



# Configuring Server/Application State Protocol for Server Load Balancing

---

The Server/Application State Protocol (SASP), developed by IBM, provides a mechanism for global workload managers (GWM) to monitor backend servers and their applications. The GWM uses SASP to send updated service weights to the CSS for load balancing decisions. This chapter will refer to the GWM as the SASP agent.

This chapter contains the following major sections:

- [SASP Overview](#)
- [Configuring SASP](#)
- [Displaying SASP Information](#)

Information in this chapter applies to all CSS models except where noted.



**Note**

---

You cannot configure SASP and DFP on the CSS at the same time.

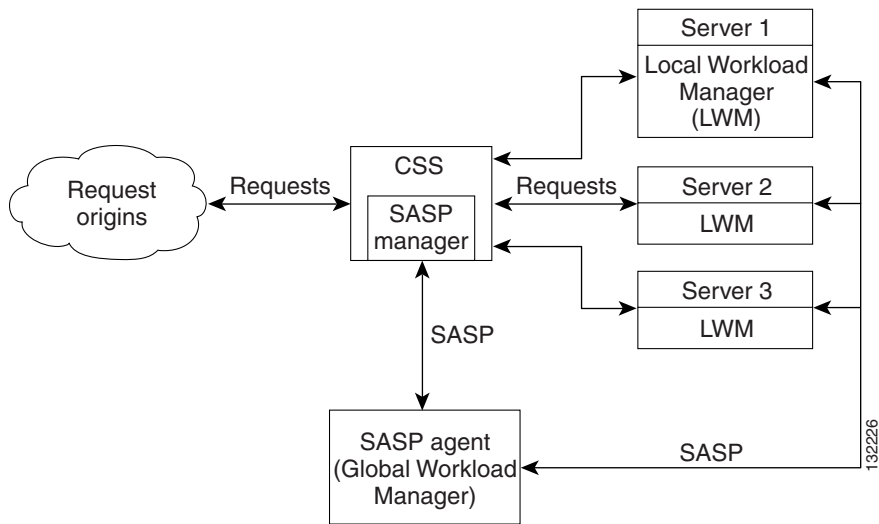
---

For more information on CSS server load-balancing, refer to [Chapter 10, Configuring Content Rules](#).

# SASP Overview

Figure 7-1 shows the CSS load-balancing requests across three servers.

**Figure 7-1 Example of SASP Functional Overview**



The SASP feature allows you to configure the CSS to act as an SASP manager. The SASP manager is responsible for interpreting SASP messages sent from a SASP agent. The SASP manager communicates with the agent across an unencrypted unsecured TCP connection while the agent monitors the associated CSS services.

For an agent to monitor CSS services, you must configure the services on a content rule and then associate the agent with the rule. When you associate an agent with a content rule, services configured on the rule become an SASP group. The SASP manager informs the SASP agent of the services in the SASP group.

Each service in the group is an SASP member. As an SASP member, each service has a local workload manager (LWM) that communicates the status of the service to the SASP agent. As the agent monitors the status of the CSS services, it reports service weights through SASP messages to the SASP manager on the CSS. The manager informs the CSS of the updated service weights from the SASP agent. Then, the CSS uses the weight information to modify the load-balancing distribution.

## SASP Messages

For communications between the SASP agent and CSS SASP manager, SASP has five pairs of request and reply messages, and a single Send Weights Message. These messages are:

- Registration Request and Reply
- Deregistration Request and Reply
- Get Weights Request and Reply
- Set Load Balancer State Request and Reply
- Set Member State Request and Reply
- Send Weights Message

The CSS SASP manager uses the Registration request to inform the SASP agent that members are added to a group. Likewise, the Deregistration request informs the agent when members are removed from the group. The response for both the Registration and Deregistration messages inform the SASP manager of a successful or failed request.

The SASP manager issues the Get Weights request to the SASP agent to retrieve the most recent weight information for the members of a group. The response from the agent should contain all of the requested weight information.

The Send Weights message is similar to the Get Weights response as this message contains member weights. However, by default, the SASP manager receives weights from the SASP agent within a time interval determined by the agent. If the agent does not provide updated weights with the interval, the SASP manager sends a Set Load Balancer Request to validate that the agent is functioning.

The SASP manager sends a message every 30 seconds until it receives a response from the agent. If two attempts fail, the SASP manager reverts all service weights reported by the agent to their default weights. After five unanswered responses, the SASP manager terminates the connection. Then the SASP Manager waits 60 seconds before trying to reconnect to the SASP agent.

The SASP manager uses the Set Load Balancer request to inform the agent of the CSS health and the current options. The SASP manager also uses the request as a keepalive in the event that it does not receive regular weight updates from the agent. The Set Load Balancer response from the agent indicates a failure or successful request.

The SASP manager uses the Set Member State request to notify the agent of the service availability. The service is either up or down, as per the CSS. The response to this message indicates a successful or failed request.

## Weight Precedence and Range

There are three different ways that you can set weights on the CSS. You can set the weight on the service, the content rule, or through a reporting mechanism such as SASP or DFP. The weight precedence on the CSS, starting with the most precedent to least precedent, is:

- Weight on the content rule
- Reported weight through SASP or DFP
- Weight on the service

**Note**

---

Because the content rule weight takes precedence over any other weight, if you add a service to a content rule with a weight, the reported weight by the SASP agent is not used. However, if you configure SASP on this rule, the service is associated with the SASP agent and cannot be used on another agent.

---

By default, the CSS can accept an SASP reported weight in the range of 0 to 64. However, the CSS weight range is 0 to 10. Due to this differences in weight ranges, the CSS performs a weight scaling conversion calculation that maps the SASP weight range to the CSS weight range. By default, the CSS weight scaling formula is:

$$\text{CSS Weight} = (\text{SASP Weight} * 10) / 64 + 1 \text{ (SASP Weight in range from 1 to 63)}$$

**Note**

---

If the agent reports a weight of zero and the converted weight is the value of one, the CSS weight is zero. If the agent reports a weight of 64 and the converted weight is the value of 11, the CSS weight is 10.

---

You can change the maximum weight that the CSS can accept from an SASP agent and thus change the CSS weight scaling for SASP. For more information, see the [“Changing the Maximum Accepted SASP Weight”](#) section.

# Configuring SASP

To configure the CSS to use SASP, you must:

- Configure the CSS as a valid load balancer on the SASP agent before you can configure the agent on the CSS. For more information, refer to the SASP documentation that came with your SASP agent.
- Configure the SASP agent.
- Associate the SASP agent with the content rule.
- Enable SASP on the CSS.

The following sections provide information to configure SASP on the CSS:

- [Configuring an SASP Agent](#)
- [Configuring an SASP Agent on a Content Rule](#)
- [Enabling and Disabling SASP on the CSS](#)
- [Changing the Maximum Accepted SASP Weight](#)

## Configuring an SASP Agent

An SASP agent monitors CSS services and reports weights to the SASP manager on the CSS. When you configure the agent, you configure its IP address and TCP listening port from which the CSS will retrieve weight information for services. You will also configure a label for the agent that you will use to add the agent to a content rule.



### Note

---

You must configure the CSS as a valid load balancer on the SASP agent before you can configure the agent on the CSS.

---

To configure the SASP agent, use the global configuration mode **sasp-agent** command. You can configure a total of eight agents on the CSS. The syntax for this command is:

```
sasp-agent label ip_address port
```

The variables are:

- *label* - The label that identifies the SASP agent when associating the agent with a content rule. Enter a text string with a maximum of 31 alphanumeric characters, including hyphen (-) and underscore (\_) characters, with no spaces.
- *ip\_address* - The IP address for the SASP agent. Enter an IP address in dotted-decimal notation (for example, 192.168.11.1).
- *port* - The TCP listening port for the agent. Enter a number from 1 to 65535.

For example, to configure an SASP agent with a label of agent1, an IP address of 14.3.3.10, and a port number of 3500, enter:

```
(config)# sasp-agent agent1 14.3.3.10 3500
```

If you enter a label that already exists, the following message appears:

```
%% The sasp-agent label is already in use
```

If you enter a duplicate IP address and port number, the following message appears:

```
%% The sasp-agent is already configured
```

If you enter an overlapping IP address, the following message appears:

```
%% SASP Agent IP address conflicts with local IP address
```

You can only delete an SASP agent from the CSS after you remove the agent from all associated content rules. For information on removing an agent from a rule, see the [“Configuring an SASP Agent on a Content Rule”](#) section. If you do not remove the agent from the content rules, the following error message appears:

```
%% The sasp-agent is still monitoring services
```

To delete the SASP agent, enter the agent label, for example:

```
(config)# no sasp-agent agent1
```

You can also delete the SASP agent by entering the agent label, IP address, and port number, for example:

```
(config)# no sasp-agent agent1 14.3.3.10 3500
```

## Configuring an SASP Agent on a Content Rule

After you configure an SASP agent on the CSS, you can add it to the content rule. When you add the agent to a rule, the CSS populates the agent group with the services configured on the rule.

Make sure that the rule load-balancing method is weighted round-robin. The following message appears if the rule is configured with another type of method:

```
%% WeightedRR should be configured with SASP.
```

A service can only be associated with one SASP agent. If you try to configure a SASP agent on a content rule containing a service that is configured on another content rule and SASP agent, the following error message appears:

```
%% One or more services contained in the content rule are associated with a different sasp-agent
```

If you try to configure a service on a content rule that is configured with an SASP agent and the service is associated with another SASP agent, the follow error message appears:

```
%% The service is already associated with a different sasp-agent
```

To add an SASP agent to a content rule, use the **add sasp-agent** command in owner-content configuration mode. The syntax for the command is:

```
add sasp-agent label
```

The *label* variable is the label configured for the SASP agent through the global configuration **sasp-agent** command.

For example, to add the SASP agent with the label agent1 to the content rule sasp-rule1, enter:

```
(config-owner-content [sasp-rule1])# add sasp-agent agent1
```

If the SASP agent does not exist, the following message appears:

```
%% The sasp-agent does not exist.
```

**Note**

The CSS identifies the SASP member by the IP address, port, and protocol type. If you configure services with these identical parameters on a content rule, you cannot add the SASP agent to the rule. The following error appears:

```
%% A duplicate SASP Member would be created from the list of services
```

If you try to add a service to a content rule that has a current rule with the same parameters, and the rule is configured with a SASP agent, the following error message:

```
%% Adding this service would create a duplicate SASP member
```

To remove the SASP agent agent1 from the content rule sasp-rule1, enter:

```
(config-owner-content [sasp-rule1])# remove sasp-agent
```

## Enabling and Disabling SASP on the CSS

To allow the CSS to communicate with the SASP agent, enable SASP on the CSS to start the CSS SASP manager. You can enable or disable the SASP functionality without removing all of the SASP agents and removing the content rule associations.

**Note**

SASP and DFP cannot be configured on the CSS at the same time. If you try to enable SASP on the CSS when DFP is already enabled, the following error message appears:

```
%% A DFP agent is already configured; SASP and DFP cannot be run simultaneously
```



To enable SASP, use the global configuration mode **sasp** command. The syntax for the command is:

```
sasp {unique-id "name"}
```

The CSS requires an identifier (ID) when communicating with an SASP agent. By default, when you enable the CSS and do not define an ID, the CSS uses the ID of Cisco-CSS and appends the last three octets of its base MAC address to it. For example, for a CSS with a MAC address of AA:BB:CC:DD:EE:FF, the corresponding default ID is Cisco-CSS\_DD-EE-FF. To enable the CSS with a default ID, enter:

```
(config)# sasp
```

You can also define a unique SASP ID for the CSS by using the **unique-id** option and **"name"** variable. Enter a name as a quoted text string with a maximum of 63 alphanumeric characters including hyphen (-) and underscore (\_) characters. The name cannot contain spaces. You cannot use control and special characters or the following message occurs:

```
%% Invalid character in SASP unique-identifier
```

For example, to define an SASP name of Cisco-CSS11506\_UID-001-001-1031, enter:

```
(config)# sasp unique-id "Cisco-CSS11506_UID-001-001-1031"
```

If the SASP manager receives errors indicating that there is a duplicate SASP ID, it closes the connection. You can reestablish the connection by disabling SASP and then reenabling it.

**Note**

---

To support box-to-box and VIP redundancy, each CSS must have a unique SASP ID. The `commit_redundancy` script copies all of the SASP commands and does not copy the SASP ID.

---

You can disable SASP without removing all of the associated configuration commands. To disable SASP on the CSS, enter:

```
(config)# no sasp
```

**Note**

When you disable SASP on the CSS, the SASP ID is no longer configured. When you reenable SASP, you can enable SASP with the default ID or configure a unique ID again.

## Changing the Maximum Accepted SASP Weight

By default, the CSS supports a maximum weight of 64 reported by the SASP agent. Because the CSS supports a weight range of 0 to 10, the CSS performs a weight scaling conversion calculation that maps the SASP weight range to the CSS weight range. If the SASP agent reports weights higher than 64, the CSS maps them to a weight of 10. If the SASP agent reports a maximum weight less than 64, the full weight range of the CSS will not be used.

If you change the maximum weight that the CSS accepts, the weight scaling conversion calculation that maps the SASP weight range to the CSS weight range changes accordingly. To configure the maximum weight that an SASP agent can report to the CSS, use the **sasp max-agent-weight** command. The syntax for this command is:

```
sasp max-agent-weight weight
```

The *weight* variable is the maximum weight that the SASP agent can report to the CSS. Enter a number from 1 to 65535. The default is 64.

For example, to change the maximum weight to 50, enter:

```
(config)# sasp-agent max-agent-weight 50
```

To reset the maximum weight to 64, enter:

```
(config)# no sasp-agent max-agent-weight
```

# Displaying SASP Information

You can display all of the SASP agents configured on the CSS, or weight and statistical data for services associated with an agent. The following sections provide this information:

- [Displaying the SASP Manager Configuration on the CSS](#)
- [Displaying SASP Agent Configuration and States](#)
- [Displaying Service Information and Reports](#)

You can also view whether SASP is enabled for reporting the weight of a service.

- Use the **show service** command to display service-specific information. The **show service** command output includes a Weight Reporting field that indicates whether weight reporting through SASP is enabled. For details on the **show service** command, see the “[Showing Service Configurations](#)” section in [Chapter 3, Configuring Services](#).
- Use the **show rule owner rule service** command in global configuration mode or the **show rule** command in owner-content configuration modes to display content rule information. The outputs of these commands display the weight assigned to each service preceded by a code letter. If the code letter is A, an SASP agent is reporting the weight for the service. For details on the **show rule** command, see [Chapter 10, Configuring Content Rules](#).

The **show service** and **show rule** commands display SASP weight information when the following conditions exist:

- SASP is enabled.
- The SASP member (the owner, content rule, and service) is registered with the SASP agent.
- The SASP agent reported weight for the SASP member is confident (accurate).
- The SASP agent and the CSS do not have discrepancies with the registration and quiesced flag values.

## Displaying the SASP Manager Configuration on the CSS

To view the SASP manager configuration on the CSS, use the **show sasp** command. This command displays whether SASP is enabled or disabled on the CSS, the state of the SASP manager, and the label assigned to the SASP manager on the CSS. The syntax for this command is:

**show sasp**

For example, to display the SASP configuration on the CSS, enter:

```
# show sasp
```

[Table 7-1](#) describes the fields in the **show sasp** command output.

**Table 7-1** *Field Descriptions for the show sasp Command Output*

Field	Description
SASP Configuration	Indicates whether SASP is enabled or disabled on the CSS. The values are Enabled or Disabled.
LB Health	The state of health for the CSS as reported to the SASP agent in the Set Load Balancer State Message.
LB UID	The SASP ID label configured for CSS when the CSS is enabled.
LB Redundancy	The CSS redundancy state. The possible states are master, backup, or other. The other state appears when the state of the CSS could not be determined. For example, the CSS is master for some redundant VIPs but is backup for other redundant VIPs.

## Displaying SASP Agent Configuration and States

To view the configuration and state information for all SASP agents on a CSS, use the **show sasp-agent-summary** command. This command displays a list of each configured SASP agent including the agent IP address, port number, the current state of the connection, the number of services assigned to the agent, and whether the services are active or suspended. The syntax for this command is:

```
show sasp-agent-summary
```

For example, to display all configured SASP agents, enter:

```
# show sasp-agent-summary
```

[Table 7-2](#) describes the fields in the **show sasp-agent-summary** command output.

**Table 7-2** *Field Descriptions for the show sasp-agent-summary Command Output*

Field	Description
Label	The label configured for the SASP agent.
IP Address	The IP address of the SASP agent for weight reporting.
Port	The port number that the CSS uses when connecting to the configured agent.

**Table 7-2** *Field Descriptions for the show sasp-agent-summary Command Output (continued)*

Field	Description
State	<p>The state of the connection between the SASP agent and CSS. Possible states are:</p> <ul style="list-style-type: none"> <li>• Initialized - The SASP agent is created.</li> <li>• Closed - The SASP agent is disabled but not removed from the CSS.</li> <li>• Trying - The agent tries to open a socket to the SASP manager on the CSS.</li> <li>• Connecting - The SASP manager on the CSS opens the socket and is connecting to the SASP agent.</li> <li>• Established - An agent is communicating with the SASP manager.</li> <li>• DownTrying - A communication failure occurred between the SASP agent and CSS, and the CSS is attempting to reconnect to the agent.</li> <li>• DownDuplicate - The agent detects that a load balancer has a duplicate SASP ID is on the network.</li> <li>• Down - A communication failure occurred and the CSS is unable to reestablish the connection with the agent. The agent will be in this state for 60 seconds after a failure.</li> <li>• Other - This state occurs when no other state applies to the condition.</li> </ul>
Last State Change	The last time that the agent had a state change.
Services	<p>The number of active and inactive services configured for the agent that manages their weights. An inactive service is a service suspended on the CSS.</p> <p>The number on the left indicates the active services. The number on the right indicates the inactive services.</p>

## Displaying Service Information and Reports

To view information for services that are members of an SASP agent, and SASP agent statistics, use the **show sasp-agent** *label* command. This command displays:

- The state of the agent
- Information about the active and inactive services that are SASP members. Inactive services are services that are suspended on the CSS. The service information includes:
  - The owner and content rule for the service
  - SASP group
  - Service name
  - SASP member name for the service
  - SASP and default weight
  - SASP flags
  - Time stamp when the reported weight occurred

The SASP agent statistics include the number of SASP messages received and transmitted by the CSS, and the number of message failures.

The syntax for this command is:

```
show sasp- agent label {active-list|inactive-list|statistics}
```

The variable and options are:

- *label* - The label configured for the SASP agent through the global configuration **sasp-agent** command.
- **active-list** - (Optional) Displays the active SASP member services for each content rule.
- **inactive-list** - (Optional) Displays the inactive SASP member services for each content rule.
- **statistics** - (Optional) Displays the statistics for SASP received and transmitted messages, and their failures. To clear the SASP statistics for the agent, use the **zero sasp-agent label statistics** command.

For example, to display the state of a SASP agent with the label agent1, and information about its active and inactive SASP groups and members, and statistics, enter.

```
# show sasp-agent agent1
```

To display the state of a SASP agent with the label agent1, and information about its active SASP groups and members, enter.

```
# show sasp-agent agent1 active-list
```

To display the state of a SASP agent with the label agent1, and information about its inactive SASP groups and members, enter.

```
# show sasp-agent agent1 inactive-list
```

To display the statistics for SASP received and transmitted messages, and their failures, enter.

```
# show sasp-agent agent1 statistics
```

[Table 7-3](#) describes the fields in the **show sasp-agent label** command output.



**Table 7-3** *Field Descriptions for the show sasp-agent label Command Output*

Field	Description
Agent State	<p>The state of the connection between the SASP agent and CSS. Possible states are:</p> <ul style="list-style-type: none"> <li>• Initialized - The SASP agent is created.</li> <li>• Closed - The SASP agent is disabled but not removed from the CSS.</li> <li>• Trying - The agent tries to open a socket to the SASP manager on the CSS.</li> <li>• Connecting - The SASP manager on the CSS opens the socket and is connecting to the SASP agent.</li> <li>• Established - An agent is communicating with the SASP manager.</li> <li>• DownTrying - A communication failure occurred between the SASP agent and CSS, and the CSS is attempting to reconnect to the agent.</li> <li>• DownDuplicate - The agent detects that a load balancer has a duplicate SASP ID is on the network.</li> <li>• Down - A communication failure occurred and the CSS is unable to reestablish the connection with the agent. The agent will be in this state for 60 seconds after a failure.</li> <li>• Other - This state occurs when no other state applies to the condition.</li> </ul>
Active/Inactive Member List	Lists the active or inactive member information in the fields that follow this field. If no member is part of active or inactive list, None appears next to this field.
Owner/Rule	The owner and content rule configured on the CSS.
SASP Group	The SASP group name for owner and content rule configured on the CSS.
Service Name	The name of the service associated with the SASP agent.
SASP Member	The member name for the service configured on the CSS.

**Table 7-3** *Field Descriptions for the show sasp-agent label Command Output (continued)*

Field	Description
Weight	<p>The weight after the CSS converts it from SASP agent, and the default weight of the service. The converted CSS weight is:</p> $\text{CSS weight} = (\text{SASP Weight} * \text{CSS Max Weight}) / \text{SASP Max Weight}$ <p>For example, when 3/5 is displayed, the 3 is the converted weight from the SASP agent and 5 is the default weight.</p> <p>If the agent reports a weight of zero and the converted weight is the value of one, the CSS weight is zero. If the agent reports a weight of 64 and the converted weight is the value of 11, the CSS weight is 10.</p> <p>An asterisk (*) indicates the weight being used.</p> <p>Note that the CSS may use the default weight if the register and confident flags are all true. This default weight selection may occur when either a rule weight is configured for the service, or one of the following problems occurred within the context of the rule weight update message:</p> <ul style="list-style-type: none"> <li>• If the SASP agent reports the quiesced flag for a member that is different from the quiesced flag stored on the CSS.</li> <li>• If the SASP agent reports the registration flag for a member is false indicating that the member registered itself.</li> </ul> <p>You can change the default weight for mapping between the SASP agent weight range and the CSS weight range with the <b>sasp max-agent-weight</b> command. See <a href="#">Changing the Maximum Accepted SASP Weight</a> for details.</p>

**Table 7-3** *Field Descriptions for the show sasp-agent label Command Output (continued)*

Field	Description
Flags RICIQ	<p>The registration (R), confident (C), and quiesced (Q) flags for the SASP member. A true flag appears as a T. A false flag appears as an F.</p> <p>The registration flag identifies whether the member has successfully registered with the SASP agent. The states are:</p> <ul style="list-style-type: none"> <li>• True identifies that the member successfully registered with the SASP agent.</li> <li>• False identifies that the member could not registered with the SASP agent. If this flag occurs, the registration failure counter increments.</li> </ul> <p>The confidence flag identifies whether the SASP agent is confident in the weight reported to the CSS for the member. The states are:</p> <ul style="list-style-type: none"> <li>• True indicates the SASP agent is confident of the weight reported.</li> <li>• False indicates that the SASP agent is not confident in the reported weight.</li> </ul> <p>The quiesced flag indicates whether the member is receiving load-balanced traffic from the CSS. The states are:</p> <ul style="list-style-type: none"> <li>• True indicates that the member is not receiving load-balanced traffic. The rule may be suspended or the keepalives detect that the service is down.</li> <li>• False indicates that the member is active and is receiving traffic from the CSS. The rule is active and the keepalives detect that the service is alive.</li> </ul>
Last-Wgt-Change	The last time the SASP agent reported the weight for the service member.

**Table 7-3** *Field Descriptions for the show sasp-agent label Command Output (continued)*

<b>Field</b>	<b>Description</b>
SASP Statistics	
Tx/Rx Registration Msgs	<p>The total number of SASP Registration messages that the CSS transmitted (Tx) or received (Rx). The transmitted Registration message informs the agent to start reporting the server weight for each member listed in the message.</p> <p>The received message indicates that the transmitted registration request succeeded or failed.</p>
Tx/Rx Deregistration Msgs	<p>The total number of SASP Deregistration messages that the CSS transmitted (Tx) or received (Rx). The transmitted deregistration message informs the agent to stop reporting the server weight for each member listed in the message.</p> <p>The received message indicates that the transmitted request succeeded or failed.</p>
Tx/Rx Get Weight Msgs	<p>The total number of SASP Get Weight messages that the CSS transmitted (Tx) or received (Rx). The transmitted Get Weights message is used only when the CSS is actively retrieving the weight for each member.</p> <p>This received message contains the current weights for the requested members.</p>
Tx/Rx Set LB State Msgs	<p>The total number of SASP Set Load Balancer State messages that the CSS transmitted (Tx) or received (Rx). The transmitted message informs the SASP agent of the CSS health and its current options. This message informs the SASP agent that a member has either been activated, or has dropped out of service.</p> <p>The received message indicates whether the transmitted request succeeded or failed.</p>

**Table 7-3** *Field Descriptions for the show sasp-agent label Command Output (continued)*

<b>Field</b>	<b>Description</b>
Tx/Rx Set Member State Msgs	The total number of SASP Set Member State messages that the CSS transmitted (Tx) or received (Rx). The transmitted message updates the member weights when the CSS is passively retrieving them.  The received message indicates whether the transmitted request succeeded or failed.
Rx Send Weight Msgs	The total number of SASP Send Weight Messages that the CSS received from the SASP Agent. The CSS uses this message to update the member weights when it passively retrieves them.
<b>Errors</b>	
Registration Failures	The total number of SASP Registration message failures.
Deregistration Failures	The total number of SASP Deregistration message failures.
Get Weight Failures	The total number of SASP Get Weight message failures.
Set LB State Failures	The total number of SASP Set Load Balancer State message failures.
Set Member State Failures	The total number of SASP Set Member State message failures.
Connection Failures	The total number of times the CSS encountered a connection failure with the SASP agent.

**Table 7-3** *Field Descriptions for the show sasp-agent label Command Output (continued)*

<b>Field</b>	<b>Description</b>
Send Weight Failures	The total number of times the CSS detects an errors in the send weights message. The connection closes unless the error is recoverable.
NoChange/Send Violations	A counter that increments when the CSS receives an updated weight for a member whose weight has not changed since the previous update. By default, the CSS sets the No Change/No Send flag within the set load balancer state request. This flag informs the SASP agent not to send a weight update for members whose weights have not changed.