Technical Tip

TT-0501402a -- Information -- 10-Jan-2005

NORTEL

Insert mode cookie-based persistence

Contents

Contents 1
Introduction 1
Associated Products: 1
Overview 1
  Nortel Web and Application switch cookie persistency 3
  Insert cookie mode 3
Sample Configuration 4
  Setup 4
  Configuring the PC 4
  Configuring WebServer1 6
  Configuring WebServer2 7
  Configuring Alteon 7
    Configuring interfaces 7
    Configuring real servers 8
    Assigning real servers to a group 8
    Configuring the virtual server 9
    Enabling client/server processing on ports 10
    Enabling direct access mode 10
    Enabling SLB globally 10
  Testing the configuration 11
    Testing with expiration timer set 20
Appendix A. Alteon configuration file 23
Appendix B. Alteon configuration file with cookie expiration timer 24

Introduction

This document illustrates the operation and sample configuration of insert mode cookie-based persistence in Nortel Web and Application switches (formerly known as Alteon Web and Application switches).

Associated Products:

The information in this document is intended to be used with the following product(s) with the indicated software or hardware revisions:

Product Name or Order Number	Revision Information	
	Potentially Affected	*Corrected*
Nortel Web and Application switches (formerly known as Alteon Web and Application switches): 180e, 180 Plus, 184, AD2, AD3, AD4, 2224, 2424, 2208, 2216, 3408	All	N/A

Overview

Cookies are a mechanism used to create a stateful session with HTTP requests and responses. Prior to the introduction of the cookie concept, HTTP servers would respond to each client request without relating that request to subsequent requests. The concept of cookies allows clients and servers that wish to exchange state information to place HTTP requests and responses within a larger context, or a so-called session. This context could be used to maintain a state of the session that could be necessary for some web applications. For example, a stateful session could be used for e-commerce, when the state would reflect the client's product selections or shopping cart. Or it could be used by web server administrators to count how many returning clients have visited the web site.

©2005 Nortel Networks Limited. All Rights Reserved

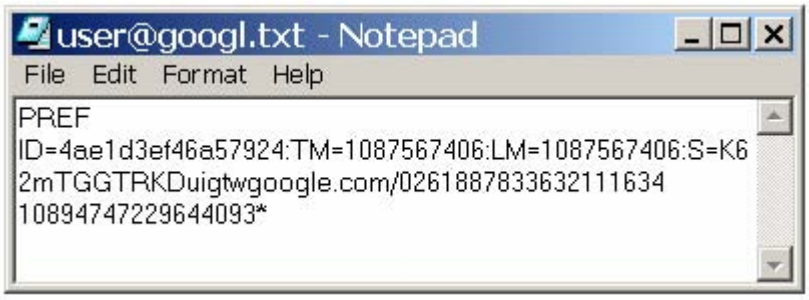
Cookies are strings that are placed into the HTTP header and passed from the server to the client's web browser. When the client initiates a connection to the server, it establishes a TCP session with a server and passes the HTTP request. A cookie-enabled server processes the HTTP request and sends the client an HTTP response containing an extra header: the Set-Cookie header. The Set-Cookie header contains:

- a cookie name and its value
- an optional comment
- an optional domain name (always starts with a leading dot)
- the optional max age of the cookie (if the max age is not specified in the Set-Cookie header, the user agent deletes the cookie as soon as the client closes the web browser)
- an optional path that specifies the subset of URLs which a particular cookie applies to
- an optional secure attribute that directs the user agent to use only a secure means to connect to the server
- a version attribute that identifies the version of the state management specification

For example, below is a Set-Cookie header sent by the HTTP server at www.google.com:

```
Set-Cookie: PREF=ID=4ae1d3ef46a57924:TM=1087567406:LM=1087567406:S=K62mTGGTRKDuigtw;
expires=Sun, 17-Jan-2038 19:14:07 GMT; path=/; domain=.google.com\r\n
```

Once the client receives the cookie in the HTTP response, it stores it locally (unless configured to ignore cookies), usually in a form of a text file. Note that the location and form of storage depend on the web browser and operating system on a client's PC. For example, the received cookie from www.google.com could be stored as follows:



Once the cookie is received (assuming the client's web browser supports cookies), the client uses that cookie to send subsequent requests to the server. The cookie request header is used to send the cookie information back to the server. The cookie header contains the same attributes as the received Set-Cookie directive, with the exception of the comment, max age and secure attributes that are not sent back to the server.

For example, below is a subsequent request to the www.google.com:

```
Hypertext Transfer Protocol
    GET /images/logo.gif HTTP/1.1\r\n
    Accept: */*\r\n
    Referer: http://www.google.com/\r\n
    Accept-Language: en-us\r\n
    Accept-Encoding: gzip, deflate\r\n
    User-Agent: Mozilla/4.0 (compatible; MSIE 5.5; Windows NT 5.0; T312461; PC SRP
2.0)\r\n
    Host: www.google.com\r\n
    Connection: Keep-Alive\r\n
    Cookie:
PREF=ID=4ae1d3ef46a57924:TM=1087567406:LM=1087567406:S=K62mTGGTRKDuigtw\r\n \r\n
```

The server might ignore the received cookie or use it to determine the state of a session.

©2005 Nortel Networks Limited. All Rights Reserved

Nortel Web and Application switch cookie persistency

Nortel Web and Application switches support cookie-based persistency that allows the switch to redirect clients to the real server that initially processed the request. With cookie-based persistent load balancing, once an initial real server is selected based on the configured group metric, the switch forwards the subsequent requests based on the cookie header in the client's HTTP request.

Nortel Web and Application switches can operate in three modes of operation – insert cookie, passive cookie and rewrite cookie. This document illustrates the insert cookie mode. For information on passive or rewrite cookie modes, please consult the technical tips titled "Passive mode cookie-based persistence" or "Rewrite mode cookie-based persistence".

Insert cookie mode

When operating in insert cookie mode, the switch inserts a cookie into the HTTP response on behalf of a real server, thus the real server does not need to have cookie server software installed on it.

When the client sends an initial HTTP request with no cookies set in it, the switch performs load balancing and selects a real server based on the configured metric (Figure 1).

The real server receives the request, processes it and sends an HTTP reply back to the client, no cookies are inserted by the real server as it was not configured to so.

The switch receives an HTTP reply, calculates a cookie value, inserts it into the HTTP reply, and forwards the reply back to the client PC. If the client needs to make a subsequent request to the server, it inserts the cookie request header into HTTP request. When the switch receives an HTTP request containing its previously inserted cookie value, it looks up the real server associated with the cookie and forwards the request to the initially selected server. Thus persistency is achieved.

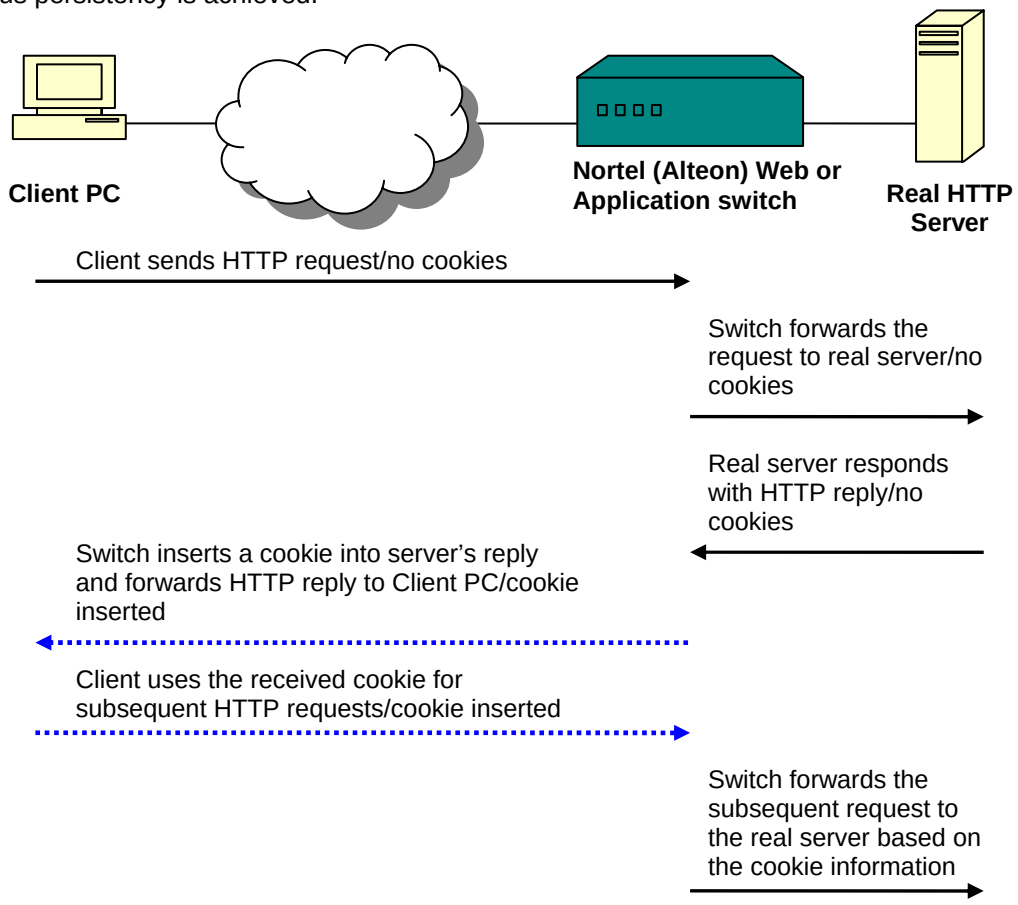
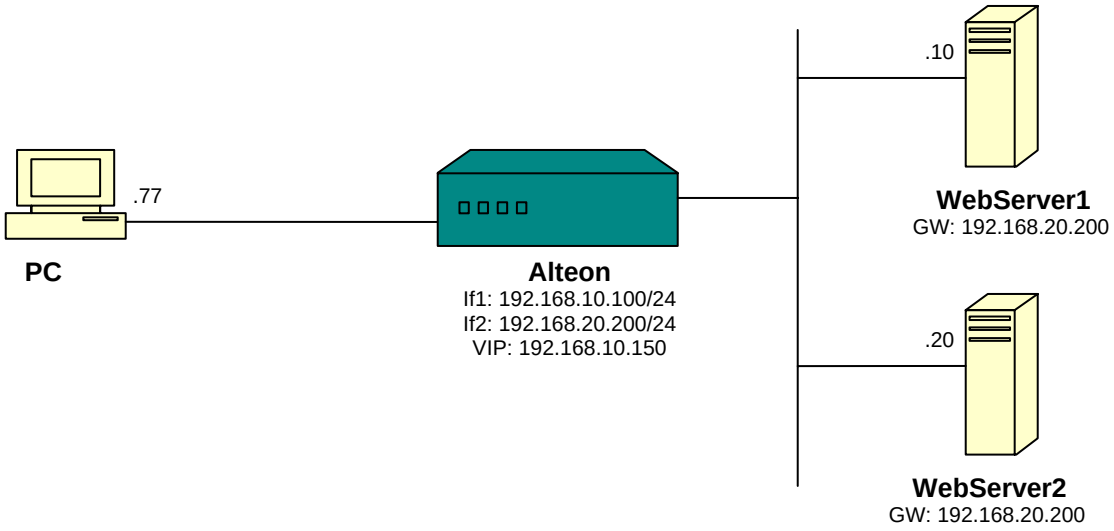


Figure 1

©2005 Nortel Networks Limited. All Rights Reserved

Sample Configuration

Setup



- **PC** – Windows 2000 professional with Microsoft Internet Explorer 5.5 installed, IP address 192.168.10.77/24
- **Alteon** – Nortel Web switch AD3, code version 10.0.30.7, interface 1 IP 192.168.10.100/24, interface 2 IP 192.168.20.200/24, Virtual IP 192.168.10.150, port 1 – client port, port 8 – server port
- **WebServer1** – Linux RedHat 7.3 with apache software installed, IP address 192.168.20.10/24
- **WebServer2** – Linux RedHat 7.3 with apache software installed, IP address 192.168.20.20/24

The goal of the configuration is to configure insert mode cookie persistency on Alteon, where PC would be redirected to the same servers it was originally redirected to.

Configuring the PC

1. Configure the IP address (192.168.10.77/24) on the PC:

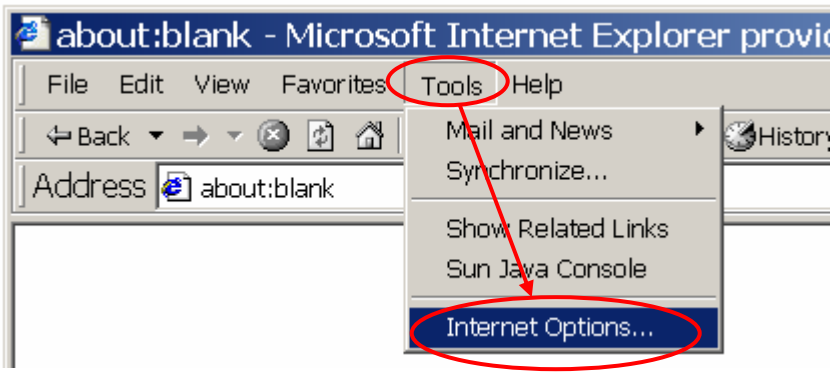
```
C:\>ipconfig

Windows 2000 IP Configuration
Ethernet adapter Local Area Connection:

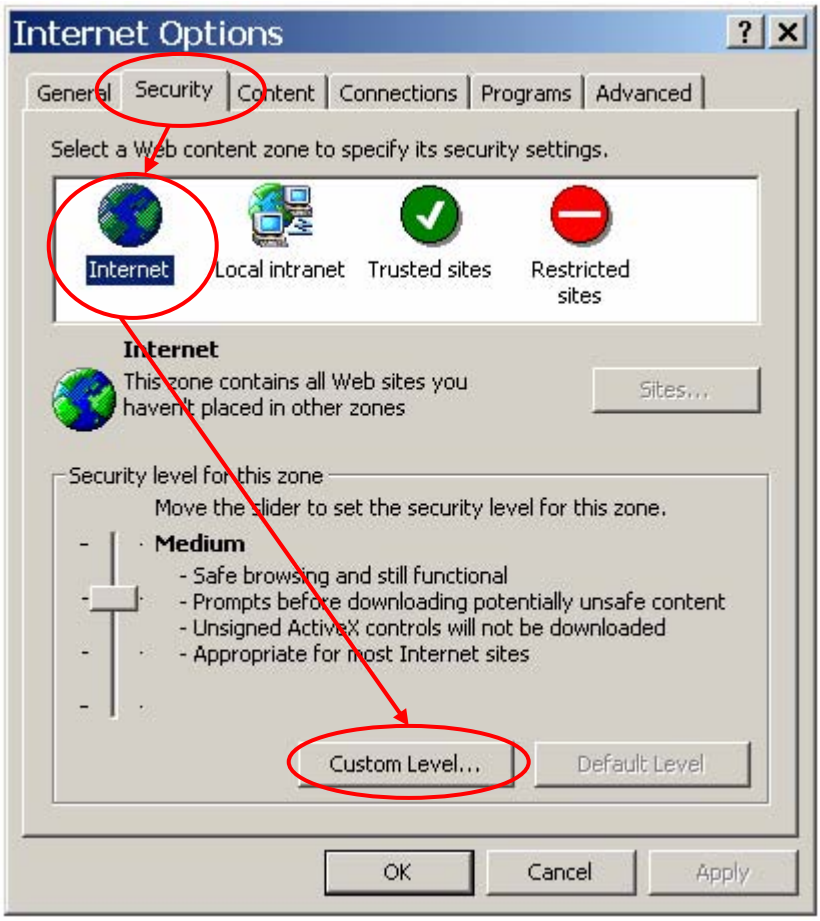
        Connection-specific DNS Suffix  . :
        IP Address. . . . . . . . . . . . : 192.168.10.77
        Subnet Mask . . . . . . . . . . . : 255.255.255.0
        Default Gateway . . . . . . . . . :
```

©2005 Nortel Networks Limited. All Rights Reserved

2. It is assumed that cookies are enabled for the browser. If this is not the case, enable cookies.
   a) Navigate **Tools→Internet Options** from the browser menu:

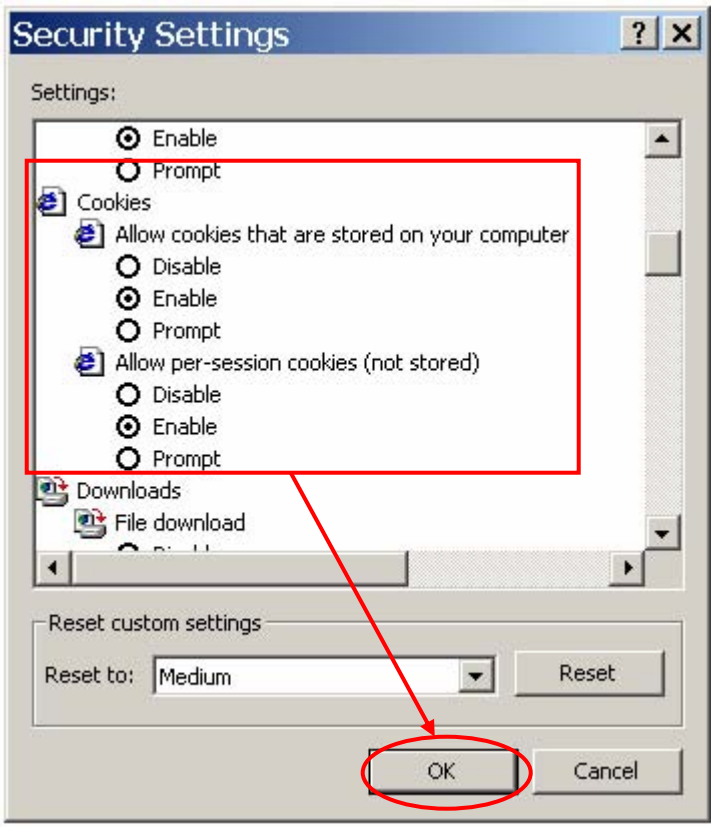


   b) The Internet Options window appears. Click on the **Security** tab.
   c) Select **Internet** and click **Custom Level**:



©2005 Nortel Networks Limited. All Rights Reserved

d) Scroll down to the **Cookies** section and select **Enable** for **Cookies**.
e) Click **OK**:



f) In the **Internet Options** window, click **OK**.

Configuring WebServer1

1. Configure the IP address (192.168.20.10/24) on WebServer1:

```
[root@linux7 user]# /sbin/ifconfig eth0 192.168.20.10 netmask 255.255.255.0
```

2. Configure Alteon's interface 2 IP (192.168.20.200) as a default gateway:

```
[root@linux7 user]# /sbin/route add default gw 192.168.20.200
```

3. Replace the default index.html file in the Apache root directory (/var/www/http) with a sample html file that will allow us to distinguish between two the web sites. For example:

```
<HTML>
 <HEAD>
  <TITLE>www.example-1.com</TITLE>
 </HEAD>
 <BODY>
  <H1 ALIGN="CENTER">www.example-1.com</H1>
 </BODY>
</HTML>
```

4. Start the HTTP server:

```
[root@linux7 user]# /etc/init.d/httpd start
Starting httpd:                                            [  OK  ]
```

©2005 Nortel Networks Limited. All Rights Reserved

Configuring WebServer2

1. Configure the IP address (192.168.20.20/24) on WebServer2:

```
[root@linux7 user]# /sbin/ifconfig eth0 192.168.20.20 netmask 255.255.255.0
```

2. Configure Alteon's interface 2 IP (192.168.20.200) as a default gateway:

```
[root@linux7 kristina]# /sbin/route add default gw 192.168.20.200
```

3. Replace the default index.html file in the root directory with a sample html file that will allow us to distinguish between two the web sites. For example:

```
<HTML>
 <HEAD>
  <TITLE>www.example-2.com</TITLE>
 </HEAD>
 <BODY>
  <H1 ALIGN="CENTER">www.example-2.com</H1>
 </BODY>
</HTML>
```

4. Start HTTP server:

```
[root@linux7 user]# /etc/init.d/httpd start
Starting httpd:                                            [  OK  ]
```

Configuring Alteon

Note: Configuration of Alteon has been started from factory default settings.

1. Log in to the switch:

```
Enter password:
System Information at 11:05:11 Wed Apr 21, 2004
Alteon AD3
sysName:
sysLocation:
. . .
```

2. Select **n** to not run the setup script:

```
The switch is booted with factory default configuration.
  To ease the configuration of the switch, a "Set Up" facility which
  will prompt you with those configuration items that are essential
  to the operation of the switch is provided.
Would you like to run "Set Up" to configure the switch? [y/n] n
```

Configuring interfaces

1. Configure the IP address (192.168.10.100/24) for interface 1 and enable the interface:

```
>> Main# /cfg/ip/if 1/mask 255.255.255.0/addr 192.168.10.100/ena
Current subnet mask:     0.0.0.0
New pending subnet mask: 255.255.255.0
Current IP address:     0.0.0.0
New pending IP address: 192.168.10.100
Pending new broadcast address:  192.168.10.255
Current status: disabled
New status:     enabled
```

©2005 Nortel Networks Limited. All Rights Reserved

2. Configure the IP address (192.168.20.200/24) for interface 2 and enable the interface:

```
>> IP Interface 1# /cfg/ip/if 2/mask 255.255.255.0/addr 192.168.20.200/ena
Current subnet mask:     0.0.0.0
New pending subnet mask: 255.255.255.0
Current IP address:     0.0.0.0
New pending IP address: 192.168.20.200
Pending new broadcast address:  192.168.20.255
Current status: disabled
New status:     enabled
```

3. Apply and save the changes:

```
>> IP Interface 2# apply/save
------------------------------------------------------------------
Apply complete; don't forget to "save" updated configuration.
Request will first copy the FLASH "active" config to "backup",
 then overlay FLASH "active" with new config.
Confirm saving to FLASH [y/n]: y
New config successfully saved to FLASH.
Switch is currently set to use factory default config block on next boot.
Do you want to change that to the active config block? [y/n] y
Next boot will use active config block.
```

Configuring real servers

1. Configure real server 1. Enter the real server 1 configuration menu, set the real server 1 IP address (192.168.20.10) and enable the real server:

```
>> IP Interface 2# /cfg/slb/real 1/rip 192.168.20.10/ena
Current real server IP address:     0.0.0.0
New pending real server IP address: 192.168.20.10
Current status: disabled
New status:     enabled
```

2. Configure real server 2. Enter the real server 2 configuration menu, set the real server 2 IP address (192.168.20.20) and enable the real server:

```
>> Real server 1 # /cfg/slb/real 2/rip 192.168.20.20/ena
Current real server IP address:     0.0.0.0
New pending real server IP address: 192.168.20.20
Current status: disabled
New status:     enabled
```

Assigning real servers to a group

1. Define a real servers group, and place real server 1 and 2 into this group:

```
>> Real server 2 # /cfg/slb/group 1/add 1/add 2
Real server 1 added to real server group 1.
Real server 2 added to real server group 1.
```

2. Apply the appropriate metric for load balancing. If the IP address of the real server handling the request is to be embedded into a cookie, select round robin or least connections metric type. If the IP address is not to be embedded, select hash metric. In this example, round robin metric is used:

```
>> Real server group 1# metric round
Current metric:         leastconns
New pending metric:     roundrobin
```

©2005 Nortel Networks Limited. All Rights Reserved

3. Set the health check to TCP:

```
>> Real server group 1# health tcp
Current health check type:      tcp
New pending health check type:  tcp
```

Configuring the virtual server

1. Configure the virtual server (server 1 in this example). Set the IP address for the virtual server, and enable the virtual server:

```
>> Real server group 1# /cfg/slb/virt 1/vip 192.168.10.150/ena
Current virtual server IP address:     0.0.0.0
New pending virtual server IP address: 192.168.10.150
Current status: disabled
New status:     enabled
```

2. Assign HTTP service to the virtual server and assign the created group 1 to the server:

```
>> Virtual Server 1# service http/group 1
Current real server group:     1
New pending real server group: 1
```

3. Enable delayed binding. Delayed binding is required as layer 7 information (HTTP) is not sent in the first packet from the client:

```
>> Virtual Server 1 http Service# dbind ena
Current delayed binding: disabled
New delayed binding:     enabled
```

4. Enable persistent binding.
    a) When prompted, enter **cookie** as a persistence mode.
    b) Enter **i** to select insert mode of operation.
    c) Enter the appropriate insert cookie expiration time.
    d) Time could be entered as an absolute time in the form of MM/dd/yy[@hh:mm], relative time specifying the elapsed time from the time when cookie was created in the form of days[:hours[:minutes]] (note that the only valid time zone is the GMT time zone), or set to none (press Enter key) indicating that the cookie should be deleted with the end of the user session.
    e) Select y if the domain should be included in the set-cookie directive, or n if it should not be included. In this example, **n** was selected:

```
>> Virtual Server 1 http Service# pbind
Current persistent binding mode: disabled
Enter clientip|cookie|sslid|disable persistence mode:   cookie
Enter passive|rewrite|insert cookie persistence mode [p/r/i]:   i
Enter insert-cookie expiration as either :
... a date     <MM/dd/yy[@hh:mm]> (e.g. 12/31/01@23:59)
... a duration <days[:hours[:minutes]]> (e.g. 45:30:90)
... or none    <return>
Enter cookie expiration:
Insert cookie domain name? (y/n) [n]n
```

©2005 Nortel Networks Limited. All Rights Reserved

Enabling client/server processing on ports
Enable client/server processing on the appropriate ports. In this configuration, the client is connected to port 1 and both servers are connected through the hub to port 8, thus client processing should be enabled on port 1 and server processing should be enabled on port 8:

```
>> Virtual Server 1 http Service# /cfg/slb/port 1/client e
Current client processing: disabled
New client processing:     enabled

>> SLB port 7# /cfg/slb/port 8/server e
Current server processing: disabled
New server processing:     enabled
```

Enabling direct access mode
Enable direct access mode (DAM), required for the delayed binding to be enabled:

```
>> SLB port 1# /cfg/slb/adv/direct e
Current Direct Access Mode: disabled
New Direct Access Mode:     enabled
```

Note: Alternatively, Proxy IP addresses could be assigned to every port and direct access mode could be disabled.

Enabling SLB globally

1. Enable SLB globally:

```
>> Layer 4 Advanced# /cfg/slb/on
Current status: OFF
New status:     ON
```

2. Apply and save the changes:

```
>> Layer 4# apply/save
------------------------------------------------------------------
Apply complete; don't forget to "save" updated configuration.
Request will first copy the FLASH "active" config to "backup",
 then overlay FLASH "active" with new config.
Confirm saving to FLASH [y/n]:y
New config successfully saved to FLASH.
```

Note: As soon as the changes are applied, the following messages appears on the console indicating that the real servers and virtual server are up and operational:

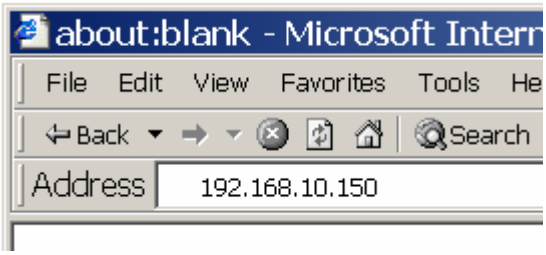
```
Jun 18 13:41:32 NOTICE  slb: real service 192.168.20.10:80 operational
Jun 18 13:41:32 NOTICE  slb: real server 192.168.20.10 operational
Jun 18 13:41:32 NOTICE  slb: Services are available for Virtual Server
1:192.168.10.150
Jun 18 13:41:33 NOTICE  slb: real service 192.168.20.20:80 operational
Jun 18 13:41:33 NOTICE  slb: real server 192.168.20.20 operational
```

©2005 Nortel Networks Limited. All Rights Reserved

Testing the configuration

1. Start traffic capture on the PC using, for example, Ethereal. Start traffic capture on one of the WebServers using, for example, tcpdump.

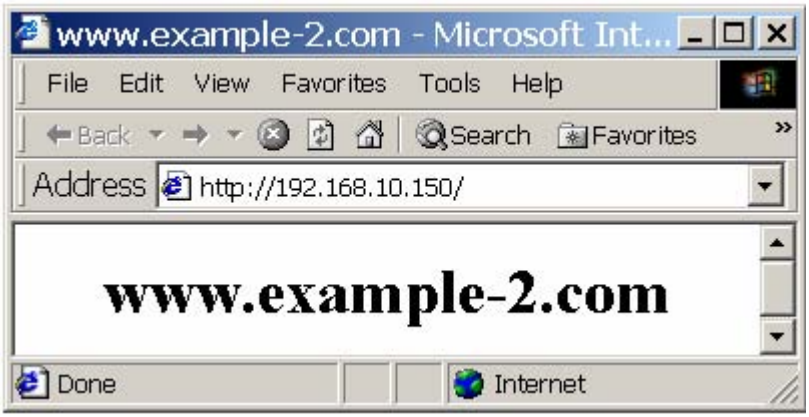
2. Open the web browser, enter the IP address of the virtual server (192.168.10.150), and press Enter:



3. While the page loads, check the session table on Alteon. WebServer2 (192.168.20.20) was selected by the configured metric algorithm:

```
>> Session Table Information# /info/slb/sess/dump
 8,4:  192.168.10.77 4519, 192.168.10.150 80 -> 4100 192.168.20.20 80  age 4  E
```

4. The WebServer2 page loads:



5. Click the Refresh button on the web browser. Check the session table again. The request has been forwarded to the same server:

```
>> Session Table Information# /info/slb/sess/dump
 8,6:  192.168.10.77 4521, 192.168.10.150 80 -> 4102 192.168.20.20 80  age 10
```

©2005 Nortel Networks Limited. All Rights Reserved

6. Let's take a look at the captured traffic.

PC (192.168.10.77) initiates an HTTP connection by sending TCP SYN to the virtual server IP (192.168.10.150):

```
Frame 118 (62 bytes on wire, 62 bytes captured)
Ethernet II, Src: 00:08:74:9a:e5:85, Dst: 00:60:cf:46:53:6e
Internet Protocol, Src Addr: 192.168.10.77 (192.168.10.77), Dst Addr: 192.168.10.150
(192.168.10.150)
Transmission Control Protocol, Src Port: 4519 (4519), Dst Port: http (80), Seq:
1398194185, Ack: 0, Len: 0
    Source port: 4519 (4519)
    Destination port: http (80)
    Sequence number: 1398194185
    Header length: 28 bytes
    Flags: 0x0002 (SYN)
    Window size: 65535
    Checksum: 0xc794 (correct)
    Options: (8 bytes)
```

Due to the enabled delayed binding, Alteon proceeds with a handshake on behalf of the real server and responds to PC with TCP ACK:

```
Frame 119 (62 bytes on wire, 62 bytes captured)
Ethernet II, Src: 00:60:cf:46:53:6e, Dst: 00:08:74:9a:e5:85
Internet Protocol, Src Addr: 192.168.10.150 (192.168.10.150), Dst Addr: 192.168.10.77
(192.168.10.77)
Transmission Control Protocol, Src Port: http (80), Dst Port: 4519 (4519), Seq:
1316551039, Ack: 1398194186, Len: 0
    Source port: http (80)
    Destination port: 4519 (4519)
    Sequence number: 1316551039
    Acknowledgement number: 1398194186
    Header length: 24 bytes
    Flags: 0x0012 (SYN, ACK)
    Window size: 65535
    Checksum: 0x9492 (correct)
    Options: (4 bytes)
```

The client acknowledges the SYN ACK with an ACK:

```
Frame 120 (54 bytes on wire, 54 bytes captured)
Ethernet II, Src: 00:08:74:9a:e5:85, Dst: 00:60:cf:46:53:6e
Internet Protocol, Src Addr: 192.168.10.77 (192.168.10.77), Dst Addr: 192.168.10.150
(192.168.10.150)
Transmission Control Protocol, Src Port: 4519 (4519), Dst Port: http (80), Seq:
1398194186, Ack: 1316551040, Len: 0
    Source port: 4519 (4519)
    Destination port: http (80)
    Sequence number: 1398194186
    Acknowledgement number: 1316551040
    Header length: 20 bytes
    Flags: 0x0010 (ACK)
    Window size: 65535
    Checksum: 0xac4f (correct)
```

©2005 Nortel Networks Limited. All Rights Reserved

Once the handshake is complete, the client PC sends an HTTP request:

```
Frame 121 (433 bytes on wire, 433 bytes captured)
Ethernet II, Src: 00:08:74:9a:e5:85, Dst: 00:60:cf:46:53:6e
Internet Protocol, Src Addr: 192.168.10.77 (192.168.10.77), Dst Addr: 192.168.10.150
(192.168.10.150)
Transmission Control Protocol, Src Port: 4519 (4519), Dst Port: http (80), Seq:
1398194186, Ack: 1316551040, Len: 379
Hypertext Transfer Protocol
    GET / HTTP/1.1\r\n
    Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, application/vnd.ms-
excel, application/vnd.ms-powerpoint, application/msword, application/x-shockwave-
flash, */*\r\n
    Accept-Language: en-us\r\n
    Accept-Encoding: gzip, deflate\r\n
    User-Agent: Mozilla/4.0 (compatible; MSIE 5.5; Windows NT 5.0; T312461; PC SRP
2.0)\r\n
    Host: 192.168.10.150\r\n
    Connection: Keep-Alive\r\n
    \r\n
```

Once the HTTP request is received by Alteon, it checks whether the request already contains a cookie header with real server information. As this is an initial request, no cookie information could be obtained and Alteon selects a server for load balancing (in this example, WebServer2 (192.168.20.20)) based on the metric configured for the group.

Once the real server is selected, Alteon starts a TCP handshake with the real server on behalf of a client by sending a TCP SYN:

```
Frame 122 (64 bytes on wire, 64 bytes captured)
Ethernet II, Src: 00:08:74:9a:e5:85, Dst: 00:0c:29:63:8f:59
Internet Protocol, Src Addr: 192.168.10.77 (192.168.10.77), Dst Addr: 192.168.20.20
(192.168.20.20)
Transmission Control Protocol, Src Port: 4100 (4100), Dst Port: http (80), Seq:
1398194185, Ack: 0, Len: 0
    Source port: 4100 (4100)
    Destination port: http (80)
    Sequence number: 1398194185
    Header length: 24 bytes
    Flags: 0x0002 (SYN)
    Window size: 65535
    Checksum: 0xd4c0 (correct)
    Options: (4 bytes)
```

The real server (192.168.20.20) responds with a SYN ACK:

```
Frame 123 (58 bytes on wire, 58 bytes captured)
Ethernet II, Src: 00:0c:29:63:8f:59, Dst: 00:60:cf:46:53:60
Internet Protocol, Src Addr: 192.168.20.20 (192.168.20.20), Dst Addr: 192.168.10.77
(192.168.10.77)
Transmission Control Protocol, Src Port: http (80), Dst Port: 4100 (4100), Seq:
3642504381, Ack: 1398194186, Len: 0
    Source port: http (80)
    Destination port: 4100 (4100)
    Sequence number: 3642504381
    Acknowledgement number: 1398194186
    Header length: 24 bytes
    Flags: 0x0012 (SYN, ACK)
    Window size: 5840
    Checksum: 0xb005 (correct)
    Options: (4 bytes)
```

©2005 Nortel Networks Limited. All Rights Reserved

Alteon acknowledges it with an ACK:

```
Frame 124 (60 bytes on wire, 60 bytes captured)
Ethernet II, Src: 00:60:cf:46:53:60, Dst: 00:0c:29:63:8f:59
Internet Protocol, Src Addr: 192.168.10.77 (192.168.10.77), Dst Addr: 192.168.20.20
(192.168.20.20)
Transmission Control Protocol, Src Port: 4100 (4100), Dst Port: http (80), Seq:
1398194186, Ack: 3642504382, Len: 0
    Source port: 4100 (4100)
    Destination port: http (80)
    Sequence number: 1398194186
    Acknowledgement number: 3642504382
    Header length: 24 bytes
    Flags: 0x0010 (ACK)
    Window size: 65535
    Checksum: 0xc6d6 (correct)
    Options: (4 bytes)
```

Once the TCP handshake is complete, Alteon sends the client's HTTP request to the real server:

```
Frame 125 (433 bytes on wire, 433 bytes captured)
Ethernet II, Src: 00:60:cf:46:53:60, Dst: 00:0c:29:63:8f:59
Internet Protocol, Src Addr: 192.168.10.77 (192.168.10.77), Dst Addr: 192.168.20.20
(192.168.20.20)
Transmission Control Protocol, Src Port: 4100 (4100), Dst Port: http (80), Seq:
1398194186, Ack: 3642504382, Len: 379
Hypertext Transfer Protocol
    GET / HTTP/1.1\r\n
    Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, application/vnd.ms-
excel, application/vnd.ms-powerpoint, application/msword, application/x-shockwave-
flash, */*\r\n
    Accept-Language: en-us\r\n
    Accept-Encoding: gzip, deflate\r\n
    User-Agent: Mozilla/4.0 (compatible; MSIE 5.5; Windows NT 5.0; T312461; PC SRP
2.0)\r\n
    Host: 192.168.10.150\r\n
    Connection: Keep-Alive\r\n
    \r\n
```

©2005 Nortel Networks Limited. All Rights Reserved

The real server acknowledges the request and replies with an HTTP response:

```
Frame 220 (545 bytes on wire, 545 bytes captured)
Ethernet II, Src: 00:0c:29:63:8f:59, Dst: 00:60:cf:46:53:60
Internet Protocol, Src Addr: 192.168.20.20 (192.168.20.20), Dst Addr: 192.168.10.77
(192.168.10.77)
Transmission Control Protocol, Src Port: http (80), Dst Port: 4100 (4100), Seq:
3642504382, Ack: 1398194565, Len: 491
Hypertext Transfer Protocol
    HTTP/1.1 200 OK\r\n
    Date: Fri, 18 Jun 2004 03:39:31 GMT\r\n
    Server: Apache/1.3.23 (Unix)  (Red-Hat/Linux) mod_ssl/2.8.7 OpenSSL/0.9.6b
DAV/1.0.3 PHP/4.1.2 mod_perl/1.26\r\n
    Last-Modified: Mon, 14 Jun 2004 03:58:23 GMT\r\n
    ETag: "d45b-80-40cd225f"\r\n
    Accept-Ranges: bytes\r\n
    Content-Length: 128\r\n
    Keep-Alive: timeout=15, max=100\r\n
    Connection: Keep-Alive\r\n
    Content-Type: text/html\r\n
    \r\n
Line-based text data: text/html
    <HTML>
     <HEAD>
      <TITLE>www.example-2.com</TITLE>
     </HEAD>
     <BODY>
      <H1 ALIGN="CENTER">www.example-2.com</H1>
     </BODY>
    </HTML>
```

©2005 Nortel Networks Limited. All Rights Reserved

Alteon receives the reply, inserts a cookie that represents the selected real server and forwards the modified packet to the client:

```
Frame 221 (545 bytes on wire, 545 bytes captured)
Ethernet II, Src: 00:60:cf:46:53:60, Dst: 00:08:74:9a:e5:85
Internet Protocol, Src Addr: 192.168.10.150 (192.168.10.150), Dst Addr: 192.168.10.77
(192.168.10.77)
Transmission Control Protocol, Src Port: http (80), Dst Port: 4519 (4519), Seq:
1316551040, Ack: 1398194565, Len: 491
Hypertext Transfer Protocol
    HTTP/1.1 200 OK\r\n
    Set-Cookie: AlteonP=3e45b0283e45aeaa; path=/\r\n
    Date: Fri, 18 Jun 2004 03:39:31 GMT\r\n
    Server: Apache/1.3.23 (Unix)  (Red-Hat/Linux) mod_ssl/2.8.7 OpenSSL/0.9.6b
DAV/1.0.3 PHP/4.1.2 mod_perl/1.26\r\n
    Last-Modified: Mon, 14 Jun 2004 03:58:23 GMT\r\n
    ETag: "d45b-80-40cd225f"\r\n
    Accept-Ranges: bytes\r\n
    Content-Length: 128\r\n
    Keep-Alive: timeout=15, max=100\r\n
    Connection: Keep-Alive\r\n
    Content-Type: text/html\r\n
    \r\n
Line-based text data: text/html
```

Take a look at the Set-Cookie directive:

Set-Cookie: AlteonP=3e45b0283e45aeaa; path=/\r\n

- **AlteonP** is the name of the cookie
- **3e45b0283e45aeaa** is its hex value
- **path=/** is a path to the root directory of the specified URL

The hex cookie value could be converted to human-readable form using the **/info/slb/cookie** *<hex value>* command.

For example, convert the hex cookie above (note the pre-pended 0x before the hex value):

```
>> Main # /info/slb/cookie 0x3e45b0283e45aeaa
Virtual IP address: 192.168.10.150
Real IP address: 192.168.20.20
```

The cookie has a virtual IP address and a real server IP address encoded.

©2005 Nortel Networks Limited. All Rights Reserved

The client receives the response with a cookie, and as it is configured to allow per-session cookies, it inserts that cookie into subsequent HTTP requests when the screen is refreshed:

```
Frame 30 (391 bytes on wire, 391 bytes captured)
Ethernet II, Src: 00:08:74:9a:e5:85, Dst: 00:60:cf:46:53:6e
Internet Protocol, Src Addr: 192.168.10.77 (192.168.10.77), Dst Addr: 192.168.10.150
(192.168.10.150)
Transmission Control Protocol, Src Port: 4521 (4521), Dst Port: http (80), Seq:
1462377828, Ack: 4004838012, Len: 337
Hypertext Transfer Protocol
    GET / HTTP/1.1\r\n
    Accept: */*\r\n
    Accept-Language: en-us\r\n
    Accept-Encoding: gzip, deflate\r\n
    If-Modified-Since: Mon, 14 Jun 2004 03:58:23 GMT\r\n
    If-None-Match: "d45b-80-40cd225f"\r\n
    User-Agent: Mozilla/4.0 (compatible; MSIE 5.5; Windows NT 5.0; T312461; PC SRP
2.0)\r\n
    Host: 192.168.10.150\r\n
    Connection: Keep-Alive\r\n
    Cookie: AlteonP=3e45b0283e45aeaa\r\n
    \r\n
```

Alteon sees the cookie it issued earlier and forwards the request to the same real server:

```
Frame 34 (391 bytes on wire, 391 bytes captured)
Ethernet II, Src: 00:60:cf:46:53:60, Dst: 00:0c:29:63:8f:59
Internet Protocol, Src Addr: 192.168.10.77 (192.168.10.77), Dst Addr: 192.168.20.20
(192.168.20.20)
Transmission Control Protocol, Src Port: 4102 (4102), Dst Port: http (80), Seq:
1462377828, Ack: 3912727578, Len: 337
Hypertext Transfer Protocol
    GET / HTTP/1.1\r\n
    Accept: */*\r\n
    Accept-Language: en-us\r\n
    Accept-Encoding: gzip, deflate\r\n
    If-Modified-Since: Mon, 14 Jun 2004 03:58:23 GMT\r\n
    If-None-Match: "d45b-80-40cd225f"\r\n
    User-Agent: Mozilla/4.0 (compatible; MSIE 5.5; Windows NT 5.0; T312461; PC SRP
2.0)\r\n
    Host: 192.168.10.150\r\n
    Connection: Keep-Alive\r\n
    Cookie: AlteonP=3e45b0283e45aeaa\r\n
    \r\n
```

©2005 Nortel Networks Limited. All Rights Reserved

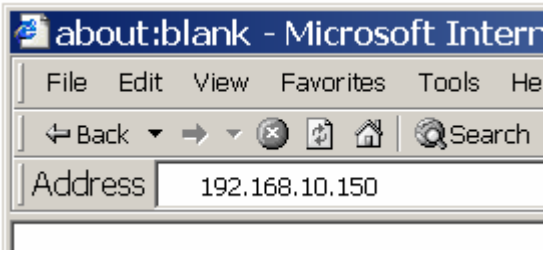
The real server responds with an HTTP reply:

```
Frame 139 (339 bytes on wire, 339 bytes captured)
Ethernet II, Src: 00:0c:29:63:8f:59, Dst: 00:60:cf:46:53:60
Internet Protocol, Src Addr: 192.168.20.20 (192.168.20.20), Dst Addr: 192.168.10.77
(192.168.10.77)
Transmission Control Protocol, Src Port: http (80), Dst Port: 4102 (4102), Seq:
3912727578, Ack: 1462378165, Len: 285
Hypertext Transfer Protocol
    HTTP/1.1 304 Not Modified\r\n
    Date: Fri, 18 Jun 2004 03:43:49 GMT\r\n
    Server: Apache/1.3.23 (Unix)  (Red-Hat/Linux) mod_ssl/2.8.7 OpenSSL/0.9.6b
DAV/1.0.3 PHP/4.1.2 mod_perl/1.26\r\n
    Connection: Keep-Alive\r\n
    Keep-Alive: timeout=15, max=100\r\n
    ETag: "d45b-80-40cd225f"\r\n
    X-Pad: avoid browser bug\r\n
    \r\n
```

Alteon forwards the HTTP reply to the client:

```
Frame 140 (339 bytes on wire, 339 bytes captured)
Ethernet II, Src: 00:60:cf:46:53:60, Dst: 00:08:74:9a:e5:85
Internet Protocol, Src Addr: 192.168.10.150 (192.168.10.150), Dst Addr: 192.168.10.77
(192.168.10.77)
Transmission Control Protocol, Src Port: http (80), Dst Port: 4521 (4521), Seq:
4004838012, Ack: 1462378165, Len: 285
Hypertext Transfer Protocol
    HTTP/1.1 304 Not Modified\r\n
    Date: Fri, 18 Jun 2004 03:43:49 GMT\r\n
    Server: Apache/1.3.23 (Unix)  (Red-Hat/Linux) mod_ssl/2.8.7 OpenSSL/0.9.6b
DAV/1.0.3 PHP/4.1.2 mod_perl/1.26\r\n
    Connection: Keep-Alive\r\n
    Keep-Alive: timeout=15, max=100\r\n
    ETag: "d45b-80-40cd225f"\r\n
    X-Pad: avoid browser bug\r\n
    \r\n
```

7. Close the browser. As we did not configure an expiration time for this example, the cookie is valid until the web browser window is closed.
8. Start traffic capture.
9. Open a new browser window and initiate a connection to virtual server 192.168.10.150 again:



10. While the page loads, check the session table on Alteon. As no cookie was sent in the client's request, Alteon load balanced the session based on the configured round robin metric and split the session to WebServer1 (192.168.20.10):

```
>> Session Table Information# /info/slb/sess/dump
 8,10:  192.168.10.77 4586, 192.168.10.150 80 -> 4106 192.168.20.10 80  age 4  EU
```

©2005 Nortel Networks Limited. All Rights Reserved

11. The WebServer1 page loads:



12. Take a look at the captured traffic.

As with the previous example, at first, Alteon establishes a TCP connection with the client. Once the TCP handshake is established, the client sends an HTTP request. Note that no cookie information is present in the request, as the cookie has expired with the closure of the browser:

```
Frame 215 (518 bytes on wire, 518 bytes captured)
Ethernet II, Src: 00:08:74:9a:e5:85, Dst: 00:60:cf:46:53:6e
Internet Protocol, Src Addr: 192.168.10.77 (192.168.10.77), Dst Addr: 192.168.10.150
(192.168.10.150)
Transmission Control Protocol, Src Port: 4586 (4586), Dst Port: http (80), Seq:
2511985648, Ack: 1994096841, Len: 464
Hypertext Transfer Protocol
    GET / HTTP/1.1\r\n
    Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, application/vnd.ms-
excel, application/vnd.ms-powerpoint, application/msword, application/x-shockwave-
flash, */*\r\n
    Accept-Language: en-us\r\n
    Accept-Encoding: gzip, deflate\r\n
    If-Modified-Since: Mon, 14 Jun 2004 03:58:23 GMT\r\n
    If-None-Match: "d45b-80-40cd225f"\r\n
    User-Agent: Mozilla/4.0 (compatible; MSIE 5.5; Windows NT 5.0; T312461; PC SRP
2.0)\r\n
    Host: 192.168.10.150\r\n
    Connection: Keep-Alive\r\n
    \r\n
```

©2005 Nortel Networks Limited. All Rights Reserved

Upon receiving an HTTP request and with the absence of cookies, Alteon decides which real server to use and splits the session to the selected real server. Once an HTTP reply is received, Alteon inserts a cookie representing this server and forwards the reply to the client:

```
Frame 322 (564 bytes on wire, 564 bytes captured)
Ethernet II, Src: 00:60:cf:46:53:60, Dst: 00:08:74:9a:e5:85
Internet Protocol, Src Addr: 192.168.10.150 (192.168.10.150), Dst Addr: 192.168.10.77
(192.168.10.77)
Transmission Control Protocol, Src Port: http (80), Dst Port: 4586 (4586), Seq:
1994096841, Ack: 2511986112, Len: 510
Hypertext Transfer Protocol
    HTTP/1.1 200 OK\r\n
    Set-Cookie: AlteonP=3e45b0283e45aeb4; path=/\r\n
    Date: Fri, 18 Jun 2004 17:39:40 GMT\r\n
    Server: Apache/1.3.23 (Unix)  (Red-Hat/Linux) mod_ssl/2.8.7 OpenSSL/0.9.6b
DAV/1.0.3 PHP/4.1.2 mod_perl/1.26\r\n
    Last-Modified: Mon, 14 Jun 2004 17:48:50 GMT\r\n
    ETag: "d45c-93-40cde502"\r\n
    Accept-Ranges: bytes\r\n
    Content-Length: 147\r\n
    Keep-Alive: timeout=15, max=100\r\n
    Connection: Keep-Alive\r\n
    Content-Type: text/html\r\n
    \r\n
Line-based text data: text/html
```

Testing with expiration timer set

1. Configure Alteon to include an expiration time for the cookie. For example:

```
>> Main# /cfg/slb/virt 1/service http/pbind
Current persistent binding mode: cookie
Enter clientip|cookie|sslid|disable persistence mode:   cookie
Enter passive|rewrite|insert cookie persistence mode [p/r/i]:   i
Enter insert-cookie expiration as either :
... a date     <MM/dd/yy[@hh:mm]> (e.g. 12/31/01@23:59)
... a duration <days[:hours[:minutes]]> (e.g. 45:30:90)
... or none    <return>
Enter cookie expiration:        6/19/04@00:45
Insert cookie domain name? (y/n) [n]
```

2. Apply the changes:

```
>> Virtual Server 1 http Service# apply
------------------------------------------------------------------
Apply complete; don't forget to "save" updated configuration.
```

3. Start the capture: open a browser window and enter the IP of the virtual server (192.168.10.150).

4. While the page loads, check the session table on Alteon. WebServer2 is selected:

```
>> Session Table Information# /info/slb/sess/dump
 8,26:  192.168.10.77 4702, 192.168.10.150 80 -> 4122 192.168.20.20 80  age 4  EU
```

5. Close the browser, open it again and initiate another HTTP session to the virtual server (192.168.10.150).

6. Check the session table again; the same web server is selected:

```
>> Session Table Information# /info/slb/sess/dump
 8,28:  192.168.10.77 4707, 192.168.10.150 80 -> 4124 192.168.20.20 80  age 4
```

©2005 Nortel Networks Limited. All Rights Reserved

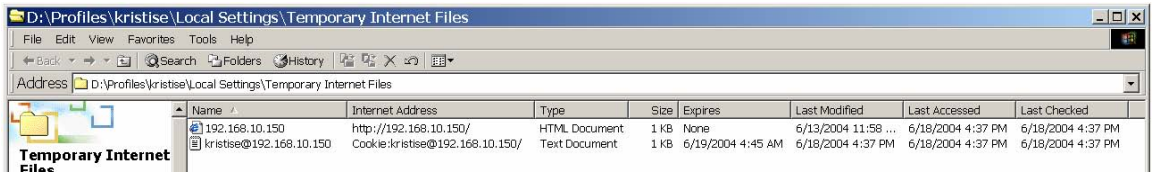
7. Stop the capture and look at the traces.

Virtually the same procedure as in the previous example takes place: client PC and Alteon establish a TCP connection, PC sends HTTP request, Alteon establishes a TCP connection with a real server based on a configured metric, Alteon sends the client's HTTP request, the real server answers with an HTTP reply, and Alteon patches the reply with a cookie that also contains an expiration timer:

```
Frame 235 (545 bytes on wire, 545 bytes captured)
Ethernet II, Src: 00:60:cf:46:53:60, Dst: 00:08:74:9a:e5:85
Internet Protocol, Src Addr: 192.168.10.150 (192.168.10.150), Dst Addr: 192.168.10.77
(192.168.10.77)
Transmission Control Protocol, Src Port: http (80), Dst Port: 4702 (4702), Seq:
1501191461, Ack: 3885857072, Len: 491
Hypertext Transfer Protocol
    HTTP/1.1 200 OK\r\n
    Set-Cookie: AlteonP=3e45b0283e45aeaa; path=/; expires=Sat, 19-Jun-2004 08:45:00
GMT\r\n
    Date: Fri, 18 Jun 2004 06:25:27 GMT\r\n
    Server: Apache/1.3.23 (Unix)  (Red-Hat/Linux) mod_ssl/2.8.7 OpenSSL/0.9.6b
DAV/1.0.3 PHP/4.1.2 mod_perl/1.26\r\n
    Last-Modified: Mon, 14 Jun 2004 03:58:23 GMT\r\n
    ETag: "d45b-80-40cd225f"\r\n
    Accept-Ranges: bytes\r\n
    Content-Length: 128\r\n
    Keep-Alive: timeout=15, max=100\r\n
    Connection: Keep-Alive\r\n
    Content-Type: text/html\r\n
    \r\n
Line-based text data: text/html
```

Note: The **Set-Cookie** directive now contains an expiration date and time in GMT.

The client receives a cookie, checks the expiration timer, and as it's not yet expired stores the cookie locally in the Cookies directory and in the Temporary Internet Files directory on the Windows 2000 system:



The cookie file contains the cookie name, value and any other received attributes:

```
AlteonP3e45b0283e45aeaa192.168.10.150/0303051929629644249543706192296441480*
```

©2005 Nortel Networks Limited. All Rights Reserved

This cookie is then used to send subsequent requests to the server until the cookie expires:

```
Frame 1404 (552 bytes on wire, 552 bytes captured)
Ethernet II, Src: 00:08:74:9a:e5:85, Dst: 00:60:cf:46:53:6e
Internet Protocol, Src Addr: 192.168.10.77 (192.168.10.77), Dst Addr: 192.168.10.150
(192.168.10.150)
Transmission Control Protocol, Src Port: 4707 (4707), Dst Port: http (80), Seq:
3923200410, Ack: 3468971856, Len: 498
Hypertext Transfer Protocol
    GET / HTTP/1.1\r\n
    Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, application/vnd.ms-
excel, application/vnd.ms-powerpoint, application/msword, application/x-shockwave-
flash, */*\r\n
    Accept-Language: en-us\r\n
    Accept-Encoding: gzip, deflate\r\n
    If-Modified-Since: Mon, 14 Jun 2004 03:58:23 GMT\r\n
    If-None-Match: "d45b-80-40cd225f"\r\n
    User-Agent: Mozilla/4.0 (compatible; MSIE 5.5; Windows NT 5.0; T312461; PC SRP
2.0)\r\n
    Host: 192.168.10.150\r\n
    Connection: Keep-Alive\r\n
    Cookie: AlteonP=3e45b0283e45aeaa\r\n
    \r\n
```

©2005 Nortel Networks Limited. All Rights Reserved

Appendix A. Alteon configuration file

```
script start "Alteon AD3" 4  /**** DO NOT EDIT THIS LINE!
/* Configuration dump taken 13:40:51 Thu Jun 17, 2004
/* Version 10.0.30.7,  Base MAC address 00:60:cf:46:53:60
/c/sys
        idle 60
/c/ip/if 1
        ena
        addr 192.168.10.100
/c/ip/if 2
        ena
        addr 192.168.20.200
/c/slb
        on
/c/slb/adv
        direct ena
/c/slb/real 1
        ena
        rip 192.168.20.10
/c/slb/real 2
        ena
        rip 192.168.20.20
/c/slb/group 1
        metric roundrobin
        add 1
        add 2
/c/slb/port 1
        client ena
/c/slb/port 8
        server ena
/c/slb/virt 1
        ena
        vip 192.168.10.150
/c/slb/virt 1/service http
        group 1
        dbind ena
/c/slb/virt 1/service 80/pbind cookie insert
/c/slb/virt 1/service 80/rcount 1
/
script end  /**** DO NOT EDIT THIS LINE!
```

©2005 Nortel Networks Limited. All Rights Reserved

Appendix B. Alteon configuration file with cookie expiration timer

```
script start "Alteon AD3" 4  /**** DO NOT EDIT THIS LINE!
/* Configuration dump taken  8:41:19 Mon Jun 21, 2004
/* Version 10.0.30.7,  Base MAC address 00:60:cf:46:53:60
/c/sys
        idle 60
/c/ip/if 1
        ena
        addr 192.168.10.100
/c/ip/if 2
        ena
        addr 192.168.20.200
/c/slb
        on
/c/slb/adv
        direct ena
/c/slb/real 1
        ena
        rip 192.168.20.10
/c/slb/real 2
        ena
        rip 192.168.20.20
/c/slb/group 1
        metric roundrobin
        add 1
        add 2
/c/slb/port 1
        client ena
/c/slb/port 8
        server ena
/c/slb/virt 1
        ena
        vip 192.168.10.150
/c/slb/virt 1/service http
        group 1
        dbind ena
/c/slb/virt 1/service 80/pbind cookie insert "06/19/04@00:45"
/c/slb/virt 1/service 80/rcount 1
/
script end  /**** DO NOT EDIT THIS LINE!
```

Copyright © 2005 Nortel Networks Limited - All Rights Reserved. Nortel, Nortel Networks, the Nortel logo, Globemark, and Alteon are trademarks of Nortel Networks Limited.

Microsoft, Microsoft Internet Explorer and Windows are trademarks of Microsoft Corporation.

RedHat is a trademark of Red Hat, Inc.

Apache is a trademark of The Apache Software Foundation.

The information in this document is subject to change without notice. The statements, configurations, technical data, and recommendations in this document are believed to be accurate and reliable, but are presented without express or implied warranty. Users must take full responsibility for their applications of any products specified in this document. The information in this document is proprietary to Nortel Networks Limited.

To access more technical documentation, search our knowledge base, or open a service request online, please visit Nortel Networks Technical Support on the web at: http://www.nortel.com/support

If after following this guide you are still having problems, please ensure you have carried out the steps exactly as in this document. If problems still persist, please contact Nortel Networks Technical Support (contact information is available online at: http://www.nortel.com/cgi-bin/comments/comments.cgi?key=techsupport_cu).

We welcome you comments and suggestions on the quality and usefulness of this document. If you would like to leave a feedback please send your comments to: CRALT@nortel.com

©2005 Nortel Networks Limited. All Rights Reserved