



## CHAPTER 6

# Configuring Stickiness

---

**Date:** 7/8/09

This section provides information about sticky behavior and procedures for configuring stickiness with the ANM. Sections include:

- [Stickiness Overview, page 6-1](#)
- [Sticky Types, page 6-2](#)
- [Sticky Groups, page 6-6](#)
- [Sticky Table, page 6-6](#)
- [Configuring Sticky Groups, page 6-7](#)

## Stickiness Overview

When customers visit an e-commerce site, they usually start out by browsing the site, the Internet equivalent of window shopping. Depending on the application, the site may require that the client become “stuck” to one server once the connection is established, or the application may not require this until the client starts to build a shopping cart.

In either case, once the client adds items to the shopping cart, it is important that all of the client requests get directed to the same server so that all the items are contained in one shopping cart on one server. An instance of a customer's shopping cart is typically local to a particular Web server and is not duplicated across multiple servers.

E-commerce applications are not the only types of applications that require stickiness. Any Web application that maintains client information may require stickiness, such as banking applications or online trading. Other uses include FTP and HTTP file transfers.

Stickiness allows the same client to maintain multiple simultaneous or subsequent TCP or IP connections with the same real server for the duration of a session. A session, as used here, is defined as a series of transactions between a client and a server over some finite period of time (from several minutes to several hours). This feature is particularly useful for e-commerce applications where a client needs to maintain multiple connections with the same server while shopping online, especially while building a shopping cart and during the checkout process.

Depending on the configured SLB policy, the ACE “sticks” a client to an appropriate server after the ACE has determined which load-balancing method to use. If the ACE determines that a client is already stuck to a particular server, then the ACE sends that client request to that server, regardless of the load-balancing criteria specified by the matched policy. If the ACE determines that the client is not stuck to a particular server, it applies the normal load-balancing rules to the content request.

For information on stickiness, see:

- [Sticky Types, page 6-2](#)
- [Sticky Groups, page 6-6](#)
- [Sticky Table, page 6-6](#)

**Related Topics**

- [Configuring Virtual Server Default Layer 7 Load Balancing, page 4-41](#)
- [Configuring Sticky Groups, page 6-7](#)

## Sticky Types

All ACE devices support stickiness based on:

- HTTP cookies
- HTTP headers
- IP addresses

ACE 2.0 modules and the ACE 4710 A3(1.0) release also support stickiness based on:

- HTTP content
- Layer 4 payloads
- RADIUS attributes
- RTSP headers
- SIP headers

**Related Topics**

- [HTTP Content Stickiness, page 6-3](#)
- [HTTP Cookie Stickiness, page 6-3](#)
- [HTTP Header Stickiness, page 6-4](#)
- [IP Netmask Stickiness, page 6-4](#)
- [Layer 4 Payload Stickiness, page 6-4](#)
- [RADIUS Stickiness, page 6-5](#)
- [RTSP Header Stickiness, page 6-5](#)
- [SIP Header Stickiness, page 6-5](#)

## HTTP Content Stickiness

HTTP content stickiness allows you to stick a client to a server based on the content of an HTTP packet. You can specify a beginning pattern and ending pattern, the number of bytes to parse, and an offset that specifies how many bytes to ignore from the beginning of the data.

### Related Topics

- [Configuring Stickiness, page 6-1](#)
- [Sticky Types, page 6-2](#)
- [Sticky Groups, page 6-6](#)
- [Sticky Table, page 6-6](#)

## HTTP Cookie Stickiness

Client *cookies* uniquely identify clients to the ACE and the servers providing content. A cookie is a small data structure within the HTTP header that is used by a server to deliver data to a Web client and request that the client store the information. In certain applications, the client returns the information to the server to maintain the connection state or persistence between the client and the server.

When the ACE examines a request for content and determines through policy matching that the content is sticky, it examines any cookie or URL present in the content request. The ACE uses the information in the cookie or URL to direct the content request to the appropriate server.

The ACE supports the following types of cookie stickiness:

- Dynamic cookie learning

You can configure the ACE to look for a specific cookie name and automatically learn its value either from the client request HTTP header or from the server Set-Cookie message in the server response. Dynamic cookie learning is useful when dealing with applications that store more than just the session ID or user ID within the same cookie. Only very specific bytes of the cookie value are relevant to stickiness.

By default, the ACE learns the entire cookie value. You can optionally specify an offset and length to instruct the ACE to learn only a portion of the cookie value.

Alternatively, you can specify a secondary cookie value that appears in the URL string in the HTTP request. This option instructs the ACE to search for (and eventually learn or stick to) the cookie information as part of the URL. URL learning is useful with applications that insert cookie information as part of the HTTP URL. In some cases, you can use this feature to work around clients that reject cookies.

- Cookie insert

The ACE inserts the cookie on behalf of the server upon the return request, so that the ACE can perform cookie stickiness even when the servers are not configured to set cookies. The cookie contains information that the ACE uses to ensure persistence to a specific real server.

### Related Topics

- [Configuring Stickiness, page 6-1](#)
- [Sticky Types, page 6-2](#)
- [Sticky Groups, page 6-6](#)
- [Sticky Table, page 6-6](#)

## HTTP Header Stickiness

You can use HTTP-header information to provide stickiness. With HTTP header stickiness, you can specify a header offset to provide stickiness based on a unique portion of the HTTP header.

### Related Topics

- [Configuring Stickiness, page 6-1](#)
- [Sticky Types, page 6-2](#)
- [Sticky Groups, page 6-6](#)
- [Sticky Table, page 6-6](#)

## IP Netmask Stickiness

You can use the source IP address, the destination IP address, or both to uniquely identify individual clients and their requests for stickiness purposes based on their IP netmask. However, if an enterprise or a service provider uses a megaproxy to establish client connections to the Internet, the source IP address no longer is a reliable indicator of the true source of the request. In this case, you can use cookies or one of the other sticky methods to ensure session persistence.

### Related Topics

- [Configuring Stickiness, page 6-1](#)
- [Sticky Types, page 6-2](#)
- [Sticky Groups, page 6-6](#)
- [Sticky Table, page 6-6](#)

## Layer 4 Payload Stickiness

Layer 4 payload stickiness allows you to stick a client to a server based on the data in Layer 4 frames. You can specify a beginning pattern and ending pattern, the number of bytes to parse, and an offset that specifies how many bytes to ignore from the beginning of the data.

### Related Topics

- [Configuring Stickiness, page 6-1](#)
- [Sticky Types, page 6-2](#)
- [Sticky Groups, page 6-6](#)
- [Sticky Table, page 6-6](#)

## RADIUS Stickiness

RADIUS stickiness can be based on the following RADIUS attributes:

- Calling Station ID
- Username

### Related Topics

- [Configuring Stickiness, page 6-1](#)
- [Sticky Types, page 6-2](#)
- [Sticky Groups, page 6-6](#)
- [Sticky Table, page 6-6](#)

## RTSP Header Stickiness

RTSP stickiness is based on information in the RTSP session header. With RTSP header stickiness, you can specify a header offset to provide stickiness based on a unique portion of the RTSP header.

### Related Topics

- [Configuring Stickiness, page 6-1](#)
- [Sticky Types, page 6-2](#)
- [Sticky Groups, page 6-6](#)
- [Sticky Table, page 6-6](#)

## SIP Header Stickiness

SIP header stickiness is based on the SIP Call-ID header field. SIP header stickiness requires the entire SIP header, so you cannot specify an offset.

### Related Topics

- [Configuring Stickiness, page 6-1](#)
- [Sticky Types, page 6-2](#)
- [Sticky Groups, page 6-6](#)
- [Sticky Table, page 6-6](#)

# Sticky Groups

The ACE uses the concept of sticky groups to configure stickiness. A sticky group allows you to specify sticky attributes. After you configure a sticky group and its attributes, you associate the sticky group with a Layer 7 policy-map action in a Layer 7 SLB policy map. You can create a maximum of 4096 sticky groups in each context. Each sticky group that you configure on the ACE contains a series of parameters that determine:

- Sticky method
- Timeout
- Replication
- Sticky method-specific attributes

**Note**

The context in which you configure a sticky group must be associated with a resource class that allocates a portion of ACE resources to stickiness. See [Using Resource Classes, page 3-34](#) for information about configuring ACE resources.

**Related Topics**

- [Configuring Stickiness, page 6-1](#)
- [Sticky Types, page 6-2](#)
- [Sticky Table, page 6-6](#)

# Sticky Table

The ACE uses a sticky table to keep track of sticky connections. Table entries include:

- Sticky groups
- Sticky methods
- Sticky connections
- Real servers

The sticky table can hold a maximum of four million entries (four million simultaneous users). When the table reaches the maximum number of entries, additional sticky connections cause the table to wrap and the first users become unstuck from their respective servers.

The ACE uses a configurable timeout mechanism to age out sticky table entries. When an entry times out, it becomes eligible for reuse. High connection rates may cause the premature aging out of sticky entries. In this case, the ACE reuses the entries that are closest to expiration first.

Sticky entries can be either dynamic (generated by the ACE on-the-fly) or static (user-configured). When you create a static sticky entry, the ACE places the entry in the sticky table immediately. Static entries remain in the sticky database until you remove them from the configuration. You can create a maximum of 4096 static sticky entries in each context.

If the ACE takes a real server out of service for whatever reason (probe failure, no inservice command, or ARP timeout), the ACE removes from the database any sticky entries that are related to that server.

**Related Topics**

- [Configuring Stickiness, page 6-1](#)
- [Sticky Types, page 6-2](#)
- [Sticky Groups, page 6-6](#)

## Configuring Sticky Groups

Stickiness (or session persistence) is a feature that allows the same client to maintain multiple simultaneous or subsequent TCP connections with the same real server for the duration of a session. A session, as used here, is defined as a series of transactions between a client and a server over some finite period of time (from several minutes to several hours). This feature is particularly useful for e-commerce applications where a client needs to maintain multiple TCP connections with the same server while shopping online, especially while building a shopping cart and during the checkout process.

E-commerce applications are not the only types of applications that require stickiness. Any Web application that maintains client information may require stickiness, such as banking applications or online trading. Other uses include FTP and HTTP file transfers.

The ACE uses the concept of sticky groups to configure stickiness. A sticky group allows you to specify sticky attributes. After you configure a sticky group and its attributes, you associate the sticky group with a Layer 7 policy-map action in a Layer 7 SLB policy map.

Use this procedure to configure sticky groups.

**Note**

The context in which you configure a sticky group must be associated with a resource class that allocates a portion of ACE resources to stickiness. See [Using Resource Classes, page 3-34](#) for information about configuring ACE resources.

**Assumption**

The context in which you are configuring a sticky group is associated with a resource class that allocates resources to stickiness.

**Procedure**

- Step 1** Select **Config > Devices > context > Load Balancing > Stickiness**. The Sticky Groups table appears.
- Step 2** Click **Add** to add a new sticky group, or select an existing sticky group you want to modify, then click **Edit**.
- Step 3** Configure the sticky group using the information in [Table 6-1](#).

**Table 6-1** Sticky Group Attributes

Field	Description
Group Name	The sticky group identifier. Valid entries are unquoted text strings with no spaces and a maximum of 64 alphanumeric characters.
Type	<p>Select the method to be used when establishing sticky connections and configure any type-specific attributes:</p> <ul style="list-style-type: none"> <li>• HTTP Content—The ACE sticks client connections to the same real server based on a string in the data portion of the HTTP packet. See <a href="#">Table 6-2</a> for additional configuration options.</li> <li>• HTTP Cookie—The ACE either learns a cookie from the HTTP header of a client request or inserts a cookie in the Set-Cookie header of the response from the server to the client, and then uses the learned cookie to provide stickiness between the client and server for the duration of the transaction. See <a href="#">Table 6-3</a> for additional configuration options.</li> <li>• HTTP Header—The ACE sticks client connections to the same real server based on HTTP headers. See <a href="#">Table 6-4</a> for additional configuration options.</li> <li>• IP Netmask—The ACE sticks a client to the same server for multiple subsequent connections as needed to complete a transaction using the client source IP address, the destination IP address, or both. See <a href="#">Table 6-5</a> for additional configuration options.</li> </ul> <p><b>Note</b> If an organization uses a megaproxy to load balance client requests across multiple proxy servers when a client connects to the Internet, the source IP address is no longer a reliable indicator of the true source of the request. In this situation, you can use cookies or another sticky method to ensure session persistence.</p> <ul style="list-style-type: none"> <li>• Layer 4 Payload—The ACE sticks client connections to the same real server based on a string in the payload portion of the Layer 4 protocol packet. See <a href="#">Table 6-6</a> for additional configuration options.</li> <li>• RADIUS—The ACE sticks client connections to the same real server based on a RADIUS attribute. See <a href="#">Table 6-7</a> for additional configuration options.</li> <li>• RTSP Header—The ACE sticks client connections to the same real server based on the RTSP Session header field. See <a href="#">Table 6-8</a> for additional configuration options.</li> <li>• SIP Header—The ACE sticks client connections to the same real server based on the SIP Call-ID header field.</li> </ul>
Sticky Server Farm	Select a server farm you want to associate with this sticky group.
Backup Server Farm	Select a backup server farm to be associated with this sticky group. If the primary server farm is down, the ACE uses the backup server farm.
Aggregate State	<p>This field appears when a server farm and backup server farm are selected.</p> <p>Check this check box to indicate that the state of the backup server farm is tied to the virtual server state. Clear this check box if the backup server farm is not tied to the virtual server state.</p>
Sticky Enabled On Backup Server Farm	<p>This field appears when a server farm and backup server farm are selected.</p> <p>Check this check box to indicate that the backup server farm is sticky. Clear this check box if the backup server farm is not sticky.</p>

**Table 6-1** Sticky Group Attributes (continued)

Field	Description
Replicate On HA Peer	<p>Check this check box to indicate that the ACE to replicate sticky table entries on the standby ACE. If a failover occurs and this option is selected, the new active ACE can maintain the existing sticky connections.</p> <p>Clear this check box to indicate that the ACE is not to replicate sticky table entries on the standby ACE.</p>
Timeout (Minutes)	Enter the number of minutes that the ACE keeps the sticky information for a client connection in the sticky table after the latest client connection terminates. Valid entries are integers from 1 to 65535; the default is 1440 minutes (24 hours).
Timeout Active Connections	<p>Check this check box to specify that the ACE is to time out sticky table entries even if active connections exist after the sticky timer expires.</p> <p>Clear this check box to specify that the ACE is not to time out sticky table entries even if active connections exist after the sticky timer expires. This is the default behavior.</p>

**Step 4** Click:

- **Deploy Now** to deploy this configuration on the ACE. To configure sticky statics, see [Configuring Sticky Statics, page 6-14](#).
- **Cancel** to exit the procedure without saving your entries and to return to the Sticky Groups table.
- **Next** to deploy your entries and to configure another sticky group.

**Related Topics**

- [Configuring Sticky Statics, page 6-14](#)
- [Configuring Virtual Context Class Maps, page 11-6](#)
- [Configuring Virtual Context Policy Maps, page 11-30](#)
- [Configuring Real Servers, page 5-4](#)
- [Configuring Server Farms, page 5-12](#)

## Sticky Group Attribute Tables

Refer to the following topics for sticky group type-specific attributes:

- [HTTP Content Sticky Group Attributes, page 6-10](#)
- [HTTP Cookie Sticky Group Attributes, page 6-11](#)
- [HTTP Header Sticky Group Attributes, page 6-11](#)
- [IP Netmask Sticky Group Attributes, page 6-12](#)
- [Layer 4 Payload Sticky Group Attributes, page 6-12](#)
- [RADIUS Sticky Group Attributes, page 6-13](#)
- [RTSP Header Sticky Group Attributes, page 6-13](#)

## HTTP Content Sticky Group Attributes

Table 6-2 HTTP Content Sticky Group Attributes

Field	Description
HTTP Content	<p>HTTP content may change over time with only a portion remaining constant throughout a transaction between the client and a server.</p> <p>Check the check box to configure the ACE to use the constant portion of HTTP content to make persistent connections to a specific server. Clear the check box to identify specific content for stickiness in the Offset, Length, Begin Pattern, and End Pattern fields.</p>
Offset	<p>Enter the number of bytes the virtual server is to ignore starting with the first byte of the cookie. Valid entries are integers from 0 to 999. The default is 0 (zero), which indicates that the virtual server does not exclude any portion of the cookie.</p>
Length (Bytes)	<p>Enter the length of the portion of the cookie (starting with the byte after the offset value) that the ACE is to use for sticking the client to the server. Valid entries are integers from 1 to 1000.</p>
Begin Pattern	<p>Enter the beginning pattern of the HTTP content payload and the pattern string to match before hashing. If you do not specify a beginning pattern, the ACE begins parsing immediately after the offset byte. You cannot configure different beginning and ending patterns for different server farms that are part of the same traffic classification.</p> <p>Valid entries are unquoted text strings with no spaces and a maximum of 255 alphanumeric characters. You can enter a text string with spaces provided that you enclose the entire string in quotation marks ("). The ACE supports regular expressions for matching string expressions. <a href="#">Table 11-35</a> lists the supported characters that you can use for matching string expressions.</p>
End Pattern	<p>Enter the pattern that marks the end of hashing. If you do not specify an end pattern or a length, the ACE continues to parse the data until it reaches the end of the field or packet, or until it reaches the maximum body parse length. You cannot configure different beginning and ending patterns for different server farms that are part of the same traffic classification.</p> <p>Valid entries are unquoted text strings with no spaces and a maximum of 255 alphanumeric characters. You can enter a text string with spaces provided that you enclose the entire string in quotation marks ("). The ACE supports regular expressions for matching string expressions. <a href="#">Table 11-35</a> lists the supported characters that you can use for matching string expressions.</p>

## HTTP Cookie Sticky Group Attributes

**Table 6-3** HTTP Cookie Sticky Group Attributes

Field	Description
Cookie Name	Enter a unique identifier for the cookie. Valid entries are unquoted text strings with no spaces and a maximum of 64 alphanumeric characters.
Enable Insert	Check the check box if the virtual server is to insert a cookie in the Set-Cookie header of the response from the server to the client. This option is useful when you want to use a session cookie for persistence but the server is not currently setting the appropriate cookie. When selected, the virtual server selects a cookie value that identifies the original server from which the client received a response. For subsequent connections of the same transaction, the client uses the cookie to stick to the same server.  Clear the check box to disable cookie insertion.
Offset	Enter the number of bytes the virtual server is to ignore starting with the first byte of the cookie. Valid entries are integers from 0 to 999. The default is 0 (zero), which indicates that the virtual server does not exclude any portion of the cookie.
Length (Bytes)	Enter the length of the portion of the cookie (starting with the byte after the offset value) that the ACE is to use for sticking the client to the server. Valid entries are integers from 1 to 1000.
Secondary Name	Enter an alternate cookie name that is to appear in the URL string of the Web page on the server. The virtual server uses this cookie to maintain a sticky connection between a client and a server and adds a secondary entry in the sticky table. Valid entries are unquoted text strings with no spaces and a maximum of 64 characters.

## HTTP Header Sticky Group Attributes

**Table 6-4** HTTP Header Sticky Group Attributes

Field	Description
Header Name	Select the HTTP header to use for sticking client connections.
Offset	Enter the number of bytes the virtual server is to ignore starting with the first byte of the cookie. Valid entries are integers from 0 to 999. The default is 0 (zero), which indicates that the virtual server does not exclude any portion of the cookie.
Length (Bytes)	Enter the length of the portion of the cookie (starting with the byte after the offset value) that the ACE is to use for sticking the client to the server. Valid entries are integers from 1 to 1000.

## IP Netmask Sticky Group Attributes

**Table 6-5** IP Netmask Sticky Group Attributes

Field	Description
Netmask	Select the netmask to apply to the source IP address, destination IP address, or both.
Address Type	Indicate whether this sticky type is to be applied to the client source IP address, the destination IP address, or both: <ul style="list-style-type: none"> <li>Both—The sticky type is to be applied to both the source IP address and the destination IP address.</li> <li>Destination—The sticky type is to be applied to the destination IP address only.</li> <li>Source—The sticky type is to be applied to the source IP address only.</li> </ul>

## Layer 4 Payload Sticky Group Attributes

**Table 6-6** Layer 4 Payload Sticky Group Attributes

Field	Description
Offset	Enter the number of bytes the virtual server is to ignore starting with the first byte of the cookie. Valid entries are integers from 0 to 999. The default is 0 (zero), which indicates that the virtual server does not exclude any portion of the cookie.
Length (Bytes)	Enter the length of the portion of the cookie (starting with the byte after the offset value) that the ACE is to use for sticking the client to the server. Valid entries are integers from 1 to 1000, and the default is 1000.
Begin Pattern	Enter the beginning pattern of the Layer 4 payload and the pattern string to match before hashing. If you do not specify a beginning pattern, the ACE begins parsing immediately after the offset byte. You cannot configure different beginning and ending patterns for different server farms that are part of the same traffic classification.  Valid entries are unquoted text strings with no spaces and a maximum of 255 alphanumeric characters. You can enter a text string with spaces provided that you enclose the entire string in quotation marks ("). The ACE supports regular expressions for matching string expressions. <a href="#">Table 11-35</a> lists the supported characters that you can use for matching string expressions.

**Table 6-6** *Layer 4 Payload Sticky Group Attributes (continued)*

Field	Description
End Pattern	<p>Enter the pattern that marks the end of hashing. If you do not specify an end pattern or a length, the ACE continues to parse the data until it reaches the end of the field or packet, or until it reaches the maximum body parse length. You cannot configure different beginning and ending patterns for different server farms that are part of the same traffic classification.</p> <p>Valid entries are unquoted text strings with no spaces and a maximum of 255 alphanumeric characters. You can enter a text string with spaces provided that you enclose the entire string in quotation marks (""). The ACE supports regular expressions for matching string expressions. <a href="#">Table 11-35</a> lists the supported characters that you can use for matching string expressions.</p>
Enable Sticky For Response	<p>Check the check box to enable the ACE to parse server responses and perform sticky learning. The ACE uses a hash of the server response bytes to populate the sticky database. The next time that the ACE receives a client request with those same bytes, it sticks the client to the same server.</p> <p>Clear the check box to reset the behavior of the ACE to the default of not parsing server responses and performing sticky learning.</p>

### RADIUS Sticky Group Attributes

**Table 6-7** *RADIUS Sticky Group Attributes*

Field	Description
RADIUS Types	<p>Select the RADIUS attribute to use for sticking client connections:</p> <ul style="list-style-type: none"> <li>• N/A—This option is not configured.</li> <li>• RADIUS Calling ID—Stickiness is based on the RADIUS framed IP attribute and the calling station ID attribute.</li> <li>• RADIUS User Name—Stickiness is based on the RADIUS framed IP attribute and the username attribute.</li> </ul>

### RTSP Header Sticky Group Attributes

**Table 6-8** *RTSP Header Sticky Group Attributes*

Field	Description
Offset	Enter the number of bytes the virtual server is to ignore starting with the first byte of the cookie. Valid entries are integers from 0 to 999. The default is 0 (zero), which indicates that the virtual server does not exclude any portion of the cookie.
Length (Bytes)	Enter the length of the portion of the cookie (starting with the byte after the offset value) that the ACE is to use for sticking the client to the server. Valid entries are integers from 1 to 1000, and the default is 1000.

## Viewing All Sticky Groups by Context

Use this procedure to view all sticky groups associated with a virtual context.

### Procedure

- 
- Step 1** Select **Config > Devices**. The Virtual Contexts table appears.
- Step 2** Select the virtual context with the sticky groups you want to view, then select **Load Balancing > Stickiness**. The Sticky Groups table appears, listing the sticky groups associated with the selected context.
- 

### Related Topics

- [Configuring Sticky Groups, page 6-7](#)
- [Configuring Sticky Statics, page 6-14](#)

## Configuring Sticky Statics

Use this procedure to configure sticky statics.

### Assumption

A sticky group has been configured. See [Configuring Sticky Groups, page 6-7](#) for more information.

### Procedure

- 
- Step 1** Select **Config > Devices > context > Load Balancing > Stickiness**. The Sticky Groups table appears.
- Step 2** Select the sticky group you want to configure for sticky statics. The Sticky Statics table appears.
- Step 3** Click **Add** to add a new entry to the table, or select an existing entry, then click **Edit** to modify it. The Sticky Statics configuration screen appears.
- Step 4** In the Sequence Number field, either accept the automatically incremented number for this entry or enter a new sequence number. The sequence number indicates the order in which multiple sticky static configurations are applied.
- Step 5** In the Type field, confirm that the correct sticky group type is selected.
- HTTP Content—The ACE sticks client connections to the same real server based on a string in the data portion of the HTTP packet.
  - HTTP Cookie—The ACE either learns a cookie from the HTTP header of a client request or inserts a cookie in the Set-Cookie header of the response from the server to the client, and then uses the learned cookie to provide stickiness between the client and server for the duration of the transaction.
  - HTTP Header—The ACE sticks client connections to the same real server based on HTTP headers.
  - IP Netmask—The ACE sticks a client to the same server for multiple subsequent connections as needed to complete a transaction using the client source IP address, the destination IP address, or both.



---

**Note** If an organization uses a megaproxy to load balance client requests across multiple proxy servers when a client connects to the Internet, the source IP address is no longer a reliable indicator of the true source of the request. In this situation, you can use cookies or another sticky method to ensure session persistence.

---

- **Layer 4 Payload**—The ACE sticks client connections to the same real server based on a string in the payload portion of the Layer 4 protocol packet.
- **RADIUS**—The ACE sticks client connections to the same real server based on a RADIUS attribute.
- **RTSP Header**—The ACE sticks client connections to the same real server based on the RTSP Session header field.
- **SIP Header**—The ACE sticks client connections to the same real server based on the SIP Call-ID header field.

**Step 6** If you select HTTP Cookie, HTTP, RTSP, or SIP Header for sticky type, in the Static Value field, enter the cookie string value. Valid entries are unquoted text strings with a maximum of 255 alphanumeric characters. If the string includes spaces, enclose the string with quotes.

**Step 7** If you select IP Netmask for the sticky type:

- a. In the Static Source field, enter the source IP address of the client.
- b. In the Static Destination field, enter the destination IP address of the client.

**Step 8** In the Named Real Server field, select the real server to associate with this static sticky entry.

**Step 9** In the Port field, enter the port number of the real server. Valid entries are integers from 1 to 65535.

**Step 10** Click:

- **Deploy Now** to deploy this configuration on the ACE.
  - **Cancel** to exit the procedure without saving your entries and to return to the Sticky Statics table.
  - **Next** to deploy your entries and to configure another sticky static entry.
- 

#### Related Topic

[Configuring Sticky Groups, page 6-7](#)

