

Table of Contents

<u>Determining Savings When Using Caching</u>	1
<u>Document ID: 21974</u>	1
<u>Introduction</u>	1
<u>Before You Begin</u>	1
<u>Conventions</u>	1
<u>Prerequisites</u>	1
<u>Components Used</u>	1
<u>Determine Savings</u>	1
<u>Proxy Caching</u>	2
<u>Transparent Caching</u>	2
<u>Using a Router to Determine Bandwidth Savings</u>	2
<u>Explanation of the Show Statistics Commands</u>	4
<u>Related Information</u>	6

Determining Savings When Using Caching

Document ID: 21974

Introduction

Before You Begin

Conventions

Prerequisites

Components Used

Determine Savings

Proxy Caching

Transparent Caching

Using a Router to Determine Bandwidth Savings

Explanation of the Show Statistics Commands

Related Information

Introduction

This document shows how cache savings can be calculated using the tools and commands available on Cache Engines, Content Engines, and Routers.

Before You Begin

Conventions

For more information on document conventions, see the Cisco Technical Tips Conventions.

Prerequisites

Readers of this document should be knowledgeable of the following:

- Transparent and Proxy Caching
- Web Cache Communication Protocol (WCCP)
- Cisco IOS©
- Hypertext Transfer Protocol (HTTP)

Components Used

The information in this document is based on the software and hardware versions below.

- Cisco IOS 12.1 through to 12.2.7
- Caching – Content Software 2.5.1 and ACNS 4.X
- All Route Platforms
- All Caching – Content Platform CE507, CE560, CE590, CE7320, CE505, CE550, CE570

Determine Savings

There are three areas of discussion when looking at the savings available for caching; transparent caching, proxy caching and using the routers interface counters.

Cisco – Determining Savings When Using Caching

Proxy Caching

In a proxy caching setup where the clients browses are set to point directly to the cache engine on a predetermined port the calculation of savings is straight forward.

There are different commands that are used to determine exactly how the cache is performing.

show statistics http savings

show statistics ftp

show statistics wmt. savings (Only available in ACNS 4.X and above)

show statistics mediacache real savings (Only available in ACNS 4.X and above)

Note: Although you can set the secure proxy (https) up in the proxy settings on a browser we only tunnel this session and can not cache it because it is encrypted.

Transparent Caching

In a transparent caching setup, where no settings are required in the clients browser and traffic is intercepted at a router and sent to the cache using the Web Cache Communications Protocol(WCCP), the calculation of saving is just as straight forward.

These are the commands needed to determine the savings.

show statistics http savings

show statistics wmt savings (Only available in ACNS 4.X and above)

show statistics mediacache real savings (Only available in ACNS 4.X and above)

Note: In transparent mode we can not cache ftp or https sessions. These sessions are not sent to the cache and are passed directly to the origin server.

Using a Router to Determine Bandwidth Savings

When using a router to determine the saving there are a number of things to consider to get an accurate measure of the savings in raw bandwidth. The most important thing to do is get a baseline. In order to do this you need to monitor the link utilization of the upstream link from your router that is connected to the internet. In a reverse proxy caching (RPC) setup you need to monitor the load on your servers. There are many tools available to monitor link utilization via SNMP. One freely available tool is MRTG, this application is not supported or supplied by Cisco. It can be found at [Multi Router Traffic Grapher](#) .

On a Cisco router you monitor the following counters on the **show interface** command on the link to the internet, bytes in and bytes out. In order to get a base line you need to understand the makeup of the traffic heading out to the internet. In transparent mode only http, wmt(1) and real(1) requests are sent to the CE. While in proxy mode ftp(2) and https(3) can be sent to the cache as well as http, wmt(4) and real(4). A traffic analyzer is an invaluable tool in this situation. Ports used for the protocols mentioned are as follows. These are the default port and can be changed.

http tcp 80

```
https tcp 443
ftp tcp 20 and 21
wmt tcp 1755 or udp 1755 or http or multicast
real rtsp 554
```

- (1) Only if configured as a WCCP service.
- (2) This only applies when the ftp session is down from within a browser or application that has the ftp proxy set to the cache.
- (3) Although HTTPS (Secure) sessions are sent to the cache, they can not be cache and are only tunneled through.
- (4) The proxy setting needs to be set within the Windows Media Player or the Real Player. Please consult the user's guide for information on how to set these.

The methodology for calculation of savings using interface counters is as follows:

1. Without caching turned on clear the interface counters for the connection to the internet (**clear counters** is the command for Cisco IOS).
2. Wait for a 2 average days and note down the highlighted counter values below. Note the time that the counters were running for.
3. Turn your caching on, whether it be proxy or transparent caching,. You need to wait for the cache to populate. This should be about 3 days depending on load. Savings should be seen after a few hours, but three days will give you a good base line.
4. Clear the interface counters on the connection to the internet again.
5. With the caching turned on wait the same amount of time as with caching turned off.
6. Note the interface byte counters again.
7. The difference is your bandwidth savings. This will be a different percentage compared to the sh stat http saving due to the router interface seeing all traffic while the cache only sees the protocols that are going to be cached.

Note: The below stats are from a router that is doing only http and hence the good figures. On a production router this would be far less.

Note: Show interface without caching.

```
Serial0/0 is up, line protocol is up
  Hardware is PowerQUICC Serial
  Internet address is 10.64.21.10/30
  MTU 1500 bytes, BW 128 Kbit, DLY 20000 usec,
    reliability 255/255, txload 59/255, rxload 194/255
  Encapsulation HDLC, loopback not set
  Keepalive set (10 sec)
  Last input 00:00:00, output 00:00:00, output hang never
  Last clearing of "show interface" counters 00:06:52
  Input queue: 4/75/0/0 (size/max/drops/flushes); Total output drops: 6
  Queueing strategy: weighted fair
  Output queue: 0/1000/64/6 (size/max total/threshold/drops)
    Conversations 0/32/32 (active/max active/max total)
    Reserved Conversations 0/0 (allocated/max allocated)
    Available Bandwidth 96 kilobits/sec
  5 minute input rate 177000 bits/sec, 47 packets/sec
  5 minute output rate 30000 bits/sec, 44 packets/sec
    14218 packets input, 8743319 bytes, 0 no buffer
    Received 42 broadcasts, 0 runts, 0 giants, 0 throttles
    0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
```

```

13019 packets output, 1113797 bytes, 0 underruns
0 output errors, 0 collisions, 0 interface resets
0 output buffer failures, 0 output buffers swapped out
0 carrier transitions
DCD=up DSR=up DTR=up RTS=up CTS=up

```

Note: Show interface with caching turned on.

```

Serial0/0 is up, line protocol is up
Hardware is PowerQUICC Serial
Internet address is 10.64.21.10/30
MTU 1500 bytes, BW 128 Kbit, DLY 20000 usec,
    reliability 255/255, txload 1/255, rxload 1/255
Encapsulation HDLC, loopback not set
Keepalive set (10 sec)
Last input 00:00:06, output 00:00:07, output hang never
Last clearing of "show interface" counters 00:17:33
Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
Queueing strategy: weighted fair
Output queue: 0/1000/64/0 (size/max total/threshold/drops)
    Conversations 0/11/32 (active/max active/max total)
    Reserved Conversations 0/0 (allocated/max allocated)
    Available Bandwidth 96 kilobits/sec
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
7503 packets input, 5408948 bytes, 0 no buffer
Received 105 broadcasts, 0 runts, 0 giants, 0 throttles
0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
5723 packets output, 497401 bytes, 0 underruns
0 output errors, 0 collisions, 0 interface resets
0 output buffer failures, 0 output buffers swapped out
0 carrier transitions
DCD=up DSR=up DTR=up RTS=up CTS=up

```

Explanation of the Show Statistics Commands

Show statistics http savings

```

cache-1#show statistics http savings
                Statistics - Savings
                Requests                Bytes
-----
Total:                663699                5369304364
Hits:                 261820                583380690
Miss:                 401782                4612002188
Savings:                39.4 %                10.9 %

```

The percentages are the figures that give you the best indication. As you can see from the above output, we have requests savings of 39.4% and byte savings of 10%. This is within the expected range of savings.

Show statistics ftp

```

cache-1#show statistics ftp

FTP Statistics
-----

FTP requests Received = 6

FTP Hits

Requests Percentage

```

```

Number of hits =                3        50.0 %
Bytes =                54171214        50.0 %

```

FTP Misses

```

Requests Percentage
Number of misses =                3        50.0 %
Bytes =                54171214        50.0 %

```

```

Requests sent to Outgoing Proxy    = 0
Requests sent to origin ftp server = 3

```

FTP error count = 0

Show statistics wmt savings

cache-1#show statistics wmt savings

Unicast Savings Statistics

```

=====
Total bytes saved: 15271392
-----

```

	Total	% of Total Bytes Saved
By Pre-positioned content:	0	0.00%
By Live-splitting:	11770875	77.08%
By Cache-hit:	3500517	22.92%
Total		% of Total Live Outgoing Bytes

Live Splitting

```

-----
Incoming bytes:                26367513        69.14%
Outgoing bytes:                38138388        100.00%
Bytes saved:                    11770875        30.86%

Total                            % of Bytes Cache Total
-----

```

Caching

```

-----
Bytes cache-miss:                10761747        75.46%
Bytes cache-hit:                  3500517        24.54%
Bytes cache-total:                14262264        100.00%

Bytes cache-bypassed:                0

Total                            % of Req Cache Total
-----

```

Cacheable requests

```

-----
Req cache-miss:                    1        16.67%
Req cache-hit:                      1        16.67%
Req cache-partial-hit:              4        66.67%
Req cache-total:                    6        100.00%

Req cache-bypassed:                0

```

Objects not cached

```

-----
Cache bypassed:                    0
Exceed max-size:                    0

```

Show statistics mediacache real savings

```
ce-7320-cache#show statistics mediacache real savings
```

```
Media Cache Statistics - Savings
Requests                               Bytes
-----
Total:                                0                               0
Hits:                                  0                               0
Miss:                                   0                               0
Savings:                               0.0 %                            0.0 %
```

Related Information

- [WCCP V1](#)
 - [WCCP V2](#)
 - [ACNS Documentation](#)
 - [Caching Documentation](#)
 - [WCCP Commands](#)
 - [Technical Support – Cisco Systems](#)
-

All contents are Copyright © 1992–2005 Cisco Systems, Inc. All rights reserved. Important Notices and Privacy Statement.

Updated: Mar 15, 2005

Document ID: 21974
