Technical Tip

TT-0501403a -- Information -- 18-Jan-2005

Rewrite mode cookie-based persistence

Contents

Contents 1
Introduction 1
Associated Products 1
Overview 1
  Nortel Web and Application switch cookie persistency 3
  Rewrite cookie mode 3
Sample configuration 4
  Setup 4
  Configuring the PC 4
  Configuring WebServer1 6
  Configuring WebServer2 7
  Configuring Alteon 8
    Configuring interfaces 9
    Configuring real servers 9
    Assigning real servers to a group 10
    Configuring virtual server 10
    Enabling client/server processing on ports 11
    Enabling direct access mode 11
    Enabling SLB globally 11
  Testing the configuration 12
Appendix. Alteon configuration file 25

Introduction

This document illustrates the operation and sample configuration of Nortel (formerly Alteon) Web and Application switches' rewrite mode cookie-based persistence.

Associated Products

The information in this document is intended to be used with the following products with the indicated software or hardware revisions:

Product Name or Order Number	Revision Information	
	Potentially Affected	*Corrected*
Nortel (formerly 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 the client's 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),
- an optional max age of the cookie; if max age is not specified in the Set-Cookie header, the user agent deletes the cookie as soon as client closes the web browser,
- an optional path that specifies the subset of URLs that a particular cookie applies to,
- an optional secure attribute that directs the user agent to use only 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 the form of a text file. Note, the location and the 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 following:



Once the cookie is received and, 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, which 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.

The switch can operate in three modes of operation – insert cookie, passive cookie and rewrite cookie. This document illustrates the rewrite cookie mode. For information on passive or insert cookie modes, please consult "Passive mode Cookie-based persistence" or "Insert mode Cookie-based persistence" technical tips.

Rewrite cookie mode

When operating in rewrite cookie mode the Nortel Web or Application switch rewrites the cookie value inserted by the real server and forwards the subsequent request from the client based on that cookie.

When the client sends an initial HTTP request without any cookies in it (Figure 1), the switch makes a forwarding decision based on the configured load balancing metric for the real servers' group.

The real server receives the request, processes it, inserts a cookie and sends an HTTP reply back to client. The switch receives an HTTP reply, replaces the configured number of bytes in the cookie value with a new value representing the initially selected real server and forwards the reply to the client PC. If the client needs to make a subsequent request to the server it inserts the cookie request header into an 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.

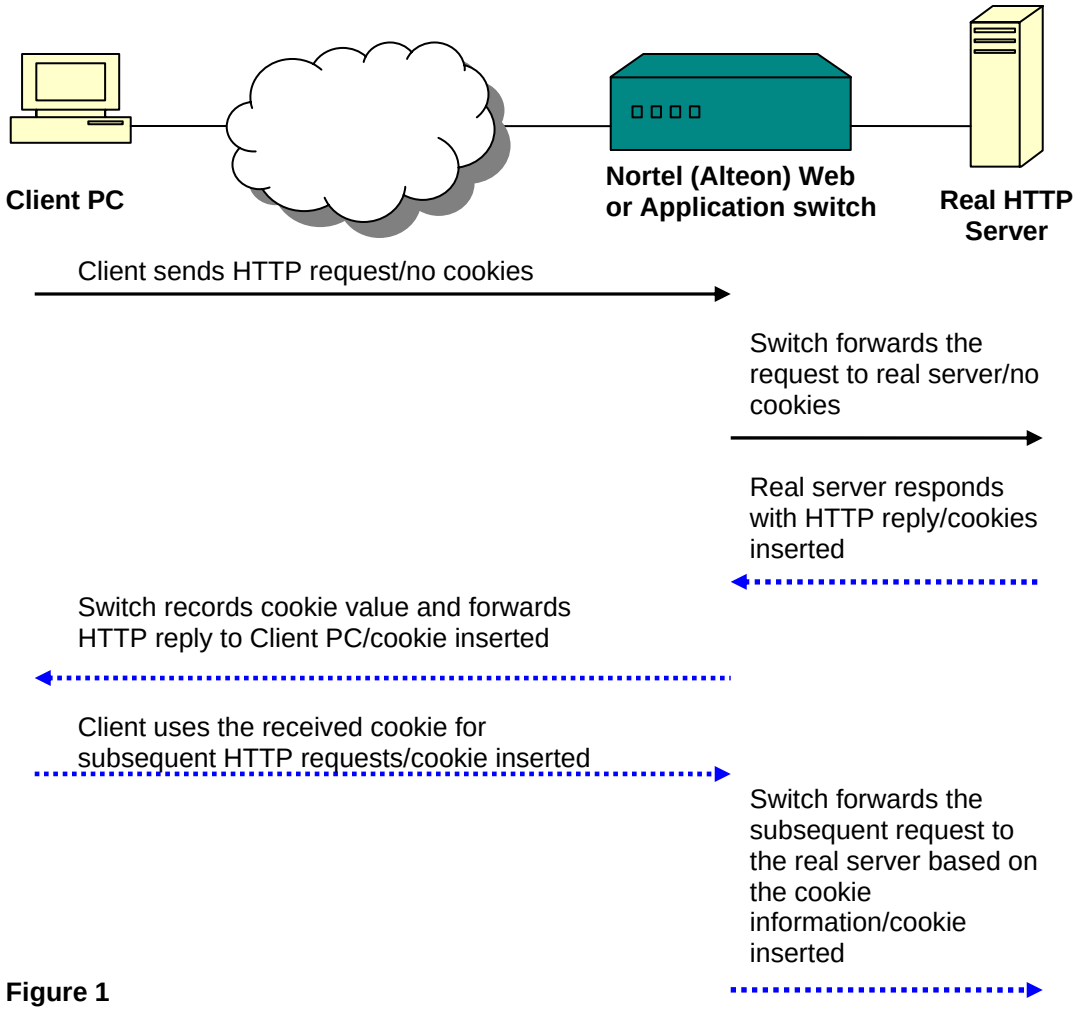
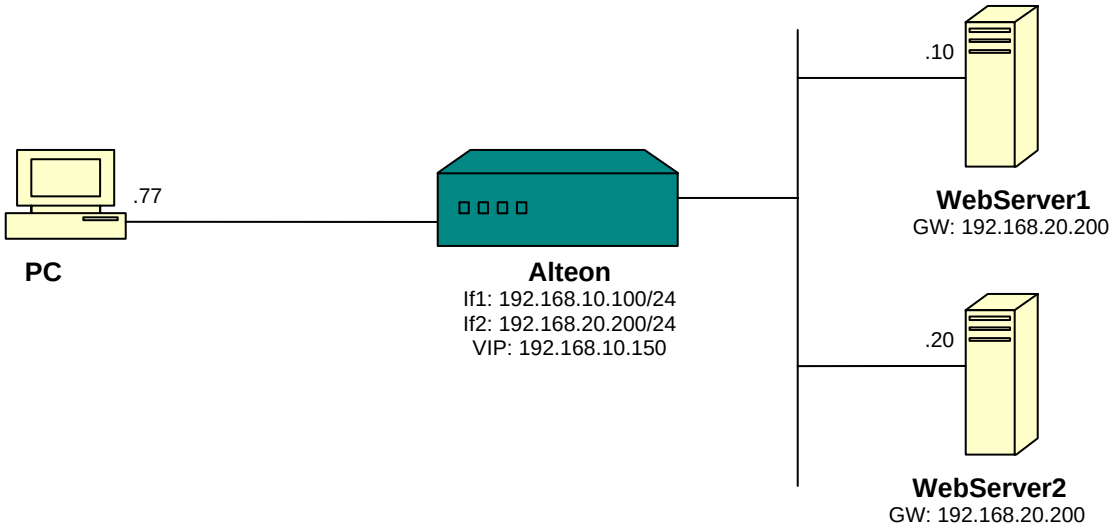


Figure 1

©2005 Nortel Networks Limited. All Rights Reserved

Sample configuration

Setup



PC – Windows 2000 professional with Mozilla 1.6 installed, IP address 192.168.10.77/24;
Alteon – Nortel (Alteon) 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 9 with apache 2.0.49 software installed, IP address 192.168.20.10/24;
WebServer2 – Linux RedHat 9 with apache 2.0.49 software installed, IP address 192.168.20.20/24;

The goal of the configuration is to configure rewrite mode cookie persistency on the Alteon, where the 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:

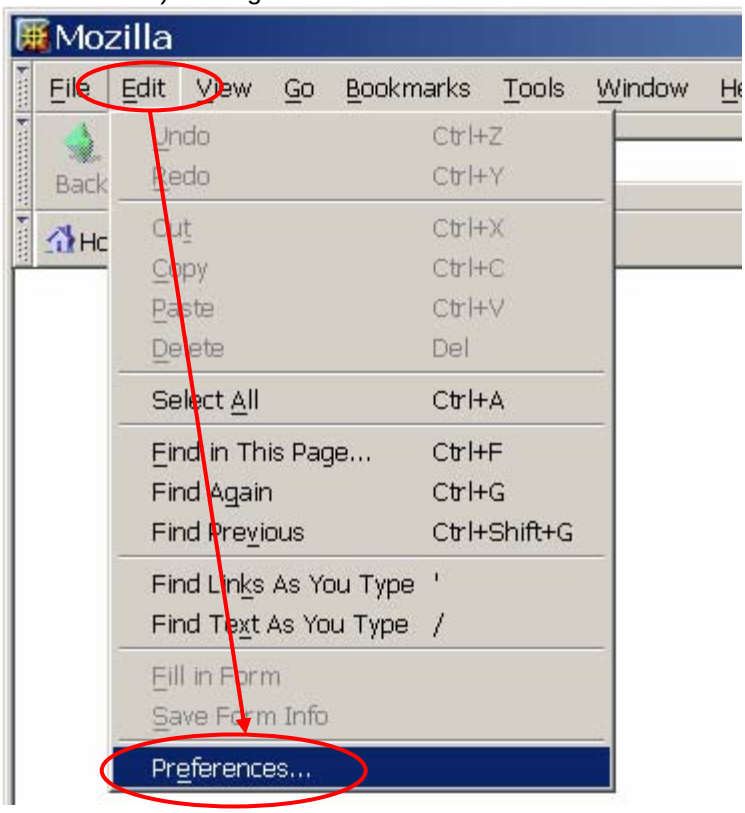
```
C:\>ipconfig

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 . . . . . . . . . :
```

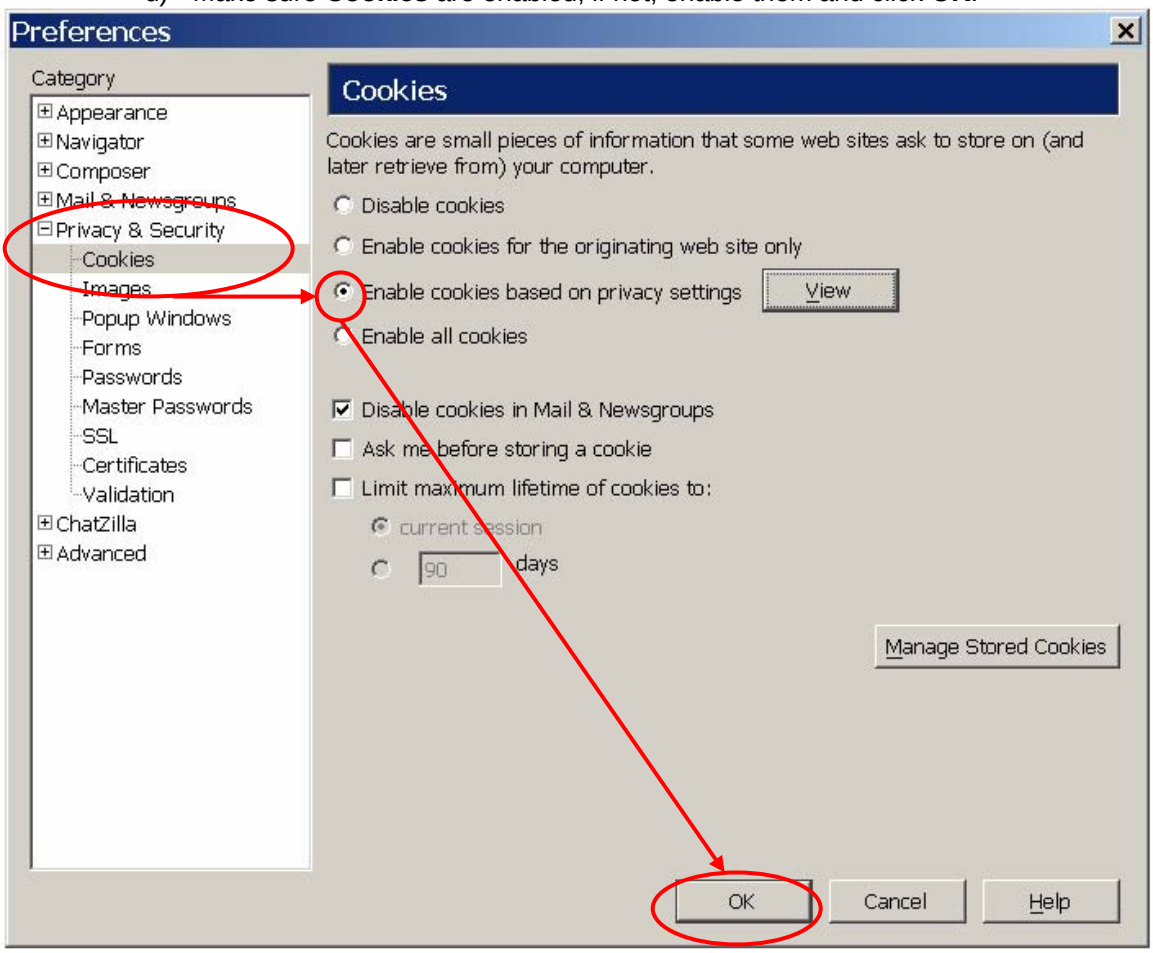
©2005 Nortel Networks Limited. All Rights Reserved

2. Note, it is assumed that the browser is enabled to support cookies. If not, enable cookies.
   a) Navigate **Edit→Preferences** from the browser menu:



©2005 Nortel Networks Limited. All Rights Reserved

b) The **Preferences** window appears.
c) Navigate **Privacy & Security → Cookies** from the tree on the left.
d) Make sure **Cookies** are enabled, if not, enable them and click **OK**:



Configuring WebServer1

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

```
[root@linux9 user]# /sbin/ifconfig eth0 192.168.20.10 netmask 255.255.255.0
```

2. Configure a static route to the PC's network 192.168.10.0/24 via Alteon's interface 2 (192.168.20.200):

```
[root@linux9 user]# /sbin/route add -net 192.168.10.0/24 gw 192.168.20.200
```

3. This configuration assumes that Apache has been compiled with user tracking module (mod_usertrack) enabled.

©2005 Nortel Networks Limited. All Rights Reserved

4. Enable user tracking; to do that add the user tracking directive to the httpd configuration file and turn the tracking on:

```
CookieTracking  on
```

```
[root@linux9 user]# tail /usr/local/apache2/conf/httpd.conf
# The first VirtualHost section is used for requests without a known
# server name.
#
#<VirtualHost *:80>
#    ServerAdmin webmaster@dummy-host.example.com
#    DocumentRoot /www/docs/dummy-host.example.com
#    CustomLog logs/dummy-host.example.com-access log common
#</VirtualHost>

CookieTracking  on
```

Note: By default, when CookieTracking is enabled Apache will add a cookie with cookie name Apache and no expiration timer set.

5. For testing purposes, replace the default index.html.en file in the Apache's root directory (htdocs by default) with a sample html file that will allow us to distinguish between the two web sites, for example:

```
<HTML>
 <HEAD>
  <TITLE>WebServer1</TITLE>
 </HEAD>
 <BODY>
  <H1 ALIGN="CENTER">WebServer1</H1>
 </BODY>
</HTML>
```

6. Start the HTTP server (assuming Apache has been installed with a default /usr/local/apache2 prefix):

```
[root@linux9 user]# /usr/local/apache2/bin/apachectl start
```

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 a static route to the PC's network 192.168.10.0/24 via Alteon's interface 2 (192.168.20.200):

```
[root@linux9 user]# /sbin/route add –net 192.168.10.0/24 gw 192.168.20.200
```

3. This configuration assumes that Apache has been compiled with user tracking module (mod_usertrack) enabled.

©2005 Nortel Networks Limited. All Rights Reserved

4. Enable user tracking; to do that add the user tracking directive to httpd configuration file and turn the tracking on:

```
CookieTracking  on
```

```
[root@linux9 user]# tail /usr/local/apache2/conf/httpd.conf
# The first VirtualHost section is used for requests without a known
# server name.
#
#<VirtualHost *:80>
#    ServerAdmin webmaster@dummy-host.example.com
#    DocumentRoot /www/docs/dummy-host.example.com
#    CustomLog logs/dummy-host.example.com-access log common
#</VirtualHost>

CookieTracking  on
```

Note: By default, when CookieTracking is enabled Apache will add a cookie with cookie name Apache and no expiration timer set.

5. For testing purposes, replace the default index.html.en file in the Apache's root directory (htdocs by default) with a sample html file that will allow us to distinguish between the two web sites, for example:

```
<HTML>
 <HEAD>
  <TITLE>WebServer2</TITLE>
 </HEAD>
 <BODY>
  <H1 ALIGN="CENTER">WebServer2</H1>
 </BODY>
</HTML>
```

6. Start the HTTP server (assuming Apache has been installed with a default /usr/local/apache2 prefix):

```
[root@linux9 user]# /usr/local/apache2/bin/apachectl start
```

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
```

©2005 Nortel Networks Limited. All Rights Reserved

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
```

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 real server 2 configuration menu, set 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
```

©2005 Nortel Networks Limited. All Rights Reserved

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 the 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
```

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 virtual server

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

©2005 Nortel Networks Limited. All Rights Reserved

4. Enable persistent binding using /**cfg/slb/virt <*virtual server number*> /service <*serivice*>/pbind** command.

    a) When prompted enter **cookie** as a persistence mode;
    b) Enter **r** to select rewrite mode of operation;
    c) Enter the cookie name to look for in the real server's reply; as we didn't change the default cookie name for the Apache real server, **Apache** is used as a name;
    d) Enter the number of bytes to extract (**16** in this example); this specifies how many bytes starting from the first byte of the cookie value string should be rewritten by Alteon. The possible choices are 8 or 16. When 8 is selected only the real server IP is encoded in cookie, if 16 is selected both the real server IP and the virtual server IP is encoded in cookie;
    e) **Disable** the **Look for cookie in URI parameter**, as there is no need to enable cookie search in URI because the cookie will not be embedded in the URL for this configuration:

```
>> 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]:   r
Enter Cookie Name:      Apache
Enter number of bytes to extract [8,16]: 16
Look for cookie in URI [e|d]:   d
```

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 be disabled.

Enabling SLB globally

1. Enable SLB globally:

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

©2005 Nortel Networks Limited. All Rights Reserved

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.
```

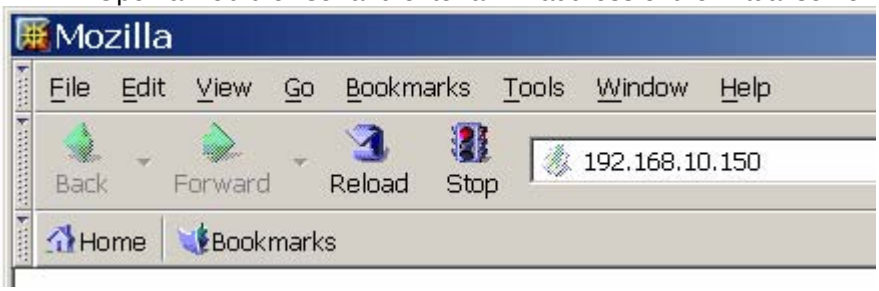
Note as soon as changes are applied, the following messages appear on the console indicating the real servers and virtual server are up and operational:

```
Jun 25 10:00:23 NOTICE  slb: real service 192.168.20.10:80 operational
Jun 25 10:00:23 NOTICE  slb: real server 192.168.20.10 operational
Jun 25 10:00:23 NOTICE  slb: Services are available for Virtual Server
1:192.168.10.150
Jun 25 10:00:24 NOTICE  slb: real service 192.168.20.20:80 operational
Jun 25 10:00:24 NOTICE  slb: real server 192.168.20.20 operational
```

Testing the configuration

1. Start traffic capture on the PC using, for example, Ethereal; start traffic capture on the WebServers using, for example, tcpdump.

2. Open a web browser and enter an IP address of the virtual server (192.168.10.150):



3. While the page loads check the session table on Alteon; Alteon has selected WebServer1 (192.168.20.10) to process the request:

```
>> Session Table Information# /info/slb/sess/dump
 8,10:  192.168.10.77 3498, 192.168.10.150 80 -> 4106 192.168.20.10 80  age 10 E
```

©2005 Nortel Networks Limited. All Rights Reserved

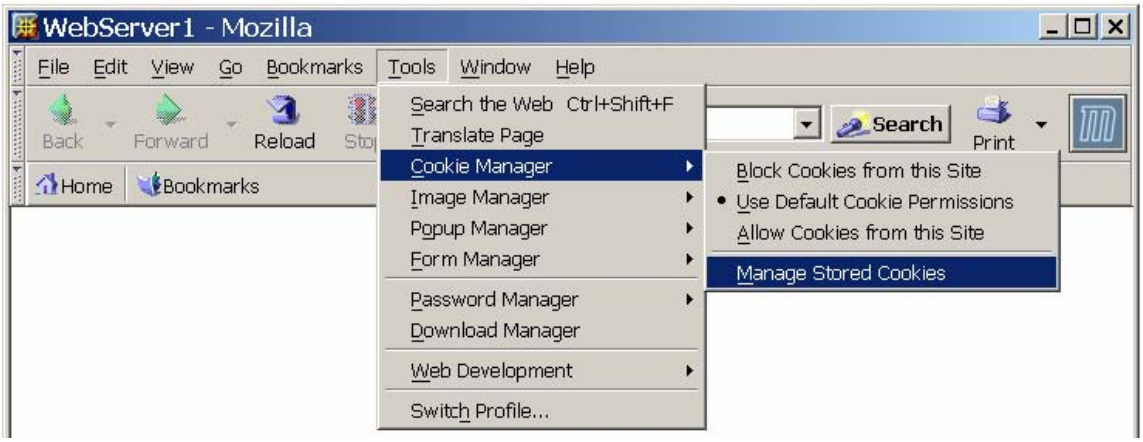
4. WebServer1 page loads:



5. Refresh a page and check the session table again:

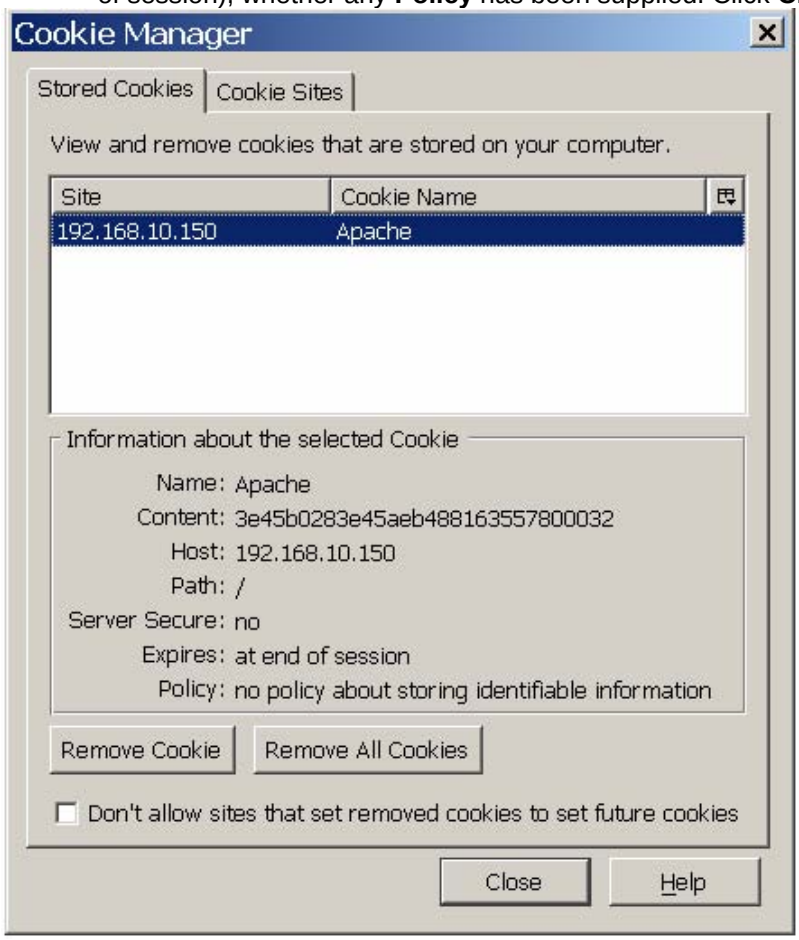
```
>> Session Table Information# /info/slb/sess/dump
 8,12:  192.168.10.77 3499, 192.168.10.150 80 -> 4108 192.168.20.10 80  age 10 E
```

6. Check the temporary stored cookie. Navigate **Tools→Cookie Manager→Manage Stored Cookies** from the browser menu:



©2005 Nortel Networks Limited. All Rights Reserved

7. The **Cookie Manager** window appears. **Note**: all the cookies were cleared prior to the test so it would be easier to find cookie inserted by virtual server.
   Note the presence of a cookie received from virtual server.
8. Check the cookie information – **Name** (Apache), **Content** (cookie value), **Host** (virtual server IP), **Path** (path to the root directory /), whether **Secure** connection has been advised (no), when the cookie **Expires** (at end of session), whether any **Policy** has been supplied. Click **Close**:



©2005 Nortel Networks Limited. All Rights Reserved

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

The client PC (192.168.10.77) initiates an HTTP session by sending TCP SYN to virtual server IP (192.168.10.150):

```
Frame 134 (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: 3498 (3498), Dst Port: http (80), Seq:
2074629387, Ack: 0, Len: 0
    Source port: 3498 (3498)
    Destination port: http (80)
    Sequence number: 2074629387
    Header length: 28 bytes
    Flags: 0x0002 (SYN)
    Window size: 65535
    Checksum: 0x0e3e (correct)
    Options: (8 bytes)
```

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

```
Frame 135 (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: 3498 (3498), Seq:
4102798857, Ack: 2074629388, Len: 0
    Source port: http (80)
    Destination port: 3498 (3498)
    Sequence number: 4102798857
    Acknowledgement number: 2074629388
    Header length: 24 bytes
    Flags: 0x0012 (SYN, ACK)
    Window size: 65535
    Checksum: 0x709e (correct)
    Options: (4 bytes)
```

The client acknowledges the SYN ACK with an ACK:

```
Frame 136 (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: 3498 (3498), Dst Port: http (80), Seq:
2074629388, Ack: 4102798858, Len: 0
    Source port: 3498 (3498)
    Destination port: http (80)
    Sequence number: 2074629388
    Acknowledgement number: 4102798858
    Header length: 20 bytes
    Flags: 0x0010 (ACK)
    Window size: 65535
    Checksum: 0x885b (correct)
```

©2005 Nortel Networks Limited. All Rights Reserved

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

```
Frame 137 (571 bytes on wire, 571 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: 3498 (3498), Dst Port: http (80), Seq:
2074629388, Ack: 4102798858, Len: 517
Hypertext Transfer Protocol
    GET / HTTP/1.1\r\n
    Host: 192.168.10.150\r\n
    User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.0; en-US; rv:1.6)
Gecko/20040113\r\n
    Accept:
text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,image/
png,image/jpeg,image/gif;q=0.2,*/*;q=0.1\r\n
    Accept-Language: en-us,en;q=0.5\r\n
    Accept-Encoding: gzip,deflate\r\n
    Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7\r\n
    Keep-Alive: 300\r\n
    Connection: keep-alive\r\n
    If-Modified-Since: Fri, 25 Jun 2004 11:56:16 GMT\r\n
    If-None-Match: "321bd-6b-f027800;3214f-961-8562af00"\r\n
    \r\n
```

Once an 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 based on the configured for the group metric (WebServer1 (192.168.20.10) was selected in this example).
Once the real server is selected, Alteon starts a TCP handshake with a real server on behalf of the client by sending a TCP SYN:

```
Frame 138 (64 bytes on wire, 64 bytes captured)
Ethernet II, Src: 00:08:74:9a:e5:85, Dst: 00:0c:29:e1:6b:93
Internet Protocol, Src Addr: 192.168.10.77 (192.168.10.77), Dst Addr: 192.168.20.10
(192.168.20.10)
Transmission Control Protocol, Src Port: 4106 (4106), Dst Port: http (80), Seq:
2074629387, Ack: 0, Len: 0
    Source port: 4106 (4106)
    Destination port: http (80)
    Sequence number: 2074629387
    Header length: 24 bytes
    Flags: 0x0002 (SYN)
    Window size: 65535
    Checksum: 0x1771 (correct)
    Options: (4 bytes)
```

©2005 Nortel Networks Limited. All Rights Reserved

A real server (192.168.20.10) responds with a SYN ACK:

```
Frame 139 (58 bytes on wire, 58 bytes captured)
Ethernet II, Src: 00:0c:29:e1:6b:93, Dst: 00:60:cf:46:53:60
Internet Protocol, Src Addr: 192.168.20.10 (192.168.20.10), Dst Addr: 192.168.10.77
(192.168.10.77)
Transmission Control Protocol, Src Port: http (80), Dst Port: 4106 (4106), Seq:
2643458766, Ack: 2074629388, Len: 0
    Source port: http (80)
    Destination port: 4106 (4106)
    Sequence number: 2643458766
    Acknowledgement number: 2074629388
    Header length: 24 bytes
    Flags: 0x0012 (SYN, ACK)
    Window size: 5840
    Checksum: 0x6831 (correct)
    Options: (4 bytes)
```

Alteon acknowledges it with an ACK:

```
Frame 140 (60 bytes on wire, 60 bytes captured)
Ethernet II, Src: 00:60:cf:46:53:60, Dst: 00:0c:29:e1:6b:93
Internet Protocol, Src Addr: 192.168.10.77 (192.168.10.77), Dst Addr: 192.168.20.10
(192.168.20.10)
Transmission Control Protocol, Src Port: 4106 (4106), Dst Port: http (80), Seq:
2074629388, Ack: 2643458767, Len: 0
    Source port: 4106 (4106)
    Destination port: http (80)
    Sequence number: 2074629388
    Acknowledgement number: 2643458767
    Header length: 24 bytes
    Flags: 0x0010 (ACK)
    Window size: 65535
    Checksum: 0x7f02 (correct)
    Options: (4 bytes)
```

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

```
Frame 141 (571 bytes on wire, 571 bytes captured)
Ethernet II, Src: 00:60:cf:46:53:60, Dst: 00:0c:29:e1:6b:93
Internet Protocol, Src Addr: 192.168.10.77 (192.168.10.77), Dst Addr: 192.168.20.10
(192.168.20.10)
Transmission Control Protocol, Src Port: 4106 (4106), Dst Port: http (80), Seq:
2074629388, Ack: 2643458767, Len: 517
Hypertext Transfer Protocol
    GET / HTTP/1.1\r\n
    Host: 192.168.10.150\r\n
    User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.0; en-US; rv:1.6)
Gecko/20040113\r\n
    Accept:
text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,image/
png,image/jpeg,image/gif;q=0.2,*/*;q=0.1\r\n
    Accept-Language: en-us,en;q=0.5\r\n
    Accept-Encoding: gzip,deflate\r\n
    Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7\r\n
    Keep-Alive: 300\r\n
    Connection: keep-alive\r\n
    If-Modified-Since: Fri, 25 Jun 2004 11:56:16 GMT\r\n
    If-None-Match: "321bd-6b-f027800;3214f-961-8562af00"\r\n
    \r\n
```

©2005 Nortel Networks Limited. All Rights Reserved

Real server acknowledges the request, process the request, inserts a cookie into the HTTP response and replies to the client:

```
Frame 154 (660 bytes on wire, 660 bytes captured)
Ethernet II, Src: 00:0c:29:e1:6b:93, Dst: 00:60:cf:46:53:60
Internet Protocol, Src Addr: 192.168.20.10 (192.168.20.10), Dst Addr: 192.168.10.77
(192.168.10.77)
Transmission Control Protocol, Src Port: http (80), Dst Port: 4106 (4106), Seq:
2643458767, Ack: 2074629905, Len: 606
Hypertext Transfer Protocol
    HTTP/1.1 200 OK\r\n
    Date: Fri, 25 Jun 2004 11:39:17 GMT\r\n
    Server: Apache/2.0.49 (Unix)\r\n
    Content-Location: index.html.en\r\n
    Vary: negotiate,accept-language,accept-charset\r\n
    TCN: choice\r\n
    Set-Cookie: Apache=192.168.10.77.1088163557800032; path=/\r\n
    Last-Modified: Fri, 25 Jun 2004 11:09:14 GMT\r\n
    ETag: "321bd-6c-66ce2a80;3214f-961-8562af00"\r\n
    Accept-Ranges: bytes\r\n
    Content-Length: 108\r\n
    Keep-Alive: timeout=15, max=100\r\n
    Connection: Keep-Alive\r\n
    Content-Type: text/html; charset=ISO-8859-1\r\n
    Content-Language: en\r\n
    \r\n
Line-based text data: text/html
```

Alteon receives an HTTP reply on its server-enabled port, looks up the cookie with a name of “Apache” (specified during configuration), replaces the first 16 bytes of the real server cookie with a hash value representing real server, and forwards the request to the client PC:

```
Frame 155 (660 bytes on wire, 660 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: 3498 (3498), Seq:
4102798858, Ack: 2074629905, Len: 606
Hypertext Transfer Protocol
    HTTP/1.1 200 OK\r\n
    Date: Fri, 25 Jun 2004 11:39:17 GMT\r\n
    Server: Apache/2.0.49 (Unix)\r\n
    Content-Location: index.html.en\r\n
    Vary: negotiate,accept-language,accept-charset\r\n
    TCN: choice\r\n
    Set-Cookie: Apache=3e45b0283e45aeb488163557800032; path=/\r\n
    Last-Modified: Fri, 25 Jun 2004 11:09:14 GMT\r\n
    ETag: "321bd-6c-66ce2a80;3214f-961-8562af00"\r\n
    Accept-Ranges: bytes\r\n
    Content-Length: 108\r\n
    Keep-Alive: timeout=15, max=100\r\n
    Connection: Keep-Alive\r\n
    Content-Type: text/html; charset=ISO-8859-1\r\n
    Content-Language: en\r\n
    \r\n
Line-based text data: text/html
```

©2005 Nortel Networks Limited. All Rights Reserved

Take a look at the **Set-Cookie** directive:

```
Set-Cookie: Apache=3e45b0283e45aeb488163557800032; path=/\r\n
```

The first 16 bytes (as per our configuration) were rewritten with a new value. The rewritten value could be converted to humanly readable form using **/info/slb/cookie** <cookie value> command (note the leading 0x in front of the cookie value):

```
>> Main# /info/slb/cookie 0x3e45b0283e45aeb4
Virtual IP address: 192.168.10.150
Real IP address: 192.168.20.10
```

The value represents the real server and virtual server IP addresses.

Note: If during configuration 8 bytes were selected to be rewritten, only the real IP address would have been encoded in the Alteon cookie value.

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

```
Frame 1063 (557 bytes on wire, 557 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: 3499 (3499), Dst Port: http (80), Seq:
2101627949, Ack: 2425496755, Len: 503
Hypertext Transfer Protocol
    GET / HTTP/1.1\r\n
    Host: 192.168.10.150\r\n
    User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.0; en-US; rv:1.6)
Gecko/20040113\r\n
    Accept:
text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,image/
png,image/jpeg,image/gif;q=0.2,*/*;q=0.1\r\n
    Accept-Language: en-us,en;q=0.5\r\n
    Accept-Encoding: gzip,deflate\r\n
    Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7\r\n
    Keep-Alive: 300\r\n
    Connection: keep-alive\r\n
    Cookie: Apache=3e45b0283e45aeb488163557800032\r\n
    Pragma: no-cache\r\n
    Cache-Control: no-cache\r\n
    \r\n
```

©2005 Nortel Networks Limited. All Rights Reserved

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

```
Frame 1067 (557 bytes on wire, 557 bytes captured)
Ethernet II, Src: 00:60:cf:46:53:60, Dst: 00:0c:29:e1:6b:93
Internet Protocol, Src Addr: 192.168.10.77 (192.168.10.77), Dst Addr: 192.168.20.10
(192.168.20.10)
Transmission Control Protocol, Src Port: 4108 (4108), Dst Port: http (80), Seq:
2101627949, Ack: 2780859594, Len: 503
Hypertext Transfer Protocol
    GET / HTTP/1.1\r\n
    Host: 192.168.10.150\r\n
    User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.0; en-US; rv:1.6)
Gecko/20040113\r\n
    Accept:
text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,image/
png,image/jpeg,image/gif;q=0.2,*/*;q=0.1\r\n
    Accept-Language: en-us,en;q=0.5\r\n
    Accept-Encoding: gzip,deflate\r\n
    Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7\r\n
    Keep-Alive: 300\r\n
    Connection: keep-alive\r\n
    Cookie: Apache=3e45b0283e45aeb488163557800032\r\n
    Pragma: no-cache\r\n
    Cache-Control: no-cache\r\n
    \r\n
```

The real server responds with an HTTP reply:

```
Frame 1070 (601 bytes on wire, 601 bytes captured)
Ethernet II, Src: 00:0c:29:e1:6b:93, Dst: 00:60:cf:46:53:60
Internet Protocol, Src Addr: 192.168.20.10 (192.168.20.10), Dst Addr: 192.168.10.77
(192.168.10.77)
Transmission Control Protocol, Src Port: http (80), Dst Port: 4108 (4108), Seq:
2780859594, Ack: 2101628452, Len: 547
Hypertext Transfer Protocol
    HTTP/1.1 200 OK\r\n
    Date: Fri, 25 Jun 2004 11:41:07 GMT\r\n
    Server: Apache/2.0.49 (Unix)\r\n
    Content-Location: index.html.en\r\n
    Vary: negotiate,accept-language,accept-charset\r\n
    TCN: choice\r\n
    Last-Modified: Fri, 25 Jun 2004 11:09:14 GMT\r\n
    ETag: "321bd-6c-66ce2a80;3214f-961-8562af00"\r\n
    Accept-Ranges: bytes\r\n
    Content-Length: 108\r\n
    Keep-Alive: timeout=15, max=100\r\n
    Connection: Keep-Alive\r\n
    Content-Type: text/html; charset=ISO-8859-1\r\n
    Content-Language: en\r\n
    \r\n
Line-based text data: text/html
```

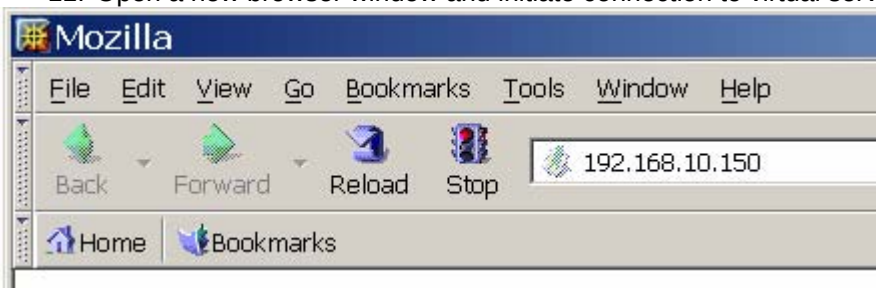
©2005 Nortel Networks Limited. All Rights Reserved

Alteon forwards the reply to the PC:

```
Frame 1071 (601 bytes on wire, 601 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: 3499 (3499), Seq:
2425496755, Ack: 2101628452, Len: 547
Hypertext Transfer Protocol
    HTTP/1.1 200 OK\r\n
    Date: Fri, 25 Jun 2004 11:41:07 GMT\r\n
    Server: Apache/2.0.49 (Unix)\r\n
    Content-Location: index.html.en\r\n
    Vary: negotiate,accept-language,accept-charset\r\n
    TCN: choice\r\n
    Last-Modified: Fri, 25 Jun 2004 11:09:14 GMT\r\n
    ETag: "321bd-6c-66ce2a80;3214f-961-8562af00"\r\n
    Accept-Ranges: bytes\r\n
    Content-Length: 108\r\n
    Keep-Alive: timeout=15, max=100\r\n
    Connection: Keep-Alive\r\n
    Content-Type: text/html; charset=ISO-8859-1\r\n
    Content-Language: en\r\n
    \r\n
Line-based text data: text/html
```

10. As we did not configure any expiration time for the cookies on the apache servers, the default “None” expiration time is applied, therefore the cookie expires with a web browser closure.
    Close the web browser. Start traffic capture.

11. Open a new browser window and initiate connection to virtual server 192.168.10.150 again:



12. While the page loads check the session table on Alteon. As round robin has been configured as the load balancing algorithm and as the cookie has expired, the client request has been forwarded to the next available real server – WebServer2 (192.168.20.20):

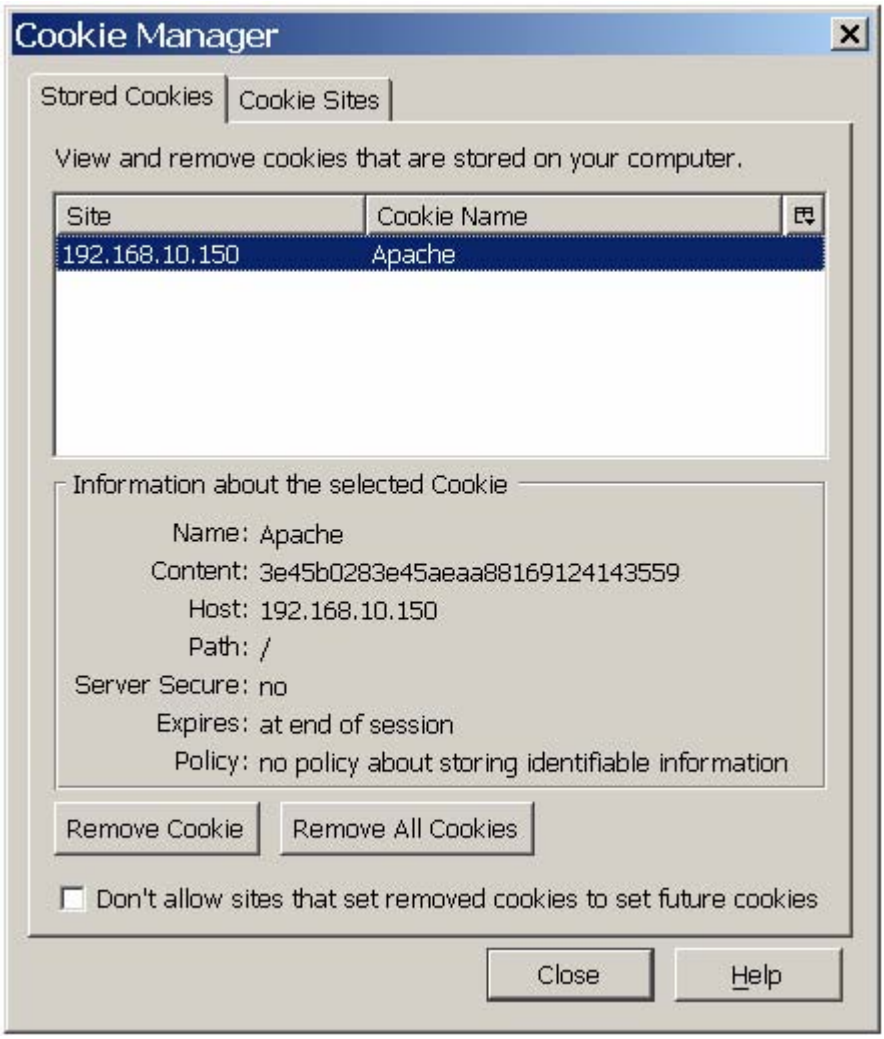
```
>> Session Table Information# /info/slb/sess/dump
 8,18:  192.168.10.77 3545, 192.168.10.150 80 -> 4114 192.168.20.20 80  age 10 E
```

©2005 Nortel Networks Limited. All Rights Reserved

13. The WebServer2 page loads:



14. Take a look at the new cookie by navigating **Tools→Cookie Manager→Manage Stored Cookies** from the browser menu:



©2005 Nortel Networks Limited. All Rights Reserved

15. Take a look at the captured traffic.

As with the previous example, at first, Alteon establishes a TCP connection with a client. Once a 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 227 (572 bytes on wire, 572 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: 3545 (3545), Dst Port: http (80), Seq:
2781057722, Ack: 2457365426, Len: 518
Hypertext Transfer Protocol
    GET / HTTP/1.1\r\n
    Host: 192.168.10.150\r\n
    User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.0; en-US; rv:1.6)
Gecko/20040113\r\n
    Accept:
text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,image/
png,image/jpeg,image/gif;q=0.2,*/*;q=0.1\r\n
    Accept-Language: en-us,en;q=0.5\r\n
    Accept-Encoding: gzip,deflate\r\n
    Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7\r\n
    Keep-Alive: 300\r\n
    Connection: keep-alive\r\n
    If-Modified-Since: Fri, 25 Jun 2004 11:09:14 GMT\r\n
    If-None-Match: "321bd-6c-66ce2a80;3214f-961-8562af00"\r\n
    \r\n
```

Upon receiving an HTTP request and with the absence of cookies, Alteon decides on the real server to use and splits the session to the selected real server (real server 2 – 192.168.20.20 was selected in this case). The real server processes the request and replies with an HTTP reply that has a cookie inserted:

```
Frame 234 (658 bytes on wire, 658 bytes captured)
Ethernet II, Src: 00:0c:29:a5:bc:19, 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: 4114 (4114), Seq:
4229609406, Ack: 2781058240, Len: 604
Hypertext Transfer Protocol
    HTTP/1.1 200 OK\r\n
    Date: Fri, 25 Jun 2004 13:12:04 GMT\r\n
    Server: Apache/2.0.49 (Unix)\r\n
    Content-Location: index.html.en\r\n
    Vary: negotiate,accept-language,accept-charset\r\n
    TCN: choice\r\n
    Set-Cookie: Apache=192.168.10.77.1088169124143559; path=/\r\n
    Last-Modified: Fri, 25 Jun 2004 11:56:16 GMT\r\n
    ETag: "321bd-6b-f027800;3214f-961-8562af00"\r\n
    Accept-Ranges: bytes\r\n
    Content-Length: 107\r\n
    Keep-Alive: timeout=15, max=100\r\n
    Connection: Keep-Alive\r\n
    Content-Type: text/html; charset=ISO-8859-1\r\n
    Content-Language: en\r\n
    \r\n
Line-based text data: text/html
```

©2005 Nortel Networks Limited. All Rights Reserved

Alteon receives a reply on its server port, replaces the first 16 bytes (per configuration) of the cookie value with its own cookie value and forwards the reply to the client:

```
Frame 235 (658 bytes on wire, 658 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: 3545 (3545), Seq:
2457365426, Ack: 2781058240, Len: 604
Hypertext Transfer Protocol
    HTTP/1.1 200 OK\r\n
    Date: Fri, 25 Jun 2004 13:12:04 GMT\r\n
    Server: Apache/2.0.49 (Unix)\r\n
    Content-Location: index.html.en\r\n
    Vary: negotiate,accept-language,accept-charset\r\n
    TCN: choice\r\n
    Set-Cookie: Apache=3e45b0283e45aeaa88169124143559; path=/\r\n
    Last-Modified: Fri, 25 Jun 2004 11:56:16 GMT\r\n
    ETag: "321bd-6b-f027800;3214f-961-8562af00"\r\n
    Accept-Ranges: bytes\r\n
    Content-Length: 107\r\n
    Keep-Alive: timeout=15, max=100\r\n
    Connection: Keep-Alive\r\n
    Content-Type: text/html; charset=ISO-8859-1\r\n
    Content-Language: en\r\n
    \r\n
Line-based text data: text/html
```

16. Look up a new Alteon cookie value:

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

©2005 Nortel Networks Limited. All Rights Reserved

Appendix. Alteon configuration file

```
script start "Alteon AD3" 4  /**** DO NOT EDIT THIS LINE!
/* Configuration dump taken 11:09:53 Fri Jun 25, 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 rewrite Apache 1 16 disable
/c/slb/virt 1/service 80/rcount 1
/
script end  /**** DO NOT EDIT THIS LINE!
```

©2005 Nortel Networks Limited. All Rights Reserved

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

Microsoft and Windows are trademarks of Microsoft Corporation.

RedHat is a trademark of Red Hat, Inc.

Apache is a trademark of The Apache Software Foundation.

Mozilla is a trademark of Netscape Communications Corporation.

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