context.

For example, to see all load-balancing statistics, enter:

```
host1/Admin# show stats loadbalance
```

Table 3-8 describes the fields in the `show stats loadbalance` command output for an HTTP parameter map.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Version</td>
<td>VIP configuration changed during the load-balancing decision and the ACE rejected the connection.</td>
</tr>
<tr>
<td>Mismatch</td>
<td></td>
</tr>
<tr>
<td>Total Layer4</td>
<td>Total number of times that the ACE made a load-balancing decision at Layer 4.</td>
</tr>
<tr>
<td>Decisions</td>
<td></td>
</tr>
<tr>
<td>Total Layer4</td>
<td>Total number of Layer 4 connections that the ACE rejected.</td>
</tr>
<tr>
<td>Rejections</td>
<td></td>
</tr>
</tbody>
</table>
### Table 3-8  Field Descriptions for the `show stats loadbalance` Command Output (continued)

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Layer7 Decisions</td>
<td>Total number of times that the ACE made a load-balancing decision at Layer 7.</td>
</tr>
<tr>
<td>Total Layer7 Rejections</td>
<td>Total number of Layer 7 connections that the ACE rejected.</td>
</tr>
<tr>
<td>Total Layer4 LB Policy Misses</td>
<td>Total number of connection requests that did not match a configured Layer 4 load-balancing policy.</td>
</tr>
<tr>
<td>Total Layer7 LB Policy Misses</td>
<td>Total number of connection requests that did not match a configured Layer 7 load-balancing policy.</td>
</tr>
<tr>
<td>Total Times Rserver Was</td>
<td>Total number of times that the ACE attempted to load balance a request to a real server in a server farm, but the real server was not in service or was otherwise unavailable.</td>
</tr>
<tr>
<td>Unavailable</td>
<td></td>
</tr>
<tr>
<td>Total ACL denied</td>
<td>Total number of connection requests that were denied by an ACL.</td>
</tr>
<tr>
<td>Total IDMap Lookup Failures</td>
<td>Total number of times that the ACE failed to find the local ACE to peer ACE ID mapping for a real-server or a server-farm in the ID Map table for redundancy. A failure can occur if the peer ACE did not send a proper remote ID for the local ACE to look up and so the local ACE could not perform a mapping or if the ID Map table was not created.</td>
</tr>
</tbody>
</table>

Table 3-9 describes the fields in the `show stats loadbalance radius` command output.

### Table 3-9  Field Descriptions for the `show stats loadbalance radius` Command Output

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Requests Received</td>
<td>Total number of RADIUS client (NAS) requests received</td>
</tr>
<tr>
<td>Total Responses Received</td>
<td>Total number of RADIUS server responses received</td>
</tr>
</tbody>
</table>
### Table 3-9  Field Descriptions for the show stats loadbalance radius Command Output (continued)

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Retry Packets Received</td>
<td>Total number of retransmitted RADIUS requests received</td>
</tr>
<tr>
<td>Total Header Parse Results Received</td>
<td>Total number of RADIUS headers parsed and received by the load balance module</td>
</tr>
<tr>
<td>Total Body Parse Results Received</td>
<td>Total number of RADIUS attributes parsed and received by the load balance module</td>
</tr>
<tr>
<td>Total Data Parse Results Received</td>
<td>Total number of RADIUS packets (requests and responses) parsed and received by the load balance module</td>
</tr>
<tr>
<td>Total Packets Sent Out</td>
<td>Total number of RADIUS packets sent out by the ACE on both inbound and outbound interfaces</td>
</tr>
<tr>
<td>Total Sessions Allocated</td>
<td>Total number of RADIUS sessions allocated</td>
</tr>
<tr>
<td>Total Sessions Deleted</td>
<td>Total number of RADIUS sessions deleted</td>
</tr>
<tr>
<td>Total Username Sticky Added</td>
<td>Total number of sticky entries added based on the User-Name attribute</td>
</tr>
<tr>
<td>Total Calling-station Sticky Added</td>
<td>Total number of sticky entries added based on the Calling-Station-Id attribute</td>
</tr>
<tr>
<td>Total Framed-ip Sticky Added</td>
<td>Total number of sticky entries added based on the Framed-IP-Address attribute</td>
</tr>
<tr>
<td>Total End-user Packet Sticky Success</td>
<td>Total number of End-user packets that hit a FIP sticky entry previously created during RADIUS AAA processing</td>
</tr>
<tr>
<td>Total End-user Packet Sticky Failure</td>
<td>Total number of End-user packets that failed to hit a FIP sticky entry previously created during RADIUS AAA processing</td>
</tr>
</tbody>
</table>
### Table 3-9  
**Field Descriptions for the show stats loadbalance radius Command Output (continued)**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Acct-On/Off Requests Received</td>
<td>Total number of RADIUS Accounting On/Off requests received</td>
</tr>
<tr>
<td>Total Acct-On/Off Responses Received</td>
<td>Total number of RADIUS Accounting On/Off responses received</td>
</tr>
<tr>
<td>Total Acct-On/Off with No Rules</td>
<td>Total number of RADIUS Accounting On/Off requests for which no valid policy is found</td>
</tr>
<tr>
<td>Total Acct-On/Off Req Processing Done</td>
<td>Total number of RADIUS Accounting On/Off requests processed successfully by the ACE</td>
</tr>
<tr>
<td>Total NULL Packet Received Errors</td>
<td>Total number of invalid (with no data) RADIUS packets received</td>
</tr>
<tr>
<td>Total Parse Errors</td>
<td>Total number of RADIUS packets received with no valid RADIUS header</td>
</tr>
<tr>
<td>Total Proxy Mapper Errors</td>
<td>Number of proxy mapper entries creation failure. The proxy mapper entry structure maps each inbound connection with multiple outbound connections for point-to-multipoint protocols</td>
</tr>
<tr>
<td>Total Sticky Addition Failures</td>
<td>Total number of failures in creating a RADIUS-based sticky entry</td>
</tr>
<tr>
<td>Total memory allocation failures</td>
<td>Total number of failures in the allocation of any of the main RADIUS structures is use</td>
</tr>
<tr>
<td>Total stale packet errors</td>
<td>Total number of stale RADIUS packets upon receiving a new RADIUS request</td>
</tr>
</tbody>
</table>
Table 3-10 describes the fields in the `show stats loadbalance rdp` command output.

**Table 3-10 Field Descriptions for the show stats loadbalance rdp Command Output**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Parse Results Received</td>
<td>Total number of RDP parse results received by the load balance module</td>
</tr>
<tr>
<td>Total Packets Load Balanced</td>
<td>Total number of RDP packets load balanced</td>
</tr>
<tr>
<td>Total Packets with Routing Token</td>
<td>Total number of RDP packets received containing a Routing Token</td>
</tr>
<tr>
<td>Total Packets with Token Matching No Rservr</td>
<td>Total number for RDP packets for which the Routing Token does not match any of the configured real servers</td>
</tr>
</tbody>
</table>

### Displaying HTTP Parameter Map Statistics

You can display statistics for an HTTP parameter map by using the `show parameter-map` command in Exec mode. The syntax of this command is as follows:

```
show parameter-map [name]
```

The optional `name` argument is the identifier of an HTTP parameter map. Enter an unquoted text string with a maximum of 64 alphanumeric characters.

For example, to display statistics for the HTTP parameter map called HTTP_PARAMMAP, enter:

```
host1/Admin# show parameter-map HTTP_PARAMMAP
```
Table 3-11 describes the fields in the `show parameter-map` command output for an HTTP parameter map.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter-map</td>
<td>Unique identifier of the HTTP parameter map.</td>
</tr>
<tr>
<td>Type</td>
<td>HTTP.</td>
</tr>
<tr>
<td>Server-side connection reuse</td>
<td>Status of TCP server reuse feature: enabled or disabled.</td>
</tr>
<tr>
<td>Case-insensitive parsing</td>
<td>Status of the <code>case-insensitive</code> command: enabled or disabled.</td>
</tr>
<tr>
<td>Persistence-rebalance</td>
<td>Status of the <code>persistence-rebalance</code> command: strict, enabled or disabled.</td>
</tr>
<tr>
<td>Header-maxparse-length</td>
<td>Configured value or the default value of the <code>header-maxparse-length</code> command.</td>
</tr>
<tr>
<td>Content-maxparse-length</td>
<td>Configured value or the default value of the <code>content-maxparse-length</code> command.</td>
</tr>
<tr>
<td>Parse length-exceed action</td>
<td>Configured action for the <code>length-exceed</code> command: continue or drop.</td>
</tr>
<tr>
<td>Urlcookie-delimiters</td>
<td>Configured URL cookie delimiters.</td>
</tr>
</tbody>
</table>
Displaying Load-Balancing Configuration Information and Statistics

Table 3-11  Field Descriptions for the show parameter-map Command  
Output (continued)

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Urlcookie-start</td>
<td>Displays the start string of the secondary cookie or the none setting configured by the set secondary-cookie-start command in parameter map HTTP configuration mode. The default string of ?.</td>
</tr>
<tr>
<td>User-agent</td>
<td>Text string in the request to match as specified in the HTTP parameter map. A user agent is a client that initiates a request. Examples of user agents include browsers, editors, or other end user tools. The ACE does not compress the response to a request when the request contains a matching user agent string. The maximum size is 64 characters. The default is none.</td>
</tr>
</tbody>
</table>

Displaying Service-Policy Statistics

You can display statistics for service policies enabled globally within a context or on a specific interface by using the show service-policy command. If you do not enter an option with this command, the ACE displays all enabled policy statistics. This command also allows you to display statistics for a specific class map in a policy or a summary of statistics. The syntax of this command is as follows:

```
show service-policy [policy_name [class-map class_name]] [detail | summary | url-summary]
```

The keywords, arguments, and options are as follows:

- `policy_name`—(Optional) The name of an existing policy map that is currently in service (applied to an interface). Enter an unquoted text string with no spaces. If you do not enter the name of an existing policy map, the ACE displays information and statistics for all policy maps.
- `class-map class_name`—(Optional) Displays the statistics for the specified class map associated with the policy.
- `detail`—(Optional) Displays detailed statistics and status for the policy or class map.
• **summary**—(Optional) Displays a summary of policy or class map statistics and status information in a table format.

• **url-summary**—(Optional) Displays the number of times that a connection is established (hit count) based on match HTTP URL statements for a class map in a Layer 7 (L7) HTTP policy map. For information on this option and descriptions of the fields, see the “Displaying the Layer 7 Match HTTP URL Statement Hit Counts” section.

---

**Note**

The ACE updates the counters that the `show service-policy` command displays after the applicable connections are closed.

For example, to display detailed statistics and current status of the service policy MGMT_POLICYMAP, enter:

```
host1/Admin# show service-policy MGMT_POLICYMAP detail
```

**Table 3-12** describes the fields in the `show service-policy` command output.

**Table 3-12**  **Field Descriptions for the show service-policy Command Output**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Status</td>
<td>Current operational state of the service policy. Possible states are ACTIVE or INACTIVE.</td>
</tr>
<tr>
<td>Description</td>
<td>User-entered description of the policy map.</td>
</tr>
<tr>
<td>Interface</td>
<td>VLAN ID of the interface to which the policy map has been applied</td>
</tr>
<tr>
<td>Service Policy</td>
<td>Unique identifier of the policy map.</td>
</tr>
<tr>
<td>Class</td>
<td>Name of the class map associated with the service policy.</td>
</tr>
<tr>
<td>Loadbalance</td>
<td></td>
</tr>
<tr>
<td>L7 Policy</td>
<td>Name of the Layer 7 policy map associated with the service policy.</td>
</tr>
<tr>
<td>VIP Route Metric</td>
<td>Specifies the distance metric for the route as specified with the <code>loadbalance vip advertise</code> command. The ACE writes the value you specify in its routing table. Possible values are integers from 1 to 254.</td>
</tr>
</tbody>
</table>
### Displaying Load-Balancing Configuration Information and Statistics

**Table 3-12** Field Descriptions for the `show service-policy` Command Output (continued)

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VIP Route Advertise</td>
<td>Operational state of the <code>loadbalance vip advertise</code> command: ENABLED or DISABLED. This command is used with route health injection (RHI) to allow the ACE to advertise the availability of a VIP address throughout the network.</td>
</tr>
<tr>
<td>VIP ICMP Reply</td>
<td>Operational state of the <code>loadbalance vip icmp-reply</code> command. Possible states are: ENABLED, DISABLED, ENABLED-WHEN-ACTIVE, or ENABLED-WHEN-PRIMARY-SF-UP.</td>
</tr>
<tr>
<td>VIP State</td>
<td>Operational state of the virtual server: INSERVICE or OUTOFSERVICE.</td>
</tr>
<tr>
<td>Curr Conns</td>
<td>Number of active connections to the VIP.</td>
</tr>
<tr>
<td>Note</td>
<td>The Curr Conns field does not increase in the <code>show service-policy</code> command output when you configure the UDP booster feature (see “Configuring UDP Booster” section). The ACE performs a destination hash when it receives a UDP packet. If the destination hash does not exist in the hash table then a new load-balancing decision is made which increments the <code>show service-policy</code> connection counter and a new connection is added to the connection table. However, if there is a destination hash match, then the ACE creates a new connection without involving a load-balancing decision. In this case, the connection table includes the connection, but the <code>show service-policy</code> Curr Conns output does not update.</td>
</tr>
<tr>
<td>Hit Count</td>
<td>Number of times a connection was established with this VIP.</td>
</tr>
<tr>
<td>Dropped Conns</td>
<td>Number of connections that the ACE discarded.</td>
</tr>
<tr>
<td>Client Pkt Count</td>
<td>Number of packets received from the client.</td>
</tr>
</tbody>
</table>
### Table 3-12  Field Descriptions for the show service-policy Command Output (continued)

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Client Byte Count</td>
<td>Number of bytes received from the client.</td>
</tr>
<tr>
<td>Server Pkt Count</td>
<td>Number of packets received from the server.</td>
</tr>
<tr>
<td>Server Byte Count</td>
<td>Number of bytes received from the server.</td>
</tr>
<tr>
<td>Max-conn-limit</td>
<td>Configured value of the <code>conn-limit max</code> command. See Chapter 2, Configuring Real Servers and Server Farms.</td>
</tr>
<tr>
<td>Drop-count</td>
<td>Number of connections that were dropped because the maximum connection limit was exceeded.</td>
</tr>
<tr>
<td>Conn-rate-limit</td>
<td>Configured value of the <code>rate-limit connection</code> command. See Chapter 2, Configuring Real Servers and Server Farms.</td>
</tr>
<tr>
<td>Drop-count</td>
<td>Number of connections that were dropped because the connection rate limit was exceeded.</td>
</tr>
<tr>
<td>bandwidth-rate-limit</td>
<td>Configured value of the <code>rate-limit bandwidth</code> command. See Chapter 2, Configuring Real Servers and Server Farms.</td>
</tr>
<tr>
<td>Drop-count</td>
<td>Number of connections that were dropped because the bandwidth rate limit was exceeded.</td>
</tr>
<tr>
<td>L4 Policy Stats</td>
<td></td>
</tr>
<tr>
<td>Total Req/Resp</td>
<td>Total number of requests and responses for the policy map.</td>
</tr>
<tr>
<td>Total Allowed</td>
<td>Total number of packets received and allowed.</td>
</tr>
<tr>
<td>Total Dropped</td>
<td>Total number of packets received and discarded.</td>
</tr>
<tr>
<td>Total Logged</td>
<td>Total number of errors logged.</td>
</tr>
<tr>
<td>L7 loadbalance policy</td>
<td>Identifier of the Layer 7 policy map.</td>
</tr>
<tr>
<td>Class-map</td>
<td>Identifier of the associated class map.</td>
</tr>
</tbody>
</table>
You can display the number of times that a connection is established (hit count) based on match HTTP URL statements for a class map in a Layer 7 (L7) HTTP policy map by using the `show service-policy url-summary` command. If you do not enter a policy map name with this command, the ACE displays the match URL statement hit counts for all class maps in L7 HTTP policy maps.

The syntax of this command is as follows:

```
show service-policy [policy_name [class-map class_name]] url-summary
```
The options are as follows:

- **policy_name**—(Optional) Name of an existing Layer 3 and Layer 4 HTTP policy map. Enter an unquoted text string with no spaces. If you do not enter a policy map name with this command, the ACE displays the match URL statement hit counts for all class maps in L7 HTTP policy maps.

- **class-map class_name**—(Optional) Displays the statement hit counts for the specified class map associated with the policy. Enter the name as an unquoted text string with no spaces.

For example, to display the hit count for the match HTTP URL statements for all class maps in all policy maps, enter the following command:

```
host1/Admin# show service-policy url-summary
```

Table 3-13 describes the fields in the `show service-policy url-summary` command output.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service Policy</td>
<td>Unique identifier of the policy map.</td>
</tr>
<tr>
<td>L3-Class</td>
<td>Name of the Layer 3 class map associated with the service policy.</td>
</tr>
<tr>
<td>L7-Class</td>
<td>Identifier of the Layer 7 class map.</td>
</tr>
<tr>
<td>match http url</td>
<td>The HTTP URL match statement.</td>
</tr>
<tr>
<td>hit</td>
<td>The number of times that a connection is established based on a specific URL match statement.</td>
</tr>
</tbody>
</table>

**Note**

The URL hit counter is per match statement per load-balancing Layer 7 policy. If you are using the same combination of Layer 7 policy and class maps with URL match statements in different VIPs, the count is combined. If the ACE configuration exceeds 64K URL and load-balancing policy combinations, this counter displays NA.
Displaying HTTP Statistics

You can display HTTP statistics, including header insertion and server reuse statistics by using the `show stats http` command in Exec mode. The syntax of this command is as follows:

```
show stats http
```

For example, enter:

```
host1/Admin# show stats http
```

```
+------------------------------------------+
|-------------- HTTP statistics -----------|
+------------------------------------------+
| LB parse result msgs sent : 0       , TCP data msgs sent : 0 |
| Inspect parse result msgs : 0       , SSL data msgs sent : 0 |
| TCP fin msgs sent : 0               , TCP rst msgs sent : 0 |
| Bounce fin msgs sent : 0            , Bounce rst msgs sent : 0 |
| SSL fin msgs sent : 0               , SSL rst msgs sent : 0 |
| Drain msgs sent : 0                 , Particles read : 0    |
| Reuse msgs sent : 0                 , HTTP requests : 0     |
| Reproxed requests : 0               , Headers removed : 0   |
| Headers inserted : 0                , HTTP redirects : 0    |
| HTTP chunks : 0                     , Pipelined requests : 0|
| HTTP unproxy conns : 0              , Pipeline flushes : 0  |
| Whitespace appends : 0              , Second pass parsing : 0|
| Response entries recycled : 0       , Analysis errors : 0   |
| Header insert errors : 0            , Max parselen errors : 0|
| Static parse errors : 0             , Resource errors : 0   |
| Invalid path errors : 0             , Bad HTTP version errors : 0|
| Headers rewritten : 0               , Header rewrite errors : 0|
| Unproxy msgs sent : 0               |
```
Clearing SLB Statistics

This section describes the commands that you can use to clear load-balancing statistics. It includes the following topics:

- Clearing Load-Balancing Statistics
- Clearing Service-Policy Statistics
- Clearing HTTP Statistics

Clearing Load-Balancing Statistics

You can clear all load-balancing statistics in the current context by using the `clear stats loadbalance` command in Exec mode. The syntax of this command is as follows:

```
clear stats loadbalance [radius | rdp | rtsp | sip]
```

The keywords and options are as follows:

- **radius**—(Optional) Clears Remote Authentication Dial-In User Service (RADIUS) load-balancing statistical information.
- **rdp**—(Optional) Clears Reliable Datagram Protocol (RDP) load-balancing statistical information.
- **sip**—(Optional) Clears Session Initiation Protocol (SIP) load-balancing statistical information.

For example, enter:

```
host1/Admin# clear stats loadbalance
```

**Note**

If you have redundancy configured, you need to explicitly clear load-balancing statistics on both the active and the standby ACEs. Clearing statistics on the active module only leaves the standby module’s statistics at the old values.
Clearing Service-Policy Statistics

You can clear service policy statistics by using the `clear service-policy` command. The syntax of this command is as follows:

```
clear service-policy policy_name
```

For the `policy_name` argument, enter the identifier of an existing policy map that is currently in service (applied to an interface).

For example, to clear the statistics for the policy map L4SLBPOLICY that is currently in service, enter:

```
host1/Admin# clear service-policy L4SLBPOLICY
```

**Note**
If you have redundancy configured, you need to explicitly clear service-policy statistics on both the active and the standby ACEs. Clearing statistics on the active module only leaves the standby module’s statistics at the old values.

Clearing HTTP Statistics

You can clear all HTTP statistics in the current context by using the `clear stats http` command in Exec mode. The syntax of this command is as follows:

```
clear stats http
```

For example, enter:

```
host1/Admin# clear stats http
```

**Note**
If you have redundancy configured, you need to explicitly clear HTTP statistics on both the active and the standby ACEs. Clearing statistics on the active module only leaves the standby module’s statistics at the old values.
Where to Go Next

To configure health probes for your real servers, go to Chapter 4, Configuring Health Monitoring. To configure stickiness (connection persistence), see Chapter 5, Configuring Stickiness. To configure firewall load balancing (FWLB), see Chapter 6, Configuring Firewall Load Balancing.