

Technical Tip

TT-0411405a -- Information -- 24-Nov-2004



Alteon Browser-Smart Load Balancing

Contents:

Introduction:	1
Associated Products:	1
Overview	1
Sample Configuration	3
Setup	3
Configuring PC1	4
Configuring PC2	4
Configuring WebServer1	4
Configuring WebServer2	5
Configuring Alteon	5
Configuring interfaces	6
Defining strings to be used for Smart Browser load balancing	6
Configuring real servers	7
Assigning real servers to a group	8
Creating virtual server	8
Enabling client/server processing on ports	9
Enabling direct access mode	9
Enabling SLB globally	9
Testing the configuration	9
Appendix A. Alteon configuration file	14

Introduction:

This technical tip illustrates the operation and sample configuration of browser-smart balancing on Alteon 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:

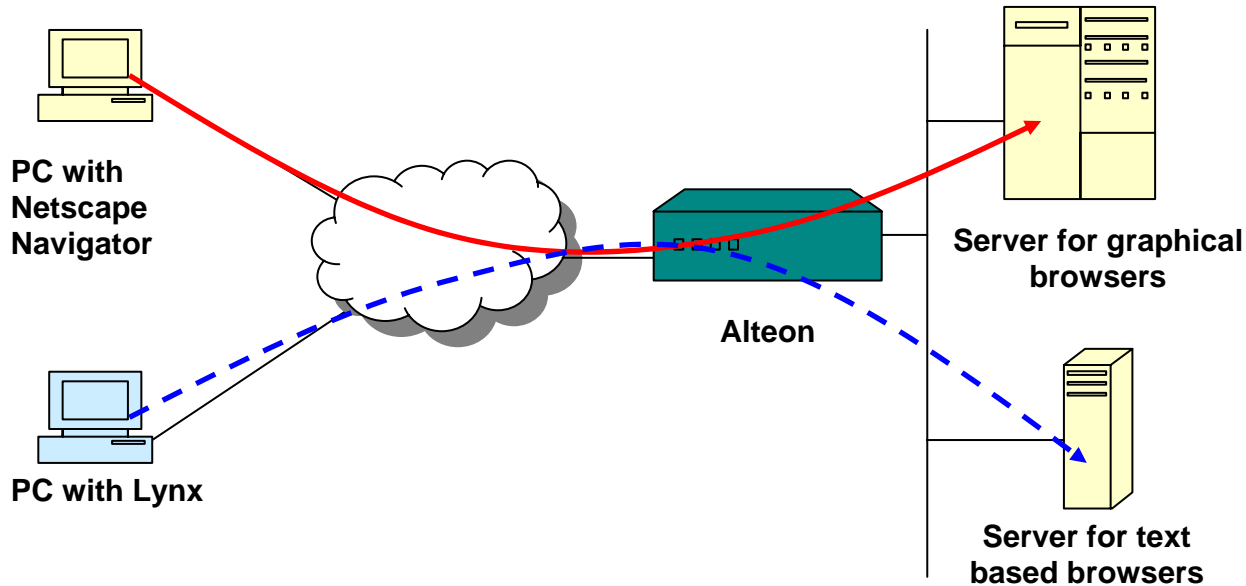
Product Name or Order Number	Revision Information	
	Potentially Affected	Corrected
Alteon WebOS and Application switches: 180e, 180 Plus, 184, AD2, AD3, AD4, 2224, 2424, 2208, 2216, 3408	All	N/A

Overview

With a wide variety of different web browsers available nowadays and since each web browser implementation differs from one another (different page rendering, and different features supported) it is desirable to be able to detect users web browser type so that HTTP requests can be served appropriately.

Alteon switch smart browser load balancing feature allows administrators to load balance the traffic based on the browser type of the client. For example, clients who use text based browser like Lynx can be handled by one server and clients who use a graphical browser like Netscape Navigator or Microsoft Internet Explorer can be handled by a different server see the following figure 1.

Figure 1



The smart browser feature when enabled forces Alteon to look into an HTTP request and examine the User-Agent field. Upon examination Alteon determines whether browser type provided in the request matches any configured types (defined as URL strings), looks up the real server configured to handle the requests from a particular browser and forwards the request to the appropriate real server.

The User-Agent field starts off with a name of the product (the browser) and a version number separated with a slash character (/). For example below is a sample User-Agent field sent by Lynx 2.8.5 text based web browser:

```
User-Agent: Lynx/2.8.5dev.7 libwww-FM/2.14 SSL-MM/1.4.1 OpenSSL/0.9.7
```

The product and version information is sometimes followed by a comment in parentheses, which usually indicates the operating system. For example, the following is a sample User-Agent field sent by Microsoft Internet Explorer 5.5 installed on Windows 2000:

```
User-Agent: Mozilla/4.0 (compatible; MSIE 5.5; Windows NT 5.0)
```

The following is a sample User-Agent field sent by Mozilla 1.6 browser installed on Windows 2000:

```
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.0; en-US; rv:1.6) Gecko/20040113
```

Next is a sample User-Agent field sent by Opera 7.51 browser installed on Windows 2000:

```
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.0) Opera 7.51 [en]
```

It is not a trivial task to write the appropriate string to be matched against User-Agent string. All the sections of the field must be taken into consideration. Specifying only the product type as a string can be useful to distinguish the text-only browsers from the graphical browsers, as there are not many text-only browsers, and most likely the majority of the users with text browsers use a Lynx browser.

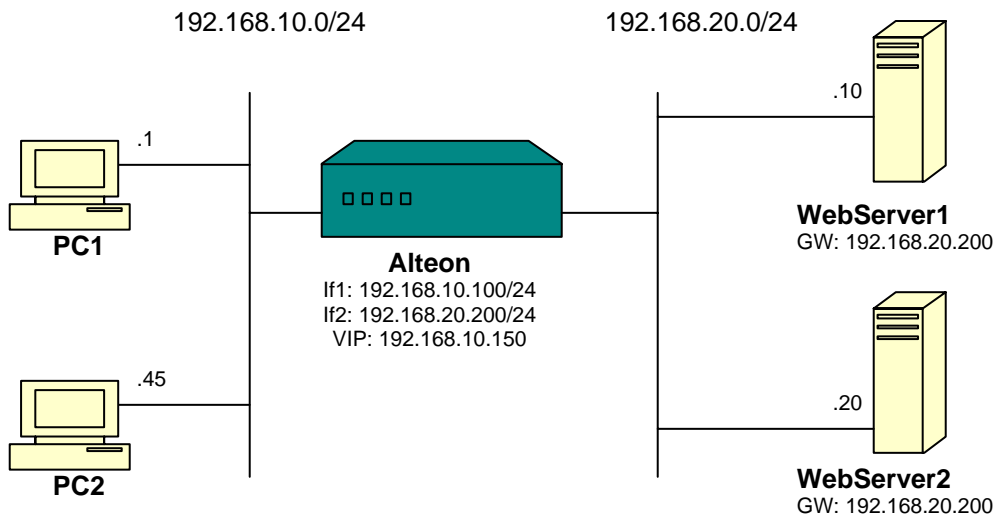
On the other hand using only product type to distinguish between different graphical browsers might not be a best solution as for example Microsoft Internet Explorer, Netscape Navigator, Mozilla and Opera browsers, all report Mozilla as a product. In this case, the comment must be examined.

The check could be run against a platform, version or any other attributes encoded in the User-Agent field. There is always a chance that a user has some custom browser that does not fit into any defined strings.

Note: If different strings are defined for different real servers (for example, a “Mozilla” string is defined for one real server and “MSIE” is defined for another real server) and a request that contains both strings in the User-Agent field is received (for example, request generated by Microsoft Internet Explorer), the request will be load balanced according to the configured load balancing metric.

Sample Configuration

Setup



PC1 – Windows 2000 workstation with Microsoft Internet Explorer 5.5 and Mozilla 1.6 installed, IP address 192.168.10.1/24;

PC2 – Linux Red Hat 9.0 with Lynx 2.8.5-11 installed, IP address 192.168.10.45/24;

Alton – Alteon 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, ports 7, 8 – server ports;

WebServer1 – Linux Red Hat 7.3 server with apache software installed, IP address 192.168.20.10/24, hosts company A web server;

WebServer2 – Linux Red Hat 7.3 server with apache software installed, IP address 192.168.20.20/24, hosts company B web server.

The goal of the configuration is to configure Alton to perform load balancing based on the client’s browser type, and redirect requests initiated by Microsoft Internet Explorer browser to WebServer1, redirect request initiated by Mozilla to WebServer2 and redirect request initiated by Lynx to WebServer2.

Configuring PC1

Note: It is assumed that Microsoft Internet Explorer and Mozilla are installed on PC1.

1. Configure IP address 192.168.10.1/24 on PC:

```
C:\>ipconfig

Windows 2000 IP Configuration
Ethernet adapter Local Area Connection:

    Connection-specific DNS Suffix  . : 
    IP Address. . . . .               : 192.168.10.1
    Subnet Mask . . . . .             : 255.255.255.0
    Default Gateway . . . . .         :
```

Configuring PC2

Note: It is assumed that Lynx is installed on PC2.

1. Configure IP address 192.168.10.45/24 on PC:

```
[root@linux user]# /sbin/ifconfig eth0 192.168.10.45 netmask 255.255.255.0
```

Configuring WebServer1

1. Configure 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 interface 2 IP 192.168.20.200 as a default gateway:

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

3. For testing purposes replace the default index.html file in the root directory with a sample html file that allows you to distinguish between two web servers, for example:

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

4. Start HTTP server:

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

Configuring WebServer2

1. Configure 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 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 root directory with a sample html file that will allow you to distinguish between two web servers, 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 into 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 IP address 192.168.10.100/24 for interface 1 and enable 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 IP address 192.168.20.200/24 for interface 2 and enable 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.
```

Defining strings to be used for Smart Browser load balancing

Define strings to be used to check the User-Agent field.

1. Enter layer 7 SLB configuration menu:

```
>> IP Interface 2# /cfg/slb/layer7/slb/
-----
[Server Loadbalance Resource Menu]
message - Set HTTP error message
add      - Add SLB string for load balance
rename   - Rename SLB string for load balance
rem      - Remove SLB string for load balance
cur      - Display current configuration
```

- View the current settings. Note: Only one default string "any" is defined by default:

```
>> Server Loadbalance Resource# cur
Error message:
No available server to handle this request

Number of entry: 1
1: any
```

- Add a string to look up Microsoft Internet Explorer (MSIE):

```
>> Server Loadbalance Resource# add "MSIE"
```

- Add a string to look up Mozilla version 5.0 product:

```
>> Server Loadbalance Resource# add "Mozilla/5.0"
```

- Add a string to look up Lynx browser:

```
>> Server Loadbalance Resource# add "Lynx"
```

- Apply the changes:

```
>> Server Loadbalance Resource# apply
-----
Apply complete; don't forget to "save" updated configuration.
```

- View the configured strings and note the numbers for each entry as they are going to be used later to assign the matched requests to appropriate real servers:

```
>> Server Loadbalance Resource# cur
Error message:
No available server to handle this request

Number of entries: 4
1: any
2: MSIE
3: Mozilla/5.0
4: Lynx
```

Configuring real servers

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

```
>> Server Loadbalance Resource# /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
```

- Assign requests that match "MSIE" string (entry number 2) to be sent to real server 1:

```
>> Real server 1 # layer7/addlb 2
SLB string ID 2 added to server 1.
```

- 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
```

- Assign "Mozilla/5.0" string (entry number 3) to real server 2:

```
>> Real server 2 # layer7/addlb 3
SLB string ID 3 added to server 2.
```

- Assign "Lynx" string (entry number 4) to real server 2:

```
>> Real Server 2 Layer 7 Commands# addlb 4
SLB string ID 4 added to server 2.
```

Assigning real servers to a group

- Define a real servers group. Place real servers into this group, define appropriate metric (round robin is used in this example), and define appropriate health check mechanism (TCP in this example):

```
>> Real Server 2 Layer 7 Commands# /cfg/slb/group 1/add 1/add 2/metric round/health
tcp
Real server 1 added to real server group 1.
Real server 2 added to real server group 1.
Current metric:      leastconns
New pending metric:  roundrobin
Current health check type: tcp
New pending health check type: tcp
```

Creating virtual server

- Configure virtual server, server 1 in this example. Set IP address 192.168.10.150 in this case 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
```

- Assign HTTP service to virtual server. 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
```

- Enable delayed binding:

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


4. Enable web browser type parsing:

```
>> Virtual Server 1 http Service# httpslb browser
```

Enabling client/server processing on ports

1. Enable client/server processing on appropriate ports. In this configuration client is connected to port 1, server 1 and 2 are connected to port 8, therefore 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

1. Enable direct access mode (DAM):

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

Note: Alternatively Proxy IP addresses can 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
```

2. Apply and save the changes:

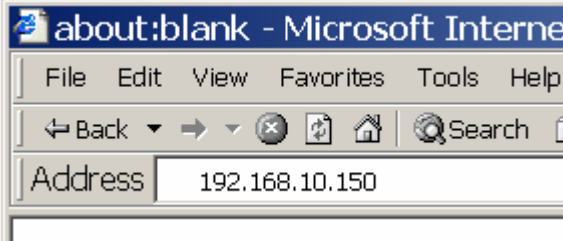
```
>> Layer 4# apply/save
-----
Apply complete; don't forget to "save" updated configuration.
Request will first copy the FLASH "active" config to "backup",
then overlay FLASH "active" with new config.
Confirm saving to FLASH [y/n]:y
New config successfully saved to FLASH.
```

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 10 17:09:22 NOTICE slb: real service 192.168.20.10:80 operational
Jun 10 17:09:22 NOTICE slb: real server 192.168.20.10 operational
Jun 10 17:09:22 NOTICE slb: Services are available for Virtual Server
1:192.168.10.150
Jun 10 17:09:23 NOTICE slb: real service 192.168.20.20:80 operational
Jun 10 17:09:23 NOTICE slb: real server 192.168.20.20 operational
```

Testing the configuration

1. Open Internet Explorer on PC1 and enter the virtual server address 192.168.10.150. Hit Enter:



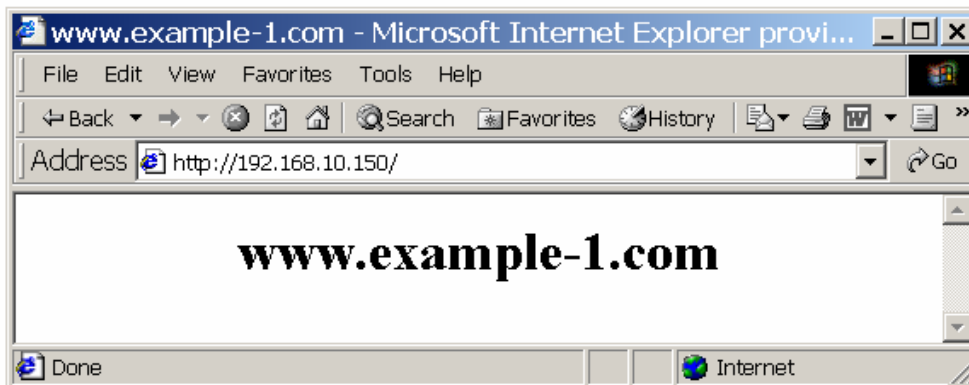
Alteon receives TCP SYN to its virtual server address 192.168.10.150, completes 3-way handshake as delayed binding is enabled. Once TCP handshake is complete, client sends HTTP request:

```

Frame 57 (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: 192.168.10.1 (192.168.10.1), Dst Addr: 192.168.10.150
(192.168.10.150)
Transmission Control Protocol, Src Port: 3987 (3987), Dst Port: http (80), Seq:
3537427588, Ack: 1363764707, 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 17:48:50 GMT\r\n
  If-None-Match: "d45c-93-40cde502"\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
  
```

Upon receiving HTTP request, Alteon inspects the User-Agent portion of the request and looks for any of the defined strings (MSIE, Mozilla/5.0, or Lynx). Once MSIE string is found, Alteon looks up the appropriate real server configured to handle this type of web browser and splits the TCP session to the real server (WebServer1 in this example). Real server processes the request and sends the requested page back to client.

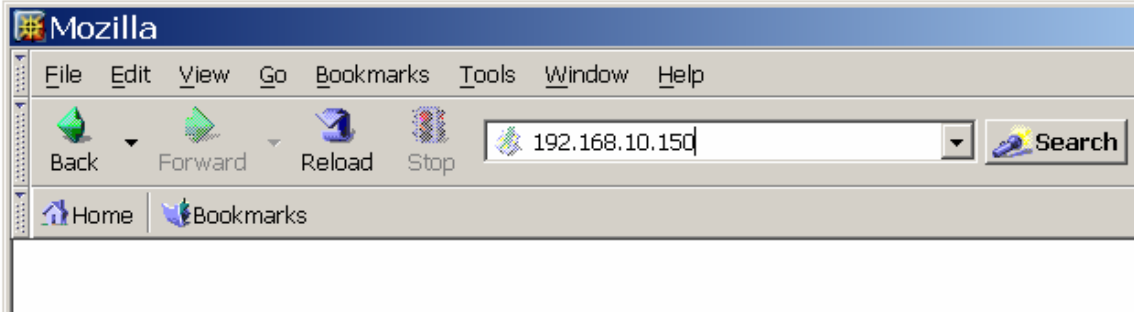
The www.example-1.com page from WebServer1 is displayed:



- a. Check the sessions dump on Alteon. HTTP request containing "MSIE" string in the User-Agent field of the HTTP request was forwarded to the WebServer1 (192.168.20.10):

```
>> Session Table Information# /info/slb/sess/dump
4,4: 192.168.10.1 3987, 192.168.10.150 80 -> 4100 192.168.20.10 80 age 10 E
```

2. Open a Mozilla browser on PC1, enter the IP address of virtual server 192.168.10.150. Hit Enter:

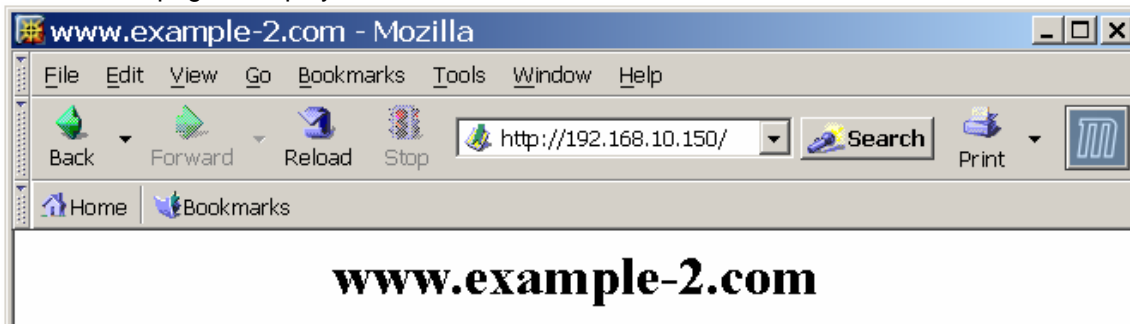


Alteon receives TCP SYN to its virtual server address 192.168.10.150, completes 3-way handshake as delayed binding is enabled. Once TCP handshake is complete, client sends HTTP request:

```
Frame 1427 (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: 192.168.10.1 (192.168.10.1), Dst Addr: 192.168.10.150
(192.168.10.150)
Transmission Control Protocol, Src Port: 3991 (3991), Dst Port: http (80), Seq:
3579340931, Ack: 3382129291, Len: 498
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: Mon, 14 Jun 2004 03:58:23 GMT\r\n
  If-None-Match: "d45b-80-40cd225f"\r\n
\r\n
```

Upon receiving HTTP request, Alteon inspects the User-Agent portion of the request and looks for any of the defined strings (MSIE, Mozilla/5.0, or Lynx). Once "Mozilla/5.0" string is found, Alteon looks up the appropriate real server configured to handle this type of web browser and splits the TCP session to the real server (WebServer2 in this example). Real server processes the request and sends the requested page back to client.

WebServer2 page is displayed:



- a. Check the session table on Alteon. HTTP request containing www.example-2.com string has been forwarded to WebServer2 (192.168.20.20):

```
>> Session Table Information# /info/slb/sess/dump
4,6: 192.168.10.1 3991, 192.168.10.150 80 -> 4102 192.168.20.20 80 age 10 E
```

3. Initiate a text based HTTP session using Lynx from PC2:

```
[user@Linux user]$ /usr/bin/lynx 192.168.10.150
```

Once again, upon TCP handshake completion, client sends HTTP request:

```
Frame 3415 (354 bytes on wire, 354 bytes captured)
Ethernet II, Src: 00:60:cf:46:53:60, Dst: 00:0c:29:63:8f:59
Internet Protocol, Src Addr: 192.168.10.45 (192.168.10.45), Dst Addr: 192.168.20.20
(192.168.20.20)
Transmission Control Protocol, Src Port: 4100 (4100), Dst Port: http (80), Seq:
37006571, Ack: 2381532384, Len: 300
Hypertext Transfer Protocol
  GET / HTTP/1.0\r\n
  Host: 192.168.10.150\r\n
  Accept: text/html, text/plain, audio/mod, image/*, application/msword,
application/pdf, application/postscript, text/sgml, */*;q=0.01\r\n
  Accept-Encoding: gzip, compress\r\n
  Accept-Language: en\r\n
  User-Agent: Lynx/2.8.5dev.7 libwww-FM/2.14 SSL-MM/1.4.1 OpenSSL/0.9.7\r\n
\r\n
```

Alteon inspects User-Agent field, finds "Lynx" string and forwards the request to the real server 2 (WebServer2).

The text based page from WebServer2 appears on PC2 screen:

```
www.example-2.com
```

```
www.example-2.com
```

```
Commands: Use arrow keys to move, '?' for help, 'q' to quit, '<-' to go back.
Arrow keys: Up and Down to move. Right to follow a link; Left to go back.
H)elp O)ptions P)rint G)o M)ain screen Q)uit /=search [delete]=history list
```

- a. Check the sessions table on Alteon:

```
>> Session Table Information# /info/slb/sess/dump
8,8: 192.168.10.45 1033, 192.168.10.150 80 -> 4104 192.168.20.20 80 age 10
```

4. Check URL string matching statistics. Note: Every entered string was matched once:

```
>> Session Table Information# /stats/slb/layer7/str
-----
SLB String stats:
ID SLB String           Hits
1 any                   0
2 MSIE                  1
3 Mozilla/5.0           1
4 Lynx                  1
```

5. Reload web pages in different browsers using Ctrl-F5 to make sure, redirection is preserved. Check statistics again.

Note: In this example, a real server was not defined to process requests from any other web browser, as a result when an unmatched request is received it could not be mapped to any server and therefore the request could not be processed.

Appendix A. Alteon configuration file

```
script start "Alteon AD3" 4 /**** DO NOT EDIT THIS LINE!  
/* Configuration dump taken 17:54:05 Tue Jun 15, 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/layer7/slb  
    ren 2 "MSIE"  
    ren 3 "Mozilla/5.0"  
    ren 4 "Lynx"  
/c/slb/real 1/layer7  
    addlb 2  
/c/slb/real 2/layer7  
    addlb 3  
    addlb 4  
/c/slb/virt 1/service http  
    httpslb useragent  
/  
script end /**** DO NOT EDIT THIS LINE!
```

Copyright © 2004 Nortel Networks Limited - All Rights Reserved. Nortel, Nortel Networks, the Nortel logo, Globemark, and Alteon are trademarks of Nortel Networks Limited.

Microsoft Internet Explorer is a trademark of Microsoft Corporation

Mozilla is a trademark of Netscape Communications Corporation

Netscape Navigator is a trademark of Netscape Communications Corporation

Opera is a trademark of Opera Software AS

Red Hat is a trademark of Red Hat, Inc.

Windows is a trademark of Microsoft 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 your 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