Troubleshooting

This appendix provides information on troubleshooting. The following topics are covered in this appendix:

For more troubleshooting tools, see Chapter 7, "Monitoring the Internet Streamer VDS."

Troubleshooting Service Router Configurations

Because there are many steps required for the Service Router to redirect the request properly, you might see some content request errors from the Service Router when the configuration is not quite complete. Here are some areas to look at when troubleshooting:

- DNS delegation
  - Is the requested domain delegated to the Service Router on the DNS server that is authoritative for the parent domains? The Service Router's DNS name should be forward resolvable. Check with the system administrator to delegate a domain.
- Service Router routing properties
  - Is the Service Router activated? See the "Activating a Service Router" section on page 4-103 to activate a Service Router.
  - Is a default coverage zone set for a Service Engine, or is there a VDS network-wide Coverage Zone file or a local Coverage Zone file set for the Service Router? See the "Coverage Zone File Registration," page 6-12 to set a Coverage Zone file. See the "Configuring the Service Engine" section on page 4-9 to set a default coverage zone.
  - Is the content request from an end system covered by a Service Engine in a coverage zone based on the default coverage zone or the Coverage Zone file? This Service Engine is the "serving Service Engine." See the "Coverage Zone File" section on page 1-39 for information on coverage zones. See Appendix C, "Creating Coverage Zone Files," for information on creating a Coverage Zone file.
  - Is the serving Service Engine activated? See the "Activating a Service Engine" section on page 4-10 to activate a Service Engine.

  - Is there a delivery service created for the requested domain and a serving Service Engine assigned to this delivery service? See the "Creating Delivery Service" section on page 5-16.
  - Is the serving Service Engine alive? Use the **show statistics service-routing se** command to show the status of a Service Engine. See the "Using show and clear Commands" section on page 7-16.
- Content prefetched on a Service Engine
  - Is a Manifest file assigned to the delivery service associated with the serving Service Engine? See the "Working with Manifest Files" section on page B-2.
  - Is the Manifest file accessible from the CDSM? See the "Identifying Content Using a Manifest File" section on page 5-43.
  - Is there any syntax error in the Manifest file? See the "Manifest File Structure and Syntax" section on page B-19.
  - Is the requested content specified in the Manifest file? See the "Specifying a Single Content Item" section on page B-2.
  - If the requested content is streaming media, is the protocol engine enabled? See the "Application Control" section on page 4-35.

For general information, use the **show statistics service-router all** command.

Troubleshooting the Distribution Hierarchy

Because distribution-related problems are design-dependent, your initial strategy is to discover whether or not the correct Service Engine is sending content in the correct distribution path.

- To determine which Service Engines are in the distribution path of a particular Service Engine, use the **show distribution remote traceroute** EXEC command, as shown in the following example:

```
ce1# show distribution remote traceroute ?
  forwarder-next-hop  next forwarder along the path
  unicast-sender      check status for unicast sender

ce1# show distribution remote traceroute forwarder-next-hop ?
  delivery-service-id  Delivery-service-id of a Delivery Service

ce1# show distribution remote traceroute forwarder-next-hop delivery-service-id 133 ?
  max-hop          Trace route till specified number of hops is reached
  trace-till-good  traceroute till probe is good or the object is found
  trace-till-root  traceroute till the acquirer

ce1# show distribution remote traceroute forwarder-next-hop delivery-service-id 133
trace-till-root

Hop NextHop_SEId  NextHop_SEName NextHop_SEIp      GenID Status/Reason
--- ------------  -------------- ------------      ----- -------------
  1    1100               ce3     10.255.0.43     1       LOC-LEAD
  1    1100               ce3 128.107.193.183     1       LOC-LEAD (Reached RootCE)
```

- To verify that the Service Engine is reachable and that it is in the distribution hierarchy, use the **show distribution remote traceroute** EXEC command, as shown in the following example:

```
se1# show distribution remote traceroute unicast-sender delivery-service-id 133 ?
  cdn-url           check the object on remote SE using cdn-url
  probe             probe the remote unicast sender
  relative-cdn-url  check the object on remote SE using relative-cdn-url
```

```
se1# show distribution remote traceroute unicast-sender delivery-service-id 133 probe
?
  max-hop          Max-hop to traceroute to
  trace-till-good  traceroute till probe is good or the object is found
  trace-till-root  traceroute till the root se

se1# show distribution remote traceroute unicast-sender delivery-service-id 133 probe
trace-till-root
Polling .... se3 [10.255.0.43]  Fwdr_Id:1100
Polling .... se3 [128.107.193.183]  Fwdr_Id:1100
(Reached RootSE)
```

Troubleshooting Content Acquisition

To monitor acquisition progress and to troubleshoot, use the following commands from the Content Acquirer CLI:

- Use the **show acquirer delivery-services** EXEC command to obtain delivery service information, such as the delivery service ID and delivery service name, that you need to enter in other **show acquirer** commands, such as the **show acquirer progress** command. In the following example, the delivery service ID is 793 and the delivery service name is group01-cifs.

```
SE# show acquirer delivery-services
Querying Database.......
Acquirer information for all delivery services:
---------------------------------------
Delivery-service-id       : 793
Delivery-service-Name     : group01-cifs
WebSite-Name     : group01-cifs
Root-CE-Type     : Configured
State            : Enabled
Disk Quota       : 200 MB
Origin FQDN      : cdn.allcisco.com
Delivery-service Priority : 500
Manifestfile-TTL : 5
Manifestfile-URL : ftp://10.1.1.1/cifs.xml
```

- Use the **show acquirer** EXEC command to make sure that the acquirer process on the Content Acquirer is working correctly, and that the device is using the expected amount of bandwidth for acquisition. The following example shows that the acquirer is running properly and that the device is configured with unlimited bandwidth for acquisition of content.

```
SE# show acquirer
Acquirer is running OK
Current Acquisition Bandwidth:Not Limited
```

- Use the **show acquirer progress** EXEC command to check how far the acquisition of content has progressed. A specific delivery service ID or delivery service name can be specified to obtain the progress for a specific delivery service. In the following example, the acquirer has already acquired 2237 items.

```
SE# show acquirer progress delivery-service-id 793
Querying Database.......

Acquirer progress information for delivery service ID:793
Delivery-service-Name:group01-cifs
-----------------------------------------------------------------

Acquired Single Items   :         0 / 0
Acquired Crawl Items    :     2237 / 2500     -- start-url=www.mtv.com//
```

- Use the **show statistics acquirer delivery-service-id** or **show statistics acquirer delivery-service-name** EXEC command to obtain the detailed acquisition statistics for a given delivery service. In the following example, there was an error acquiring two items.

```
SE# show statistics acquirer delivery-service-id 793
Querying Database.......

Statistics for Delivery Service Delivery-service-id :793 Delivery-service-Name
:group01-cifs
-------------------------------------------------------

Manifest:
---------
        Fetch Errors    :0
        Parsing Errors  :0
        Parsing Warnings:0

Acquisition:
------------
        Total Number of Acquired Objects        :2237
        Total Disk Used for Acquired Objects    :981511280 Bytes
        Total Number of Failed Objects          :2
        Total Number of Re-Check Failed Objects :0
```

- Use the **show statistics acquirer errors delivery-service-id** or **show statistics acquirer errors delivery-service-name** EXEC command to see the reasons why the errors occurred. In the following example, one error occurred because there was a problem acquiring the URL. The other error occurred because the disk quota for the delivery service configured in the Content Distribution Manager GUI would have been exceeded if the specified URL had been acquired. You can increase the delivery service disk quota to correct this error.

```
SE# show statistics acquirer errors delivery-service-id 793
Querying Database.......

Acquisition Errors for the Delivery Service ID:793
-------------------------------------
Crawl job:start-url http://www.mtv.com//
Crawl Errors
---------------
Internal Server Error(500):http://cgi.cnn.com/entries/intl-emailsubs-confirm
Exceeded Disk Quota(703):http://www.cdt.org/copyright/backgroundchart.pdf
```

- If more detailed troubleshooting of content acquisition is required, you can increase the debug level of the acquirer using the **debug acquirer trace** EXEC command. The logs are written to local1/errorlog/acquirer-errorlog.current.
- To verify that an expected object has been pre-positioned on the Service Engine, use the **show distribution object-status** EXEC command, as shown in the following example:

```
SE# show distribution object-status
http://172.18.81.168/Videos/SM-final%20Innebandy%202003.wmv

        ========== Website Information ==========
Name:           RTPServer5
Origin Server FQDN:       172.18.81.168
Service Routing FQDN:     N/A
Content UNS Reference #:  1
        ========== Delivery Services Information ==========
        *** Delivery Service 1903 (name = A_Multicast) ***

Object Replication
------------------
```

```
Replication:              Done
File State:               Ready for distribution
Multicast for Delivery Service: Not Enabled
Replication Lock:         Received by Unicast-Receiver/Acquirer
Reference Count:          1
Total Size:               2756437
Transferred Size:         2756437
MD5 of MD5:               tjS#DxqE5oUc024Z8XtFDw..
Source Url:               http://172.18.81.168/Videos/SM-final%20Innebandy%202003.wmv
Source Last Modified Time:Wed Jan  7 19:03:48 2004

Object Properties
-----------------
Redirect To Origin:       Yes
Requires Authentication:  No
Alternative URL:
Serve Start Time:         N/A
Serve End Time:           N/A
Play servers:             HTTP HTTPS WMT
Content Metadata:         None
Content uns_id:           NgcJTCU#JaY4ZGIPbsrONw..
Content gen-id:           1768:1136512329:2

========== CDNFS Information ==========
Internal File Name:
/disk00-04/d/http-172.18.81.168-k5bsm1o+y14jgiqsvwaohq/19/19f6d5cec7266c33f419709dc28c
8d9b.0.data.wmv
Actual File Size:         2756437 bytes
MD5 of MD5 (Re-calculated):tjS#DxqE5oUc024Z8XtFDw..
Content metadata:         None
Metadata match with:      Delivery Service 1903
Number of Source-urls:    1

Source-url to CDN-object mapping:
Source-url:               http://172.18.81.168/Videos/SM-final%20Innebandy%202003.wmv
Used by CDN object:       ---- Yes ----
Internal File Name:
/disk00-04/d/http-172.18.81.168-k5bsm1o+y14jgiqsvwaohq/19/19f6d5cec7266c33f419709dc28c
8d9b.0.data.wmv
Actual File Size:         2756437 bytes

 ========== CDNFS lookup output ==========
CDNFS File Attributes:
  Status                  3  (Ready)
  File Size               2756437 Bytes
  Start Time              null
  End Time                null
  Allowed Playback via    HTTP WMT HTTPS
  Last-modified Time      Wed Jan  7 19:03:48 2004
  cache-control           max-age=864000
  cdn_uns_id              NgcJTCU#JaY4ZGIPbsrONw..
  content-type            video/x-ms-wmv
  etag                    "042e6fa50d5c31:b39"
  file_duration           65
  last-modified           Wed, 07 Jan 2004 19:03:48 GMT
  server                  Microsoft-IIS/6.0
  x-powered-by            ASP.NET
Internal path to data file:
/disk00-04/d/http-172.18.81.168-k5bsm1o+y14jgiqsvwaohq/19/19f6d5cec7266c33f419709dc28c
8d9b.0.data.wmv
```

By comparing fields, such as Total Size, Transferred Size, and Source URL in the Object Replication output and Actual File Size and Source URL in the Source-URL to CDN-Object Mapping output, you can determine whether or not the object that is stored is the same as the object that was requested.

- To view the file directory structure on the Service Engine and verify the physical file on the disk, you can use the **cdnfs browse** EXEC command, as shown in the following example:

```
SE# cdnfs browse

------  CDNFS interactive browsing  ------
dir, ls:   list directory contents
cd,chdir:  change current working directory
info:      display attributes of a file
more:      page through a file
cat:       display a file
exit,quit: quit CDNFS browse shell

/>ls
                    srv2-static.cisco.com/
                    http-172.18.81.151-glfc4h-b9gywnf5rlnfweg/
                    http-172.18.81.163-og5o21u178nrhw1mctgtiq/
                    172.18.81.163/
                    file--xrnfwxifgu62jtiwtyixvg/
                    172.18.81.168/
                    http-172.18.81.168-k5bsm1o+y14jgiqsvwaohq/
                    172.18.81.155/
/>cd 172.18.81.168/
/172.18.81.168/>ls
376 Bytes           manifest-Delivery_service_1903.xml-lEYmrfnjt2o5GbUNwLCApA
                    Videos/
/172.18.81.168/>cd Videos
/172.18.81.168/Videos/>ls
2756437 Bytes       SM-final Innebandy 2003.wmv <===============Physical file on disk
/172.18.81.168/Videos/>quit
SE#
```

Enabling the Kernel Debugger

Cisco VDS software allows you to enable or disable access to the kernel debugger from the CDSM. Once enabled, the kernel debugger is automatically activated when kernel problems occur.

Note The "hardware watchdog" is enabled by default and automatically reboots a device that has stopped responding for over ten minutes. Enabling the kernel debugger disables the "hardware watchdog."

If the device runs out of memory and kernel debugger (KDB) is enabled, the KDB is activated and dump information. If the KDB is disabled and the device runs out of memory, the syslog reports only dump information and reboots the device.

To enable the kernel debugger, do the following:

Step 1 Choose **Devices > Devices > General Settings > Troubleshooting > Kernel Debugger**. The Kernel Debugger page is displayed.

Step 2 To enable the kernel debugger, check the **Enable** check box, and click **Submit**.

Troubleshooting Web Engine Cache Status Codes

Generally speaking, a TCP_MISS/504 status code can occur in a multi-tier system at two places:

- Between the downstream SE and the upstream SE
- Between the Content Acquirer and the Origin Server

Normally, the verification of a 504 status code is a bottom up procedure; that is, start from the edge SE and end at the Origin Server. Following is an example of the bottom-up checking procedure for a 504 status code in a three-tiered system:

1. If a 504 status code is observed on the edge SE, then the transaction logs on the middle SE should be checked to see what the status code is for that request.
2. If the status code on the middle SE is also 504, then check the transaction logs on the Content Acquirer to see what the status code of that request is:
   a. If the status code on the Content Acquirer is also 504, it means the Origin Server is taking too much time to respond and the Content Acquirer timeout period expires. Check the Origin Server to see why it is not responding in a timely manner.
   b. If the status code on the Content Acquirer is not 504 (for example, it is 200), it means the Origin Server is responding to the Content Acquirer quickly enough; however, the Content Acquirer somehow takes too much time to respond to the middle SE. If this is the case, investigate why the Content Acquirer is busy.
3. If the status code on the middle SE is not 504, similar to the Content Acquirer case mentioned above, it means the Content Acquirer is responding to the middle SE quickly enough (the transaction logs on the Content Acquirer should confirm this); however, the middle SE somehow take too much time to respond the edge SE. If this is the case, investigate why the middle SE is busy.

Basically, a 504 status code is normally caused by the upstream device taking too much time to respond to the downstream device. This troubleshooting section only covers the investigation of the upstream SEs. A 504 status code can occur on the upstream SE for the following reasons:

- Network-related issues between the downstream device and upstream device
- Issues inside the upstream SEs
- Configuration related issue

Table A-1 provides more information about the scenarios where the 504 status code was observed, the tools used to investigate the scenarios, and the investigation results of each one.

Table A-1 Scenarios of 504 Status Codes

Scenario	Investigation Results
Network Related Issue Between Downstream Device and Upstream Device	
Tools used to investigate were the following: tcpdump, netstat, netmon, iostat, mpstat, and ping.	
• Three-tier system • 504 seen on the middle SE • Content Acquirer had 200 hits for the same entry	• Network interface cards throughput was under limit • Disks were not overloaded • Traffic had no problems for a long time, but at certain moments saw all dropped to 504s • Output from the **tcpdump** command shows response packet never reached Content Acquirer to middle SE • Checked reachability by running the **ping** command at the same time as when the 504 status code was seen. The ping request was getting dropped all of the sudden. • System administrators confirmed that Content Acquirer was on a different switch, and that here might be some other load on that switch.
• Two-tier system • 504 seen on the edge SE • Content Acquirer had 200 hits for the same entry	• Client to the edge SE has 40 GB network capacity, while edge SE to Content Acquirer has only 4 GB network capacity • Client simulator puts out 2,000 Sim Users with 3Mbps traffic for peak (6 Gbps), stressing the 4-GB limit between edge SE and Content Acquirer • Smooth traffic eventually balances itself out with lower Mbps, but some 504 status codes are seen between the Content Acquirer and middle SE, resulting in packet drops • Output from the **netmon** command shows steady traffic increase followed by sudden drop
• Two-tier system • 504 seen on the edge SE • 200 seen on the Content Acquirer for the same request	• Client to edge SE has 2 GB throughput capacity, which was not exceeded in the tests • Origin Server to Content Acquirer has 2 GB throughput capacity, which was not exceeded in the tests • Disk is not a bottleneck on the Content Acquirer as verified by iostat • CPU is not a bottleneck on the Content Acquirer as verified by mpstat • Running netmon on both the edge SE and Content Acquirer reveals that all edge interfaces were used on edge SE; however, only one or two interfaces on the Content Acquirer were used • Found that the Cisco Catalyst 3750 switch is used in the test bed, which only supports IP-based load balancing. SEs have interfaces bound to a particular port channel with one IP address so that only one interface on the switch is selected for the data transfer between the edge SE and Content Acquirer; therefore, the 1-GB interface can cause issue when traffic is over 1 GB

Table A-1 Scenarios of 504 Status Codes (continued)

Scenario	Investigation Results
Issues Inside the Upstream SEs	
Tools used to investigate were the following: mpstat and iostat.	
• Three-tier system • 504 seen on edge and middle SEs • Content Acquirer had 20 million content objects cached	• Output from the **isostat** command for certain disks on the Content Acquirer was sometimes showing individual request serving time in range of seconds • Output from the **iostat** command for particular disk showed the usage was 100 percent, and the time interval between I/O requests issued to the device, and being served from the device, was very high • Output from the **show alarms history detail** command showed CPU and diskthreshold alarm raised
• Three-tier system • 504 seen on edge and middle SEs • Reload Content Acquirer	• Only CDE250-2M0 shows this behavior and only during reload • After investigating components (network and disk), questioned lookup queue response • Output from the **show statistics web-engine detail** command showed large outstanding lookup requests on the Content Acquirer • Inserted an instrument into lookup queue size check, which showed queue size for lookup increasing to 200–300 range during reload, while in normal operation the queue was always in the single digits. • Observed CPU cycle during reload, it showed that even though there were some free cycles, it was not enough to finish sudden intake of request lookup • Content Manager scans during the beginning of a reload, and because the CDE250-2M0 has different disk types, it takes up a lot of the CPU, leaving a large lookup queue
• -Two-tier system • 504 seen on edge SE • 504 and NONE/000 seen on Content Acquirer • 20 million content objects cached on both edge SE and Content Acquirer • 800 transactions per second • Small file (240 KB) all unique cache miss • CDE250-2G2 • No core dump files • Long running time (92 hours)	• Output from the **show alarms history detail** command on the edge SE showed servicedead, memory, session, CPU, disk, and cal_diskwrite_exceed alarms raised • Output from the **show alarms history detail** command on the Content Acquirer showed CPU, disk, and cal_diskwrite_exceed alarms raised • CDE250-2G2 capacity exceeded

Table A-1 Scenarios of 504 Status Codes (continued)

Scenario	Investigation Results
Configuration Related Issue (Only applicable if trace-level logging is enabled)	
• Three-tier system • 504 seen on edge and middle SE	• Trace-level errorlog was enabled on all SEs • Top showed that unified_errlogd, the process handling errorlog writing, used a lot of the CPU, when CPU usage hit 100 percent, 504 started happening • Traces in the system can be enabled for debugging purposes; however, if they are enabled, they cause a very large number of disk I/O accesses, which causes the Content Acquirer to take a long time to respond