Configuring a Primary Proxy Server

This chapter explains how to assign a Content Engine as a primary proxy server in a Content Engine cluster, and how to configure backup Content Engines. This helps the network identify which Content Engine is the primary proxy server and which ones can serve content in case the primary proxy server fails. This chapter contains the following sections:

Configuring Primary Proxy Failover

Note The primary proxy failover feature supports HTTP only, not HTTPS or FTP.

The **http proxy outgoing** option can configure up to eight backup Content Engines or any standard proxy servers for the HTTP proxy failover feature. One outgoing proxy server functions as the primary server to receive and process all cache-miss traffic. If the primary outgoing proxy server fails to respond to the HTTP request, the server is noted as failed and the requests are redirected to the next outgoing proxy server until one of the proxies services the request. The **no http proxy outgoing connection-timeout** option causes the timeout to be set to the default value of 300 milliseconds.

To explicitly designate a proxy as primary, use the **http proxy outgoing host** *ip-address port* **primary** command. If several hosts are configured with the **primary** keyword, the last one configured becomes the primary failover host. Failover occurs in the order that the proxy servers were configured. If all of the configured proxy servers fail, the Content Engine can optionally redirect HTTP requests to the origin server specified in the HTTP header with the **http proxy outgoing origin-server** command. If the **origin-server** option is not enabled, the client receives an error message. Response errors and read errors are returned to the client, because it is not possible to detect whether these errors are generated at the origin server or at the proxy.

A background process monitors the state of the proxy servers. A monitoring interval is configured with the **http proxy outgoing monitor** command. This monitor interval is the interval of time over which the proxy servers are polled. If one of the proxy servers is unavailable, the polling mechanism waits for the connect timeout (300 milliseconds) before polling the next server. The state of the proxy servers can be viewed in syslog NOTICE messages and with the **show http proxy** command.

By default, the Content Engine strips the hop-to-hop 407 (Proxy Authentication Required) error code sent by Internet proxy. If the **http proxy outgoing preserve-407** command is invoked, the Content Engine sends the 407 error code to the client, and the Internet proxy authenticates the client.

Note Only one of the outgoing proxy servers is available at a time. They cannot be used simultaneously.

Requests with a destination specified in the **proxy-protocols outgoing-proxy exclude** command bypass the primary outgoing proxy and the failover proxies.

WMT over HTTP

You can also designate an outgoing HTTP proxy server for streaming media in MMS format. Use the **wmt proxy outgoing http host** global configuration command to configure the outgoing proxy for this format. This allows the forwarding of MMS data over HTTP to a standard 8080 proxy port.

Note The MMS protocol can run on top of three different data protocols: MMS over TCP, MMS over UDP, and MMS over HTTP. ACNS 4.2 software supports only MMS over HTTP at this time.

HTTP Transparent and Proxy Caching

Certain scenarios involve the deployment of a Content Engine in proxy mode at company headquarters and Content Engines in transparent mode at remote locations in branch offices. In this scenario if a cache miss occurs at the remote Content Engine, company policy requires that the request be routed to the Content Engine at headquarters.

When an HTTP request intended for another proxy server is intercepted by the Content Engine in transparent mode, the Content Engine forwards the request to the intended proxy server if the **proxy-protocols transparent original-proxy** command was entered. If this command was not entered, then the Content Engine forwards the request to the origin server where the initial HTTP request was made. See the "Handling Proxy-Style Requests" section on page 8-4 for more information on this feature.

Examples

In this example, the host 10.1.1.1 on port 8088 is designated the primary outgoing proxy server, and host 10.1.1.2 is a backup proxy server.

```
ContentEngine(config)# http proxy outgoing host 10.1.1.1 8088 primary
ContentEngine(config)# http proxy outgoing host 10.1.1.2 220
```

In this example, the Content Engine is configured to redirect requests directly to the origin server if all of the proxy servers fail.

```
ContentEngine(config)# http proxy outgoing origin-server
```

In this example, the Content Engine is configured to monitor the proxy servers every 120 seconds.

```
ContentEngine(config)# http proxy outgoing monitor 120
```

To disable any of the above commands, use the **no** version of the command.

Proxy Failover show Commands

```
ContentEngine# show http proxy
Incoming Proxy-Mode:
  Servicing Proxy mode HTTP connections on ports:   8080

Outgoing Proxy-Mode:
  Primary proxy server: 172.16.63.150   port 1 Failed
  Backup proxy servers: 172.16.236.151  port 8005
                        172.16.236.152  port 123
                        172.16.236.153  port 65535 Failed
                        172.16.236.154  port 10
Monitor Interval for Outgoing Proxy Servers is 60 seconds
Timeout period for probing Outgoing Proxy Servers is 300000 microseconds
Use of Origin Server upon Proxy Failures is disabled.
```

Statistics

```
ContentEngine# show statistics http requests
Statistics - Requests
                                              Total             % of Requests
                              -------------------------------------------------
     Total Received Requests:                 49103                        -
              Forced Reloads:                   109                      0.2
               Client Errors:                    23                      0.0
               Server Errors:                   348                      0.7
                 URL Blocked:                     0                      0.0
      Sent to Outgoing Proxy:                     0                      0.0
Failures from Outgoing Proxy:                     0                      0.0
Excluded from Outgoing Proxy:                     0                      0.0
             ICP Client Hits:                     0                      0.0
             ICP Server Hits:                     0                      0.0
           HTTP 0.9 Requests:                     2                      0.0
           HTTP 1.0 Requests:                 49101                    100.0
           HTTP 1.1 Requests:                     0                      0.0
       HTTP Unknown Requests:                     0                      0.0
           Non HTTP Requests:                     0                      0.0
          Non HTTP Responses:                    46                      0.1
      Chunked HTTP Responses:                     0                      0.0
        Http Miss Due To DNS:                     0                      0.0
     Http Deletes Due To DNS:                     0                      0.0
  Objects cached for min ttl:                  2674                      5.

ContentEngine# show statistics http proxy outgoing

HTTP Outgoing Proxy Statistics
IP              PORT    ATTEMPTS   FAILURES
-------------------------------------------------
172.16.23.150    8000    0          0
172.16.23.151    8080    0          0
172.16.23.152    9000    0          0
172.16.23.153    9001    0          0
172.16.23.154    9005    0          0


 Requests when all proxies were failed: 0
```

Related Commands

proxy-protocols

rule no-proxy

rule use-proxy

show http

show http proxy

show statistics http requests

show statistics http proxy outgoing

Handling Proxy-Style Requests

When in transparent caching mode, the Content Engine can intercept requests sent to another proxy and send these requests to one of the following two destinations:

- Default server—This is the default option. The Content Engine retrieves the objects from the web server itself, or, if configured to use an outgoing proxy for this protocol, it forwards the request to its outgoing proxy. In this scenario, the client browser configuration is ignored and the Content Engine configuration is used to retrieve the object from the server.
- Original proxy—The Content Engine forwards the request to the original proxy that the client addressed the request to. This may be different from the Content Engine's own outgoing proxy for the specified protocol.

Use the **proxy-protocols** global configuration command to specify a domain name, host name, or IP address to be excluded from proxy forwarding. To selectively turn off outgoing-proxy exclude lists or to force transparently received proxy-style requests to be fulfilled by the Content Engine, use the **no** form of this command.

proxy-protocols outgoing-proxy exclude {**enable** | **list** *word*}

proxy-protocols transparent {**default-server** | **original-proxy | reset**}

no proxy-protocols {**outgoing-proxy exclude** {**enable** | **list** *word*} | **transparent** {**default-server** | **original-proxy**}}

The **proxy-protocols outgoing-proxy exclude** option allows the administrator to specify a single domain name, host name, or IP address to be globally excluded from proxy forwarding. Domains are entered as an ASCII string, separated by spaces. The wildcard character * (asterisk) can be used for IP addresses (for instance, 172.16.*.*). Only one exclusion can be entered per command line. Enter successive command lines to specify multiple exclusions. Requests with a destination specified in the **proxy-protocols outgoing-proxy exclude** command bypass the Content Engine proxy as well as the failover proxies.

When you enter the **proxy-protocols transparent default-server** global configuration command, the Content Engine forwards intercepted HTTP, HTTPS, and FTP proxy-style requests to the corresponding outgoing proxy server, if one is configured. If no outgoing proxy server is configured for the protocol, the request is serviced by the Content Engine and the origin server.

The **proxy-protocols transparent original-proxy** option specifies that requests sent by a web client to another proxy server, but intercepted by the Content Engine in transparent mode, be forwarded to the intended proxy server.

The **proxy-protocols transparent reset** option specifies that requests sent by a web client to another proxy server, but intercepted by the Content Engine in transparent mode, be returned to the web client during a cache miss and the requested objects are not delivered.

The following example configures the Content Engine to forward intercepted HTTPS proxy-style requests to an outgoing proxy server. The domain name cruzio.com is excluded from proxy forwarding. The **show proxy-protocols** command verifies the configuration.

```
ContentEngine(config)# https proxy outgoing host 172.16.10.10 266
ContentEngine(config)# proxy-protocols transparent default-server
ContentEngine(config)# proxy-protocols outgoing-proxy exclude cruzio.com

ContentEngine# show proxy-protocols all
Transparent mode forwarding policies: default-server
Outgoing exclude domain name: cruzio.com
```

The following example configures the Content Engine to forward intercepted HTTP proxy-style requests to the intended proxy server.

```
ContentEngine(config)# proxy-protocols transparent original-proxy
```

Internet Cache Protocol

Internet Cache Protocol (ICP) is a lightweight message format used for communicating among Content Engines and for supporting interoperability with older proxy protocols. ICP is used to exchange hints about the existence of URLs in neighboring caches in a Content Engine farm. Content Engines exchange ICP queries and replies to gather information for use in selecting the most appropriate location from which to retrieve an object.

Although ICP has traditionally been a way to scale the overall size of a cluster of caches beyond a single unit, history has shown ICP to be a poor way of scaling a cache clustering solution. In fact, because of the way that traffic is currently directed toward a transparent network cache cluster, the requirement for ICP is all but negated for the majority of cache deployments.

The ICPv2 protocol is documented in two standards documents:

- *RFC 2186: Internet Cache Protocol (ICP), version 2*
- *RFC 2187: Application of Internet Cache Protocol (ICP), version 2*

Note The ability to act as both an ICP server (servicing requests from neighboring caches) and an ICP client (sending requests to neighboring caches) is supported.

The following example restricts ICP parent and sibling to specific domain sets:

```
ContentEngine(config)# icp client add-remote-server 10.1.1.1 parent icp-port 3130
http-port 3128 domain_x.com domain_y.com domain_z.com
ContentEngine(config)# icp client add-remote-server 10.1.1.1 sibling icp-port 3130
http-port 3128 domain_a.com domain_b.com domain_c.com

ContentEngine(config)# icp client enable
ContentEngine(config)#
```