
# Configuring Dynamic Feedback Protocol for Server Load Balancing

The Dynamic Feedback Protocol (DFP) is a mechanism that allows load-balanced servers (both local and remote) to dynamically report changes in their status and their ability to provide services to a CSS. A status report sent to a CSS from a server contains a relative weight/number of connections to define the load and availability of each server. A CSS incorporates server feedback into the load-balancing decision process in order to:

- Obtain server availability information
- Identify load imbalances over multiple sites
- Distribute traffic more evenly

This chapter contains the following major sections:



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

# DFP Overview

The DFP manager (running on the CSS as a task and part of the load manager) is responsible for establishing TCP connections with the DFP agents that reside on each server. A DFP manager can communicate simultaneously with a maximum of 127 DFP agents. DFP agents can be software running on the actual server itself or may be separate hardware devices that collect and consolidate information from one or more servers for load-balancing purposes. DFP agents are available from a number of third-party sources.

DFP agents collect relative weights from the load-balanced servers and periodically send new or adjusted weights to the DFP manager in the form of load vectors. The CSS load manager distributes the incoming connections or services (local or remote) to the servers in the order of weight assigned to the load-balanced servers. The load manager uses the reported weights to choose the best available server, resulting in optimal performance of servers and less response time.

**Note** If you configure a weight on a service using the **add service weight** command in owner-content configuration mode, the configured weight takes precedence over the service weight reported by the DFP agent for that content rule. In turn, the DFP-reported weight take precedence over the weight configured on a service in service configuration mode.

The CSS uses load-balancing algorithms such as roundrobin, weighted roundrobin, Arrowpoint Content Aware (ACA), and least connections to distribute the incoming connections or service requests. Weighted roundrobin can take advantage of the server weights reported by the DFP agents.

The weighted roundrobin load-balancing method uses weight to specify how many consecutive connections to give to the highest-weighted server before moving on to the next highest-weighted server. As a server's load changes, the DFP agent recalculates the weight for each server and reports the updated weights to the DFP manager, thereby influencing how the load manager distributes the service requests. For more information on CSS server load-balancing, refer to Chapter 9, Configuring Content Rules.

The following sections provides information on:



# Functions of a DFP Agent

A DFP agent reports server weight/connection information to the DFP manager. Multiple DFP agents can exist on a server platform. An agent provides several benefits to the load-balancing process. A DFP agent can inform the CSS that the server:

- Is congested
- Is under-utilized
- Should not be used for load balancing for a period of time

# Types of DFP Messages

The following messages are defined for communication between the DFP agent and the DFP manager in the CSS:

- The preference information message reports the status or weight of an IP server and is sent from the DFP agent to the DFP manager.
- The server state message, sent from the DFP manager to the agent, informs the agent that the load manager has decided to take the server in or out of service.
- The DFP parameters send configuration information from the DFP manager to the agent. Currently, the only configuration parameter passed is the keepalive interval.

DFP messages consist of a DFP header called a signal header followed by message vectors. Vectors are optional commands that exist in the defined messages. Each message vector contains a vector header, which is the first part of each vector in the DFP message, followed by data specific to the defined vector. The vector header allows the DFP manager or the DFP agent to discard any vectors or commands that it does not understand.

Defined vectors for DFP include:

- **Security Vector** - Allows each DFP message to be verified.
- **Load Vector** - Contains the actual load information being reported for the real servers and represents the servers' preferred capability.
- **Keepalive Vector -** Part of the DFP connection configuration. The keepalive vector allows the load manager to inform the DFP agent of the minimum time interval by which the agent must send information over the DFP connection to the CSS.

If a CSS receives a message that contains a vector type that it does not understand, The CSS discards the unknown vector.

# DFP System Flow

When you configure a DFP agent on a CSS, the DFP manager initiates a single TCP connection with the DFP agent (regardless of the number of servers the agent supports) with the parameters specified in the DFP agent configuration. The DFP manager sends a keepalive vector in a DFP message to change the default keepalive time if required.

After the connection is established, the DFP agent periodically sends update information in the form of a load-vector. If an agent has no information to send, it still must send an empty DFP packet to prevent the connection from being torn down.

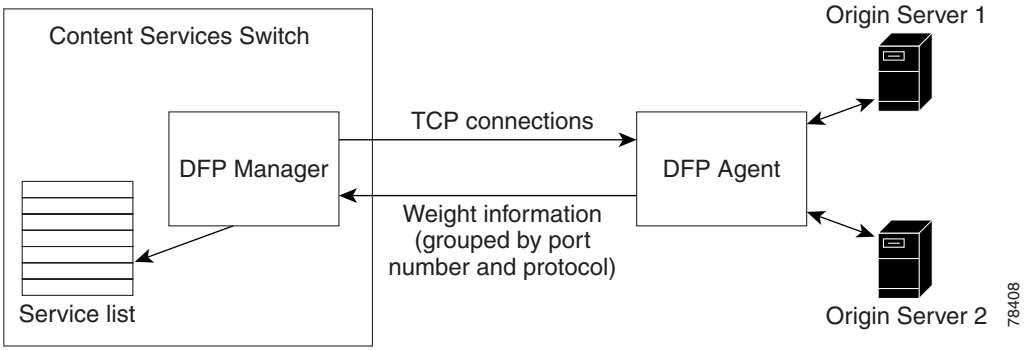
If a DFP agent is responsible for collecting information from multiple servers, the servers are grouped by their port number and protocol type, and a separate load vector is required for each grouping. A DFP agent can report weights for a maximum of 128 servers in a single weight report. Upon receiving information about an adjusted weight, the e DFP manager updates the weights of the server reported in the list of load-balanced servers.

If DFP is disabled, a CSS uses the weight configured on a service in owner-content configuration mode using the **add service weight** command (for that content rule only) or the weight configured on the service in service configuration mode, in that order. If no weight is configured on the service, the CSS uses a default weight of 1 to load balance the service. If a connection between a DFP agent and the DFP manager closes because of a timeout, a CSS uses the default weight for load balancing until the DFP manager reestablishes the connection with the DFP agent and obtains a new weight report.

If the configured DFP agent supports MD5 security, you can specify a shared key text string in the DFP manager. MD5 encryption is a one-way hash function that provides strong encryption protection. The CSS provides an MD5 secure connection between the DFP manager and the DFP agent on the server. In this secure environment, the CSS discards DFP messages from the server unless the messages contain the MD5 code.

Figure 7-1 summarizes the relationship between the DFP manager (in the CSS) and a DFP agent.

***Figure 7-1 Example of DFP Manager to DFP Agents System Flow***




# Configuring a DFP Agent

To configure a DFP agent listening for DFP connections on a particular IP address and TCP port combination on a server and to enable the DFP manager on the CSS, use the **dfp** command. You can configure a maximum of 127 DFP agents for the DFP manager in the CSS. Use the **no dfp** command to disable the DFP agent connection to a particular IP address.

The syntax for the **dfp** command is:

**dfp** *ip_or_host* {*port*} {**key "***secret***"**|[**des-encrypted** *encrypted_key*|**"***encrypt_key***"**]} {**timeout** *seconds*} {**retry** *count*} {**delay** *time*} {**max-agent-wt** *weight*}

The variables and options are:

- *ip_or_host* - The IP address or host name of the configured DFP agent. Enter an IP address in dotted-decimal notation (for example, 192.168.11.1) or a mnemonic host name (for example, myhost.mydomain.com).
- *port* - (Optional) The server TCP port that the configured DFP agent uses to listen for connections from the CSS DFP manager. Valid entries are 0 to 65535. The default is 14001.

**Note** Do not configure a service TCP keepalive to connect to the same port that the DFP agent uses to listen for connections from the DFP manager. This type of configuration causes the built-in DFP keepalive to fail.

- **key "***secret***"** - (Optional) An MD5 security key used for encryption to provide a secure data exchange between the CSS DFP manager and the DFP agents. MD5 encryption is a one-way hash function that provides strong encryption protection. Enter the secret as a case-sensitive quoted text string (maximum of 64 characters). It can include any printable ASCII character except tabs.

  For DFP to function properly, ensure that you configure the same key on each DFP agent that you configured on the DFP manager. If the key on an agent does not match the key on the DFP manager, no connection will be established and the DFP agent will not be able to send a weight report to the CSS. In this case, when the DFP manager fails to establish a connection with an agent for a given key, the CSS logs the following informational message in SYSLOG: `Secret key might not be same as DFP agent's key. Check secret key.`
- **des-encrypted** - (Optional) Specifies that a Data Encryption Standard (DES) key follows.
- *encrypted_key* - The DES key that the CSS previously encrypted. The CSS does not reencrypt this key. The CSS saves the key in the running-config the same as you entered it. Enter an unquoted case-sensitive text string with no spaces and a maximum of 128 characters.
- **"***encrypt_key***"** - The DES encryption key that you want the CSS to encrypt. The CSS saves the encrypted key in the running-config as you entered it. Enter a quoted case-sensitive text string with no spaces and a maximum of 64 characters.

- **timeout** *seconds* - (Optional) The maximum inactivity time period (the keepalive time) for the connection between the CSS DFP manager and the server DFP agent. If the inactivity time period exceeds the timeout value, the DFP manager closes the connection. The DFP manager attempts to reopen the connection as often as specified by the value of the **retry** option. The range is from 1 to 10000 seconds. The default is 3600 seconds (1 hour).
- **retry** *count* **-** (Optional) The number of times the CSS DFP manager tries to reopen a connection with the server DFP agent. The range is from 0 (for continuous retries) to 65535. The default is 3 retry attempts.
- **delay** *time* **-** (Optional) The delay time, in seconds, between each attempt to reestablish a connection. Valid entries are 1 (immediately) to 65535 seconds (18 hours). The default value is 5 seconds.
- **max-agent-wt** *value* - (Optional) Maximum value of the weight reported by a DFP agent. A CSS uses this option to scale the reported weight when the weight range of a DFP agent does not match the weight range of the DFP manager. For example, the DFP manager weight range is 0 to 255. If a DFP agent reports weight in the range 0 to 16, the CSS scales up the agent-reported weight to match the weight range of the DFP manager. If an agent reports weight in the range 0 to 65535, the CSS scales down the agent-reported weight to match the weight range of the DFP manager.

  If a DFP agent reports a weight greater than the maximum configured weight, then the CSS rejects the weight report and does not use the weight in load-balancing decisions. In this case, the CSS also logs an error in SYSLOG. Enter an integer from 1 to 65535. The default is 255.

For example, the following command configures the DFP manager to communicate with the DFP agent at the specified address running with the following options and variables:

- DFP agent IP address - 192.168.1.2
- Port - 14001 (default)
- MD5 security key - “hello”
- Connection timeout - 6000 seconds
- Number of connection retries - 3
- Delay between connection retries - 60 seconds

```
(config)# dfp 192.168.1.2 14001 key “hello” timeout 6000 retry 3
delay 60
```

To disable the DFP agent, enter:

```
(config)# no dfp 192.168.1.2
```

# Maintaining a Consistent Weight Range Among Services

The CSS has a weight range of 1 through 10; the DFP manager has a weight range of 0 through 255. Because of this difference in weight ranges, you may need to manually adjust the weights configured on the DFP agent for different services to maintain the same service weight range that exists outside of the DFP.

For example, suppose that you configure on the same content rule three services (serv1, serv2, and serv3) with weights of 1, 2, and 5, respectively. If the DFP agent reports a weight of 20 for serv1, serv1 will now receive 20 connections for every 2 connections on serv2 and 5 connections on serv3. This configuration places a disproportionate load on serv1, especially if serv2 and serv3 represent fast servers with plenty of unused resources.

To solve this problem and to maintain the same weight range for all three services, you can do either of the following:

- Force the DFP agent to report a weight in the range of 1 to 10 for serv1
- Have the DFP agent report weights for all three services to maintain the same weight range

# Displaying Configured DFP Agents

For reporting purposes, you can view the configured DFP agents on a CSS using the **show dfp** command. This command displays a list of all DFP agents or the DFP agents at the specified IP address or host name arranged by their IP addresses, the port number on which the agent is connected to the DFP manager, the current state of the DFP agent, the keepalive time for the DFP TCP connection, and the DES-encrypted key of the agent, if any.

The syntax for this command is:

**show dfp** *ip_or_host*

The *ip_or_host* variable allows you to specify the DFP agent or agents running at a particular IP address or host name.

For example, to display configuration information for all DFP agents, enter:

```
# show dfp
```

Table 7-1 describes the fields in the **show dfp** command output.

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

| Field | Description |
|---|---|
| IP Address | The IP address of the configured DFP agent. |
| Port | The port number of the configured DFP agent. The default is 14001. |
| State | The state of the DFP agent. Possible states are Active, Dead, or Connecting. |
| KAL | The configured maximum inactivity time, in seconds, for the TCP connection between the DFP manager and the DFP agent. When this time elapses, the CSS tears down the connection. |
| MD5 Key | The DES-encrypted key of the DFP agent, if configured. |

# Displaying Services Supported by Configured DFP Agents

To view the individual weights of load-balanced services reported by a configured DFP agent, use the **show dfp-reports** command. This command groups the weights by the port number of reported services, the type of protocol, and the IP address of servers.

The syntax for this command is:

**show dfp-reports** {*ip_or_host* {**port** *number* {**protocol** *text* {**ip** *ip_or_host*}}}}

The options and variables for this command are:

- *ip_or_host* - The IP address or host name of the configured DFP agent. Enter an IP address in dotted-decimal notation (for example, 192.168.11.1) or a mnemonic host name (for example, myhost.mydomain.com).
- **port** *number* - (Optional) The port number for the load-balanced server or service. Valid entries are 0 to 65535. The default is 14001.
- **protocol** *text* - (Optional) The type of protocol for the load-balanced server or service. Possible values are TCP, UDP, HTTP, or FTP.
- **ip** *ip_or_host* - (Optional) The IP address or host name of the load-balanced server or service. Enter an IP address in dotted-decimal notation (for example, 192.168.11.1) or a mnemonic host name (for example, myhost.mydomain.com).

The following example shows the weight reported by a DFP agent configured at 192.168.1.2, for server 192.168.1.3. Weights are first grouped by port number of reported servers, and then by protocol.

```
# show dfp-reports 192.168.1.2 port 80 protocol tcp ip 192.168.1.3
```

Table 7-2 describes the fields in the **show dfp-reports** command output.

***Table 7-2 Field Descriptions for the show dfp-reports Command Output***

| Field | Description |
|---|---|
| Service | The name of the configured service for which the DFP agent is reporting |
| Weight | The last weight reported by the DFP agent for the service |
| Time-Stamp | The month, day, and time of the last-received report |
| # of Reports | The total number of reports |

# Displaying DFP Information

To display DFP information, see the following sections:

- Using the show service Command
- Using the show rule services Command

## Using the show service Command

Use the **show service** command to display service-specific information. The **show service** command output includes a DFP field that indicates the state of DFP. Possible states are Enable or Disable.

The state is Enable when DFP is configured and there is no weight configured on the service in owner-content configuration mode. The state is Disable if DFP is not enabled or if DFP is enabled and you have configured a service weight in owner-content configuration mode using the **add service weight** command.

For details on the **show service** command, see the "Showing Service Configurations" section in Chapter 3, Configuring Services.

# Using the show rule services Command

Use the **show rule services** command in owner-content mode to display weights configured for services in service mode, owner-content mode, and DFP, as well as other service-related information. The output of the command includes the weight assigned to each service preceded by a code letter. The code letters have the following meanings:

- D, the weight reported by a DFP agent
- R, the weight configured for a service using the **add service weight** command in owner-content mode
- S, the weight configured for a service using the **weight** command in service mode

For details on the **show rule services** command, see Chapter 9, Configuring Content Rules.