

Sticky Cookie Configuration Mode Commands

Sticky cookie configuration mode commands allow you to configure the ACE to either use the cookie learned from the server or inject a cookie to provide stickiness between a client and a server for the duration of a transaction. To configure the ACE to use HTTP cookies for stickiness, use the **sticky http-cookie** command in configuration mode. This command creates a sticky cookie group and allows you to access sticky cookie configuration mode. The prompt changes to (config-sticky-cookie). To remove the sticky cookie group from the configuration, use the **no** form of this command.

```
sticky http-cookie name1 name2
```

```
no sticky http-cookie name1 name2
```

Syntax Description

<i>name1</i>	Cookie value from the HTTP header of the client request or from the Set-Cookie message from the server. Enter a unique identifier for the cookie with a maximum of 64 alphanumeric characters.
<i>name2</i>	Unique identifier of the sticky group. Enter an unquoted text string with no spaces and a maximum of 64 alphanumeric characters.

Command Modes

Configuration mode
Admin and user contexts

Command History

Release	Modification
A1(7)	This command was introduced.

Usage Guidelines

The commands in this mode require the sticky feature in your user role. For details about role-based access control (RBAC) and user roles, see the *Cisco 4700 Series Application Control Engine Appliance Virtualization Configuration Guide*.

You can create a maximum of 4096 sticky groups in the ACE.

By default, the maximum number of bytes that the ACE parses to check for a cookie, HTTP header, or URL is 2048. 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 [\(config-parammap-http\) set header-maxparse-length](#) command in HTTP parameter-map configuration mode.

You can also change the default behavior of the ACE when a cookie, header, or URL exceeds the maximum parse length using the [\(config-parammap-http\) length-exceed](#) command in HTTP parameter-map configuration mode.

Examples

To create a sticky group for cookie stickiness, enter:

```
host1/Admin(config)# sticky http-cookie cisco.com GROUP3
host1/Admin(config-sticky-cookie)#
```

To remove the sticky group from the configuration, enter:

```
host1/Admin(config)# no sticky http-cookie cisco.com GROUP3
```

Related Commands

- [show running-config](#)
- [show sticky database](#)
- [\(config\) sticky http-header](#)
- [\(config\) sticky ip-netmask](#)

(config-sticky-cookie) cookie insert

To enable cookie insertion, use the **cookie insert** command. Use cookie insertion when you want to use a session cookie for persistence if the server is not currently setting the appropriate cookie. Use the **no** form of this command to disable cookie insertion.

cookie insert [browser-expire]

no cookie insert [browser-expire]

Syntax Description	browser-expire	(Optional) Allows the client's browser to expire a cookie when the session ends.
--------------------	----------------	--

Command Modes

Sticky cookie configuration mode
Admin and user contexts

Command History	Release	Modification
	A1(7)	This command was introduced.

Usage Guidelines

With cookie insertion enabled, the ACE inserts the cookie in the Set-Cookie header of the response from the server to the client. The ACE 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.

With either TCP server reuse or or persistence rebalance enabled, the ACE inserts a cookie in every client request. See the [\(config-parammap-http\) server-conn reuse](#) or [\(config-parammap-http\) persistence-rebalance](#) commands.

Examples

To specify a cookie for insertion:

```
host1/Admin(config-sticky-cookie)# cookie insert
```

To disable cookie insertion, enter:

```
host1/Admin(config-sticky-cookie)# no cookie insert
```

Related Commands

- [\(config\) sticky http-cookie](#)

(config-sticky-cookie) cookie offset

To configure the cookie offset and length, use the **cookie offset** command. Use the **no** form of this command to remove the cookie offset and length from the configuration.

cookie offset *number1* [**length** *number2*]

no cookie offset *number1* [**length** *number2*]

Syntax Description

offset <i>number1</i>	Specifies the portion of the cookie that the ACE uses to stick the client on a particular server by indicating the bytes to ignore starting with the first byte of the cookie. Enter an integer from 0 to 3999. The default is 0, which indicates that the ACE does not exclude any portion of the cookie.
length <i>number2</i>	(Optional) Specifies the length of the portion of the cookie (starting with the byte after the offset value) that the ACE uses for sticking the client to the server. Enter an integer from 1 to 4000. The default is 4000.

Command Modes

Sticky cookie configuration mode
Admin and user contexts

Command History

Release	Modification
A1(7)	This command was introduced.

Usage Guidelines

An HTTP-cookie value may change over time with only a portion remaining constant throughout a transaction between the client and a server. You can configure the ACE to use the constant portion of a cookie to make persistent connections to a specific server.

The ACE stores cookie offset and length values in the sticky table.

The offset and length can vary from 0 to 4000 bytes. If the cookie value is longer than the offset but shorter than the offset plus the length of the cookie, the ACE sticks the connection based on that portion of the cookie that starts with the byte after the offset value and ends with the byte specified by the offset plus the length. The total of the offset and the length cannot exceed 4000 bytes.

Examples

To configure the cookie offset and length, enter:

```
host1/Admin(config-sticky-cookie)# cookie offset 3000 length 1000
```

To remove the cookie offset and length from the configuration, enter:

```
host1/Admin(config-sticky-cookie)# no cookie offset 3000 length 1000
```

Related Commands

[\(config\) sticky http-cookie](#)

(config-sticky-cookie) cookie secondary

To configure a secondary cookie, use the **cookie secondary** command. Use the **no** form of this command to remove a secondary cookie from the configuration.

cookie secondary *name*

no cookie secondary

Syntax Description

<i>name</i>	Name of the secondary cookie. Enter a cookie name as an unquoted text string with no spaces and a maximum of 64 alphanumeric characters.
-------------	--

Command Modes

Sticky cookie configuration mode
Admin and user contexts

Command History

Release	Modification
A1(7)	This command was introduced.

Usage Guidelines

You can configure an alternative cookie name that appears in the URL string of the web page on the server. The ACE uses this cookie to maintain a sticky connection between a client and a server and adds a secondary entry in the sticky table.

Examples

To configure a secondary cookie, enter:

```
host1/Admin(config-sticky-cookie)# cookie secondary mysite.com
```

To remove a secondary cookie from the configuration, enter:

```
host1/Admin(config-sticky-cookie)# no cookie secondary
```

Related Commands

[\(config\) sticky http-cookie](#)

(config-sticky-cookie) replicate sticky

To instruct the ACE to replicate HTTP-cookie sticky table entries on the standby ACE, use the **replicate sticky** command. Use the **no** form of this command to restore the ACE to its default of not replicating HTTP-cookie sticky table entries.

replicate sticky

no replicate sticky

Command Modes

Sticky cookie configuration mode

Admin and user contexts

Command History

Release	Modification
A1(7)	This command was introduced.

Usage Guidelines

If you are using redundancy, you can configure the ACE to replicate HTTP-cookie sticky table entries on the standby ACE so that, if a switchover occurs, the new active ACE can maintain existing sticky connections.

The timer of a sticky table entry on the standby ACE is reset everytime the entry is synchronized with the active ACE entry. Thus, the standby sticky entry may have a lifetime up to twice as long as the active entry. However, if the entry expires on the active ACE or a new real server is selected and a new entry is created, the old entry on the standby ACE is replaced.

Examples

To instruct the ACE to replicate HTTP-cookie sticky table entries on the standby ACE, enter:

```
host1/Admin(config-sticky-cookie)# replicate sticky
```

To restore the ACE to its default of not replicating HTTP-cookie sticky table entries, enter:

```
host1/Admin(config-sticky-cookie)# no replicate sticky
```

Related Commands

[\(config\) sticky http-cookie](#)

(config-sticky-cookie) serverfarm

To complete a sticky group configuration, you must configure a server farm entry for the group. To configure a server farm entry for a sticky group, use the **serverfarm** command. Use the **no** form of this command to dissociate a server farm from a sticky group.

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

```
no serverfarm
```

Syntax Description

<i>name1</i>	Identifier of an existing server farm that you want to associate with the sticky group. You can associate one server farm with each sticky group. Enter a name as an unquoted text string with no spaces and a maximum of 64 alphanumeric characters.
backup <i>name2</i>	(Optional) Specifies the keyword and identifier of a backup server farm. If the primary server farm is down, the ACE uses the configured backup server farm. Once clients are stuck to a backup server farm, they remain stuck to the backup even if the primary server farm becomes active again. Enter a name as an unquoted text string with no spaces and a maximum of 64 alphanumeric characters.
sticky	(Optional) Specifies that the backup server farm is sticky.
aggregate-state	(Optional) Specifies that the state of the primary server farm is tied to the state of all the real servers in that server farm and in the backup server farm, if configured. The ACE declares the primary server farm down if all real servers in the primary server farm and all real servers in the backup server farm are down.

Command Modes

Sticky cookie configuration mode
Admin and user contexts

Command History

Release	Modification
A1(7)	This command was introduced.

Usage Guidelines

This command has no usage guidelines.

Examples

To associate a server farm with a sticky group and specify a sticky backup server farm, enter:

```
host1/Admin(config-sticky-cookie)# serverfarm SFARM1 backup BKUP_SFARM2 sticky aggregate-state
```

To dissociate a server farm from a sticky group, enter:

```
host1/Admin(config-sticky-cookie)# no serverfarm
```

Related Commands

[\(config\) sticky http-cookie](#)

(config-sticky-cookie) static cookie-value

To configure a static cookie, use the **static cookie-value** command. Use the **no** form of this command to remove a static cookie from the configuration.

```
[sequence_number] static cookie-value value rserver name [number]
```

```
no [sequence_number] static cookie-value value rserver name [number]
```

Syntax Description		
<i>sequence_number</i>	(Optional) Sequence number for a static cookie entry.	
<i>value</i>	Cookie string value. 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 string in quotation marks (“”).	
rserver name	Specifies the hostname of an existing real server. Enter an unquoted text string with no spaces and a maximum of 64 alphanumeric characters.	
<i>number</i>	(Optional) Port number of the real server. Enter an integer from 1 to 65535.	

Command Modes
Sticky cookie configuration mode
Admin and user contexts

Command History	Release	Modification
	A1(7)	This command was introduced.

Usage Guidelines
You can configure the ACE to use static cookies from entries based on cookie values and, optionally, real server names and ports. Static cookie values remain constant over time.
You can configure multiple static cookie entries, but only one unique real-server name can exist for a given static cookie value. When you configure a static entry, the ACE enters it into the sticky table immediately. You can create a maximum of 4096 static sticky entries in the ACE.

Examples
To configure a static cookie, enter:

```
host1/Admin(config-sticky-cookie)# static cookie-value CORVETTE rserver SERVER1 4000
```

 To remove a static cookie form the configuration, enter:

```
host1/Admin(config-sticky-cookie)# no static cookie-value CORVETTE rserver SERVER1 4000
```

Related Commands [\(config\) sticky http-cookie](#)

(config-sticky-cookie) timeout

To specify the duration for sticky connections between a client and a real server to persist, use the **timeout** command. Use the **no** form of this command to reset the sticky timeout to the default of 1440 minutes.

timeout *sticky-time* | **activeconns**

no timeout *sticky-time* | **activeconns**

Syntax Description		
<i>sticky-time</i>		Length of time in minutes that the ACE appliance remembers the last real server to which a client made a sticky connection. Enter an integer from 0 to 65535. The default timeout value is 1440 minutes (24 hours).
activeconns		Specifies that a sticky entry is timed out when the timer expires even if there are active connections associated with the sticky entry.

Command Modes	
	Sticky cookie configuration mode Admin and user contexts

Command History	Release	Modification
	A1(7)	This command was introduced.

Usage Guidelines	
	If a client initiates a new connection before the timer expires and the request matches the same server load balancing (SLB) policy as the previous connection, the ACE appliance load balances the client to the same server that was used to service the previous request.

Examples	
	To specify the duration for sticky connections between a client and a real server to persist, enter: <pre>host1/Admin(config-sticky-cookie)# timeout activeconns</pre>
	To restore the ACE to its default of not timing out HTTP-cookie sticky entries if active connections exist for those entries, enter: <pre>host1/Admin(config-sticky-cookie)# no timeout activeconns</pre>

Related Commands	
	(config) sticky http-cookie

Sticky Header Configuration Mode Commands

Sticky header configuration mode commands allow you to create an HTTP-header sticky group to enable the ACE to stick client connections to the same real server based on HTTP headers. To access sticky header configuration mode, use the **sticky http-header** command. The prompt changes to (config-sticky-header). Use the **no** form of this command to remove the sticky group from the configuration.

sticky http-header *name1 name2*

no sticky http-header *name1 name2*

Syntax Description	<i>name1</i>	<i>name2</i>
	HTTP header name. Enter an unquoted text string with no spaces and a maximum of 64 alphanumeric characters. Alternatively, you can enter one of the standard headers described in the “Usage Guidelines” section.	Unique identifier of the sticky group. Enter an unquoted text string with no spaces and a maximum of 64 alphanumeric characters.

Command Modes	Configuration mode Admin and user contexts
---------------	---

Command History	Release	Modification
	A1(7)	This command was introduced.

Usage Guidelines The commands in this mode require the sticky feature in your user role. For details about role-based access control (RBAC) and user roles, see the *Cisco 4700 Series Application Control Engine Appliance Virtualization Configuration Guide*.

You can create a maximum of 4096 sticky groups in the ACE.

By default, the maximum number of bytes that the ACE parses to check for a cookie, HTTP header, or URL is 2048. 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.

You can also change the default behavior of the ACE when a cookie, header, or URL exceeds the maximum parse length using the **length-exceed** command in HTTP parameter-map configuration mode.

[Table 2-11](#) lists and describes the standard HTTP header names.

Table 2-11 HTTP Header Names

Field Name	Description
Accept	Semicolon-separated list of representation schemes (content type metainformation values) that will be accepted in the response to the request.
Accept-Charset	Character sets that are acceptable for the response. This field allows clients that can understand more comprehensive or special-purpose character sets to signal that capability to a server that can represent documents in those character sets.
Accept-Encoding	Restricts the content encoding that a user will accept from the server.
Accept-Language	ISO code for the language in which the document is written. The language code is an ISO 3316 language code with an optional ISO 639 country code to specify a national variant.
Authorization	Specifies that the user agent wants to authenticate itself with a server, usually after receiving a 401 response.
Cache-Control	Directives that must be obeyed by all caching mechanisms on the request/response chain. The directives specify behavior intended to prevent caches from adversely interfering with the request or response.
Connection	Allows the sender to specify connection options.
Content-MD5	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.
Expect	Used by a client to inform the server about the behaviors that the client requires.
From	E-mail address of the person who controls the requesting user agent.
Host	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.
If-Match	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 asterisk (*) value matches any current entity of the resource.
Pragma	Pragma directives that are understood by servers to which the directives are relevant. The syntax is the same as for other multiple-value fields in HTTP. For example, the accept field is a comma-separated list of entries for which the optional parameters are separated by semicolons.
Referer	Address (URI) of the resource from which the URI in the request was obtained.
Transfer-Encoding	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.

Table 2-11 HTTP Header Names (continued)

Field Name	Description
User-Agent	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 for tailoring responses to avoid user agent limitations.
Via	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.

Examples

To create a group for HTTP-header stickiness, enter:

```
host1/Admin(config-sticky-header) # sticky http-header Host GROUP4
```

To remove the sticky group from the configuration, enter:

```
host1/Admin(config-sticky-header) # no sticky http-header Host GROUP4
```

Related Commands

[show running-config](#)
[show sticky database](#)
[\(config\) sticky http-cookie](#)
[\(config\) sticky ip-netmask](#)

(config-sticky-header) header offset

To configure the HTTP-header offset and length, use the **header offset** command. Use the **no** form of this command to remove the HTTP-header offset and length values from the configuration.

```
header offset number1 [length number2]
```

```
no header offset number1 [length number2]
```

Syntax Description

offset <i>number1</i>	Specifies the portion of the HTTP header that the ACE uses to stick the client on a particular server by indicating the bytes to ignore starting with the first byte of the HTTP header. Enter an integer from 0 to 31. The default is 0, which indicates that the ACE does not exclude any portion of the header.
length <i>number2</i>	(Optional) Specifies the length of the portion of the HTTP header (starting with the byte after the offset value) that the ACE uses for sticking the client to the server. Enter an integer from 1 to 32. The default is 32.

Command Modes

Sticky header configuration mode

Admin and user contexts

Command History	Release	Modification
	A1(7)	This command was introduced.

Usage Guidelines

The ACE stores header offset and length values in the sticky table.

You can configure the ACE to use a portion of an HTTP header to make persistent connections to a specific server. To define the portion of the HTTP header that you want the ACE to use, you specify HTTP-header offset and length values. The offset and length can vary from 0 to 32 bytes. The ACE sticks the connection based on that portion of the HTTP header that starts with the byte after the offset value and ends with the byte specified by the offset plus the length. The total of the offset and the length cannot exceed 32 bytes.

Examples

To configure the header offset and length, enter:

```
host1/Admin(config-sticky-header) # header offset 3000 length 1000
```

To remove the HTTP-header offset and length values from the configuration, enter:

```
host1/Admin(config-sticky-header) # no header offset 3000 length 1000
```

Related Commands

[\(config\) sticky http-header](#)

(config-sticky-header) replicate sticky

To instruct the ACE to replicate HTTP-header sticky table entries on the standby ACE, use the **replicate sticky** command. Use the **no** form of this command to restore the ACE to its default of not replicating HTTP-header sticky table entries.

replicate sticky

no replicate sticky

Syntax Description

This command has no keywords or arguments.

Command Modes

Sticky header configuration mode

Admin and user contexts

Command History	Release	Modification
	A1(7)	This command was introduced.

Usage Guidelines

If you are using redundancy, you can configure the ACE to replicate HTTP-header sticky table entries on the standby ACE so that, if a switchover occurs, the new active ACE can maintain existing sticky connections.

The timer of a sticky table entry on the standby ACE is reset everytime the entry is synchronized with the active ACE entry. Thus, the standby sticky entry may have a lifetime up to twice as long as the active entry. However, if the entry expires on the active ACE or a new real server is selected and a new entry is created, the old entry on the standby ACE is replaced.

Examples

To instruct the ACE to replicate HTTP-header sticky table entries on the standby ACE, enter:

```
host1/Admin(config-sticky-header)# replicate sticky
```

To restore the ACE to its default of not replicating HTTP-header sticky table entries, enter:

```
host1/Admin(config-sticky-header)# no replicate sticky
```

Related Commands

[\(config\) sticky http-header](#)

(config-sticky-header) serverfarm

To complete a sticky group configuration, you must configure a server farm entry for the group. To configure a server farm entry for a sticky group, use the **serverfarm** command. Use the **no** form of this command to dissociate a server farm from a sticky group.

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

```
no serverfarm
```

Syntax Description

<i>name1</i>	Identifier of an existing server farm that you want to associate with the sticky group. You can associate one server farm with each sticky group. Enter a name as an unquoted text string with no spaces and a maximum of 64 alphanumeric characters.
backup <i>name2</i>	(Optional) Specifies the keyword and identifier of a backup server farm. If the primary server farm is down, the ACE uses the configured backup server farm. Once clients are stuck to a backup server farm, they remain stuck to the backup even if the primary server farm becomes active again. Enter a name as an unquoted text string with no spaces and a maximum of 64 alphanumeric characters.
sticky	(Optional) Specifies that the backup server farm is sticky.
aggregate-state	(Optional) Specifies that the state of the primary server farm is tied to the state of all the real servers in that server farm and in the backup server farm, if configured. The ACE declares the primary server farm down if all real servers in the primary server farm and all real servers in the backup server farm are down.

Command Modes Sticky header configuration mode
Admin and user contexts

Command History	Release	Modification
	A1(7)	This command was introduced.

Usage Guidelines This command has no usage guidelines.

Examples To associate a server farm with a sticky group and specify a sticky backup server farm, enter:

```
host1/Admin(config-sticky-header) # serverfarm SFARM1 backup BKUP_SFARM2 sticky aggregate-state
```

To dissociate a server farm from a sticky group, enter:

```
host1/Admin(config-sticky-header) # no serverfarm
```

Related Commands [\(config\) serverfarm](#)
[\(config\) sticky http-header](#)

(config-sticky-header) static header-value

To configure a static header, use the **static header-value** command. Use the **no** form of this command to remove a static header from the configuration.

```
[sequence_number] static header-value value rserver name [number]
```

```
no [sequence_number] static header-value value rserver name [number]
```

Syntax Description	
<i>sequence_number</i>	(Optional) Sequence number for a static cookie entry.
<i>value</i>	Header string value. 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 string in quotation marks ("").
rserver name	Specifies the hostname of an existing real server. Enter an unquoted text string with no spaces and a maximum of 64 alphanumeric characters.
<i>number</i>	(Optional) Port number of the real server. Enter an integer from 1 to 65535.

Command Modes Sticky header configuration mode
Admin and user contexts

Command History	Release	Modification
	A1(7)	This command was introduced.

Usage Guidelines

You can configure the ACE to use static headers from entries based on header values and, optionally, real server names and ports. Static header values remain constant over time.

You can configure multiple static header entries, but only one unique real-server name can exist for a given static header value. When you configure a static entry, the ACE enters it into the sticky table immediately. You can create a maximum of 4096 static sticky entries in the ACE.

Examples

To configure a static header, enter:

```
host1/Admin(config-sticky-header) # static header-value CORVETTE rserver SERVER1 4000
```

To remove a static header from the configuration, enter:

```
host1/Admin(config-sticky-header) # no static header-value CORVETTE rserver SERVER1 4000
```

Related Commands [\(config\) sticky http-header](#)

(config-sticky-header) timeout

To specify the duration for sticky connections between a client and a real server to persist, use the **timeout** command. Use the **no** form of this command to reset the sticky timeout to the default of 1440 minutes.

timeout *sticky-time* | **activeconns**

no timeout *sticky-time* | **activeconns**

Syntax Description	
<i>sticky-time</i>	Length of time in minutes that the ACE appliance remembers the last real server to which a client made a sticky connection. Enter an integer from 1 to 65535. The default timeout value is 1440 minutes (24 hours).
activeconns	Specifies that sticky entries are timed out when the timer expires even if there are active connections.

Command Modes

Sticky header configuration mode
Admin and user contexts

Command History	Release	Modification
	A1(7)	This command was introduced.

Usage Guidelines

If a client initiates a new connection before the timer expires and the request matches the same server load balancing (SLB) policy as the previous connection, the ACE appliance load balances the client to the same server that was used to service the previous request.

Examples

To specify the duration for sticky connections between a client and a real server to persist, enter:

```
host1/Admin(config-sticky-header) # timeout activeconns
```

To restore the ACE to its default of not timing out HTTP-header sticky entries if active connections exist for those entries, enter:

```
host1/Admin(config-sticky-header) # no timeout activeconns
```

Related Commands

[\(config\) sticky http-header](#)

Sticky IP Configuration Mode Commands

Sticky IP configuration mode commands allow you to create a sticky group for IP address stickiness. To create a sticky group and access sticky IP configuration mode, use the **sticky ip-netmask** command. The prompt changes to (config-sticky-ip). You can create a maximum of 4000 sticky groups on an ACE. Use the **no** form of this command to remove the sticky group from the configuration.

sticky ip-netmask *netmask* **address** {**source** | **destination** | **both**} *name*

no sticky ip-netmask *netmask* **address** {**source** | **destination** | **both**} *name*

Syntax Description	<i>netmask</i>	Network mask that the ACE applies to the IP address. Enter a network mask in dotted-decimal notation (for example, 255.255.255.0).
	address { source destination both }	Specifies the IP address used for stickiness. Enter one of the following keywords: <ul style="list-style-type: none"> • source—Specifies that the ACE use the client source IP address to stick the client to a server. You use this keyword in web application environments. • destination—Specifies that the ACE use the destination address specified in the client request to stick the client to a server. You use this keyword in caching environments. • both—Specifies that the ACE use both the source IP address and the destination IP address to stick the client to a server.
	<i>name</i>	Unique identifier of the sticky group. Enter an unquoted text string with no spaces and a maximum of 64 alphanumeric characters.

Command Modes	Configuration mode Admin and user contexts
----------------------	---

Command History	Release	Modification
	A1(7)	This command was introduced.

Usage Guidelines The commands in this mode require the sticky feature in your user role. For details about role-based access control (RBAC) and user roles, see the *Cisco 4700 Series Application Control Engine Appliance Virtualization Configuration Guide*.

You can create a maximum of 4096 sticky groups in the ACE.

Examples

To create a sticky group that uses IP address stickiness based on both the source IP address and the destination IP address, enter:

```
host1/Admin(config)# sticky ip netmask 255.255.255.0 address both GROUP1
host1/Admin(config-sticky-ip)#
```

To remove the sticky group from the configuration, enter:

```
host1/Admin(config)# no sticky ip netmask 255.255.255.0 address both GROUP1
```

Related Commands

[show running-config](#)
[show sticky database](#)
[\(config\) sticky http-cookie](#)
[\(config\) sticky http-header](#)

(config-sticky-ip) replicate sticky

To instruct the ACE to replicate IP address sticky table entries on the standby ACE, use the **replicate sticky** command. Use the **no** form of this command to restore the ACE to its default of not replicating IP address sticky table entries.

replicate sticky

no replicate sticky

Syntax Description

This command has no keywords or arguments.

Command Modes

Sticky IP configuration mode

Admin and user contexts

Command History

Release	Modification
A1(7)	This command was introduced.

Usage Guidelines

If you are using redundancy, you can configure the ACE to replicate IP address sticky table entries on the standby ACE so that, if a switchover occurs, the new active ACE can maintain existing sticky connections.

The timer of a sticky table entry on the standby ACE is reset everytime the entry is synchronized with the active ACE entry. Thus, the standby sticky entry may have a lifetime up to twice as long as the active entry. However, if the entry expires on the active ACE or a new real server is selected and a new entry is created, the old entry on the standby ACE is replaced.

Examples

To instruct the ACE to replicate IP address sticky table entries on the standby ACE, enter:

```
host1/Admin(config-sticky-ip)# replicate sticky
```

To restore the default of the ACE to not replicate IP address sticky table entries, enter:

```
host1/Admin(config-sticky-ip)# no replicate sticky
```

Related Commands [\(config\) sticky ip-netmask](#)**(config-sticky-ip) serverfarm**

To complete a sticky group configuration, you must configure a server farm entry for the group. To configure a server farm entry for a sticky group, use the **serverfarm** command. Use the **no** form of this command to dissociate a server farm from a sticky group.

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

```
no serverfarm
```

Syntax Description

<i>name1</i>	Identifier of an existing server farm that you want to associate with the sticky group. You can associate one server farm with each sticky group. Enter a name as an unquoted text string with no spaces and a maximum of 64 alphanumeric characters.
backup <i>name2</i>	(Optional) Specifies the keyword and identifier of a backup server farm. If the primary server farm is down, the ACE uses the configured backup server farm. Once clients are stuck to a backup server farm, they remain stuck to the backup even if the primary server farm becomes active again. Enter a name as an unquoted text string with no spaces and a maximum of 64 alphanumeric characters.
sticky	(Optional) Specifies that the backup server farm is sticky.
aggregate-state	(Optional) Specifies that the state of the primary server farm is tied to the state of all the real servers in that server farm and in the backup server farm, if configured. The ACE declares the primary server farm down if all real servers in the primary server farm and all real servers in the backup server farm are down.

Command Modes

Sticky IP configuration mode
Admin and user contexts

Command History

Release	Modification
A1(7)	This command was introduced.

Usage Guidelines This command has no usage guidelines.

Examples To associate a server farm with a sticky group and specify a sticky backup server farm, enter:

```
host1/Admin(config-sticky-ip)# serverfarm SFARM1 backup BKUP_SFARM2 sticky aggregate-state
```

To dissociate a server farm from a sticky group, enter:

```
host1/Admin(config-sticky-ip)# no serverfarm
```

Related Commands [\(config\) sticky ip-netmask](#)

(config-sticky-ip) static client source

To configure static sticky-IP table entries, use the **static client** command. Use the **no** form of this command to remove the static entry from the sticky table.

The syntax of this command varies according to the **address** option that you chose when you created the sticky group using the [\(config\) sticky ip-netmask](#) command. If you configured the sticky group with the **source** option, the syntax of this command is as follows:

```
static client source ip_address rserver name [number]
```

```
no static client source ip_address rserver name [number]
```

If you configured the sticky group with the **destination** option, the syntax of this command is as follows:

```
static client source ip_address destination ip_address rserver name [number]
```

```
no static client source ip_address destination ip_address rserver name [number]
```

If you configured the sticky group with the **both** option, the syntax of this command is as follows:

```
static client source ip_address [destination ip_address] rserver name [number]
```

```
no static client source ip_address [destination ip_address] rserver name [number]
```

Syntax Description		
source <i>ip-address</i>		Specifies that the static entry is based on the source IP address. Enter an IP address in dotted-decimal notation (for example, 192.168.12.15).
rserver <i>name</i>		Specifies that the static entry is based on the real server name. Enter the name of an existing real server as an unquoted text string with no spaces and a maximum of 64 alphanumeric characters.
<i>number</i>		(Optional) Port number of the real server. Enter an integer from 1 to 65535.
destination <i>ip-address</i>		Specifies that the static entry is based on the destination IP address. Enter an IP address in dotted-decimal notation (for example, 172.16.27.3).

Command Modes Sticky IP configuration mode
Admin and user contexts

Command History	Release	Modification
	A1(7)	This command was introduced.

Usage Guidelines You can configure static sticky table entries based on the source IP address, the destination IP address, or the real server name and port. Static sticky-IP values remain constant over time and you can configure multiple static entries. When you configure a static entry, the ACE enters it into the sticky table immediately. You can configure a maximum of 4096 static sticky entries in the ACE.

Examples To configure a static sticky entry based on the source IP address, the destination IP address, and the server name and port number, enter:

```
host1/Admin(config-sticky-ip)# static client source 192.168.12.15 destination 172.16.27.3
rserver SERVER1 2000
```

To remove the static entry from the sticky table, enter:

```
host1/Admin(config-sticky-ip)# no static client source 192.168.12.15 destination
172.16.27.3 rserver SERVER1 2000
```

Related Commands [\(config\) sticky ip-netmask](#)

(config-sticky-ip) timeout

To configure an IP address sticky timeout, use the **timeout** *minutes* command. Use the **no** form of this command to reset the sticky timeout to the default of 1440 minutes (24 hours).

timeout *sticky-time* | **activeconns**

no timeout *sticky-time* | **activeconns**

Syntax Description	
<i>sticky-time</i>	Length of time in minutes that the ACE appliance remembers the last real server to which a client made a sticky connection. Enter an integer from 1 to 65535. The default timeout value is 1440 minutes (24 hours).
activeconns	Specifies that sticky entries are timed out when the timer expires even if there are active connections.

Command Modes Sticky IP configuration mode.
Admin and user contexts

Command History

Release	Modification
A1(7)	This command was introduced.

Usage Guidelines

The sticky timeout specifies the period of time that the ACE keeps (if possible) the IP address sticky information for a client connection in the sticky table after the latest client connection terminates. The ACE resets the sticky timer for a specific sticky-table entry each time that the appliance opens a new connection or receives a new HTTP GET on an existing connection matching that entry. High connection rates may cause the sticky table entries to age out prematurely.

By default, the ACE ages out a sticky table entry when the timeout for that entry expires and no active connections matching that entry exist. To specify that the ACE time out IP address sticky table entries even if active connections exist after the sticky timer expires, use the **timeout activeconns** command.

Examples

To specify a timeout value of 720 minutes, enter:

```
host1/Admin(config-sticky-ip)# timeout 720
```

To specify that the ACE time out IP address sticky table entries even if active connections exist after the sticky timer expires, enter:

```
host1/Admin(config-sticky-ip)# timeout activeconns
```

To restore the ACE to its default of not timing out IP address sticky entries if active connections exist, enter:

```
host1/Admin(config-sticky-ip)# no timeout activeconns
```

Related Commands

[\(config\) sticky ip-netmask](#)