Release Supplement

Release 6

Alteon WebSystems
Web Speed for e-Business

50 Great Oaks Boulevard
San Jose, California 95119
408-360-5500 Main
408-360-5501 Fax
www.alteon.com

Part Number: 050056, Revision B, November 1999

Release Supplement

This release supplement covers new features, bug fixes, and known issues for WebOS Release 6.0.0 (and above). This document is to be used along with the complete documentation: *WebOS Switch Software User's Guide* for Release 5.2 (Part Number 050044, Revision B). Please keep this information with your Alteon WebSystems manuals.

What's in This Document

The following list summarizes the topics, features, and enhancements implemented in WebOS Release 6 that are described in this document:

New in Release 6	Switches Supporting This Feature	See
Trunking Extended to Six Ports	ACEswitch 180, 180+, 180e, ACEdirector2, ACEdirector3	page 32
Weighting for GSLB Sites	ACEswitch 180+, 180e, ACEdirector2, ACEdirector3	page 32
UDP Stateless Load Balancing	ACEswitch 180+, 180e, ACEdirector2, ACEdirector3	page 33
Persistent Mask for SLB	ACEswitch 180+, 180e, ACEdirector2, ACEdirector3	page 33
GSLB Minimum Available Connections	ACEswitch 180+, 180e, ACEdirector2, ACEdirector3	page 34
GSLB Real Server Name Redirection	ACEswitch 180+, 180e, ACEdirector2, ACEdirector3	page 34
Server Fragment ReMap Disabling	ACEswitch 180+, 180e, ACEdirector2, ACEdirector3	page 34
32-bit Low/High Access to 64-Bit Counters	ACEswitch 180, 180+, 180e, ACEdirector2, ACEdirector3	page 34

WebOS Release 6 Upgrade Process

NOTE – WebOS Release 6 includes fundamental changes to the switch software. To avoid serious problems with your switch software upgrade, please read this *Release Supplement* thoroughly and follow the outlined procedures.

This section covers the following topics necessary for installing the WebOS Release 6 upgrade on your Alteon WebSystems web switch:

- Requirements: explains what must be done prior to upgrading your switch software.
- Upgrade procedure: describes how to use TFTP to upgrade the switch software.
- Troubleshooting: describes how to diagnose and correct problems which might occur if the upgrade is improperly installed.

Upgrade Requirements

Before upgrading to WebOS Release 6, your switch ***must*** be running WebOS Release 5.2 with boot kernel 5.x. If your switch is running an older version of switch software, first upgrade to WebOS Release 5.2. The complete procedure for upgrading to Release 5.2 from any earlier version is detailed in the *WebOS 5.2 Release Notes* (Part Number 050045, Revision D).

While upgrading from Release 5.2 to Release 6, you will need the following:

- WebOS Release 5.2.99 transitional switch software image
- WebOS Boot Kernel 6.0 software image
- WebOS Release 6 switch software image

In addition to the above requirements, the following is recommended:

- Keep your *WebOS Release 5.2 User's Guide* (Part Number 050044, Revision B) available
- Since some of the upgrade steps require resetting the switch, be sure to schedule appropriate network downtime for the upgrade process
- Before upgrading the switch software, make a backup of the switch configuration
- For redundancy, the switch can hold two software images. Make sure that both images (`image1` and `image2`) hold valid Release 5.2 software prior to installing the upgrade.

TFTP Upgrade Procedure

NOTE – Do not install WebOS Release 6 until the requirements (see above) are met.

This section explains the TFTP upgrade procedure. To avoid problems with configuration conversion during TFTP software download, please follow the procedure below carefully.

1. **If running ACElerate Release 5.1 or below, STOP!**

   Check the release level of your switch software by using the `/info/sys` command. If the switch is running Release 5.1 or below, you must first upgrade to Release 5.2. The complete procedure for upgrading to Release 5.2 from any earlier version is detailed in the *WebOS 5.2 Release Notes* (Part Number 050045, Revision D).

2. **Make sure that the switch will boot using `image2`.**

   Issue the `/boot/image` command. Specify **`image2`** as the image to use upon next reset.

3. **Reboot the switch.**

   Use the `/boot/reset` command to reboot the run the software in `image2`.

4. **Once WebOS Release 5.2 is installed, download WebOS Release 5.2.99 to `image1`.**

   Release 5.2.99 is a special software image designed to prepare the switch for the Release 6 software. Install the software using the TFTP download command (`/boot/tftp`). Specify **`image1`** as the software image to replace.

   The complete TFTP software download process is described in Chapter 9 of your *WebOS Release 5.2 User's Guide*.

5. **Make sure that the switch will boot using `image1`.**

   Issue the `/boot/image` command. Specify **`image1`** as the image to use upon next reset.

6. **Reboot the switch.**

   Use the `/boot/reset` command to reboot the switch with Release 5.2.99.

7. **Download WebOS Release 6 to `image2` on your switch.**

   NOTE – Be sure to download Release 6 into `image2` instead of `image1`.

   Use the TFTP download command (`/boot/tftp`) to install the Release 6 software. Specify **`image2`** as the software image to be replaced.

8. **Upgrade to boot kernel 6.0.**

   Install boot kernel 6.0 using the TFTP download command (`/boot/tftp`). Specify **`boot`** as the software image to replace.

9. **Make sure that the switch will boot using `image2`.**

   Issue the `/boot/image` command. Specify **`image2`** as the image to use upon next reset.

10. **Reboot the switch.**

    Use the `/boot/reset` command to reboot the switch with Release 6.

    NOTE – If you run Release 5.2 from `image1` after this step in the procedure, you could notice problems with the switch software performance. If you experience problems, see the trouble-shooting notes below.

11. **Verify proper switch operation and then download WebOS Release 6 to `image1`.**

    Install the software using the TFTP download command (`/boot/tftp`). Specify **`image1`** as the software image to replace.

Troubleshooting the Upgrade

Under some conditions, special problems can occur when you attempt to run different boot kernel and software images on the switch. Following is a list of possible problems that might occur, and the corrective actions you should take.

Table 1 Troubleshooting the Release 6 Upgrade

Symptom	Cause	Remedy
Attempt to install Release 6 software fails.	This can occur when running Release 5.2 software (prior to Release 5.2.99)	Install Release 5.2.99 software in `image1`, select `image1` to run at next reboot (`/boot/image`), reboot the switch (`/boot/reset`), and then install Release 6 software.
Attempt to download software into active image fails.	This can occur when running Release 5.2 software in combination with boot kernel 6.0.	Either boot Release 6 software and try again, or install boot kernel 5.x.
When rebooting, the switch does not run the software image selected in `/boot/image`.	This can occur when attempting to run Release 6 software in combination with boot kernel 5.x.	Install boot kernel 6.0 and reboot the switch.
An unexpected system dump appears.	This can occur when running boot kernel 5.x with Release 5.2 from `image2` while Release 6 software is in `image2`.	Install Release 5.2 software in `image2`, then clear the dump.
The Release 6 software in `image1` is corrupted.	When running boot kernel 6.0 with Release 5.2 software in `image2`, a switch panic can write dump information on top of Release 6 software in `image1`.	Boot from `image2`, then install Release 6 in `image1`, select `image1` to run at next reboot (`/boot/image`), and reboot the switch (`/boot/reset`).
You cannot retrieve the system dump after a switch panic.	This can occur when running mismatched software and boot kernels (Release 6 with boot kernel 5.x, or Release 5.2 with boot kernel 6.0).	If boot kernel 5.x is installed, boot Release 5.2 software. If boot kernel 6.0 is installed, boot Release 6 software.

Down-Versioning to WebOS Release 5.2

NOTE – WebOS Release 6 is fundamentally different from previous releases. To avoid serious problems with your switch software, please read this *Release Supplement* thoroughly and follow the proper procedures.

This section covers the following topics necessary for down-versioning from WebOS Release 6 to Release 5.2:

- Requirements: explains what must be done prior to down-versioning.
- Down-versioning procedure: describes how to use TFTP to down-version the switch software.
- Troubleshooting: describes how to diagnose and correct problems which might occur with down-versioning.

Requirements

This section explains the procedure for moving from WebOS Release 6 back to Release 5.2. You cannot down-version directly to ACElerate Release 5.1 or prior from Release 6. If you need to run Release 5.1 or prior, first down-version to Release 5.2 as outlined in this document. Then see the *WebOS 5.2 Release Notes* (Part Number 050045, Revision C) for down-versioning to earlier switch software releases.

While down-versioning to Release 5.2, you will need the following:

- WebOS Release 5.2.x switch software image (prior to Release 5.2.99).
- WebOS Boot Kernel 5.x software image

In addition to the above requirements, the following is recommended:

- Keep your *WebOS Release 5.2 User's Guide* (Part Number 040044, Revision B) available
- Since some of the steps require resetting the switch, be sure to schedule appropriate network downtime for the down-versioning process
- Before performing the procedure, make a backup of the switch configuration
- For redundancy, the switch can hold two software images. Make sure that both images (`image1` and `image2`) hold valid Release 6 software prior to down-versioning.

Down-Versioning Procedure

NOTE – Do not down-version to until the requirements (see above) are met.

This section explains the TFTP down-versioning procedure. To avoid serious problems with the switch software, please follow the procedure below carefully.

1. **If down-versioning to ACElerate Release 5.1 or prior, STOP!**

   You cannot down-version directly to ACElerate Release 5.1 or prior from Release 6. If you must first down-version to Release 5.2 as outlined in this document. Then see the *WebOS 5.2 Release Notes* (Part Number 050045, Revision D) for down-versioning to earlier switch software releases.

2. **Make sure that the switch will boot using `image2`.**

   Issue the `/boot/image` command. Specify **`image2`** as the image to use upon next reset.

3. **Reboot the switch.**

   Use the `/boot/reset` command to reboot the run the software in `image2`.

4. **Download WebOS Release 5.2 to `image1` on your switch.**

   Use the TFTP download command (`/boot/tftp`) to install the Release 5.2 software (prior to Release 5.2.99). Specify **`image1`** as the software image to be replaced.

5. **Download boot kernel 5.2.**

   Install boot kernel 5.2 using the TFTP download command (`/boot/tftp`). Specify **`boot`** as the software image to replace.

6. **Make sure that the switch will boot using `image1`.**

   Issue the `/boot/image` command. Specify **`image1`** as the image to use upon next reset.

7. **Reboot the switch.**

   Use the `/boot/reset` command to reboot the switch with Release 5.2.

NOTE – If you run Release 6 from `image2` after this step in the procedure, you could notice serious problems with the switch software performance. If you experience problems, see the troubleshooting notes below.

In some cases after rebooting, an unexpected dump file might exist on the switch. The dump is invalid and can be ignored. To clear the dump, use the `/main/cldmp` command.

8. **Verify proper switch operation and then download WebOS Release 5.2 to `image2`.**

   Install the software using the TFTP download command (`/boot/tftp`). Specify **`image1`** as the software image to replace.

Troubleshooting

Under some conditions, special problems can occur when you attempt to run different boot kernel and software images on the switch. See "Troubleshooting the Upgrade" on page 7 for a list of possible problems that might occur, and the corrective actions you should take.

URL-Based Web Cache Redirection

WebOS Release 6.0 enables you to send requests with specific URLs or URL sub-strings to designated cache servers. The URL-based redirection option allows you to perform cache server farm tuning and offload overhead processing from the cache servers by only sending appropriate requests to the cache server farm. This feature supports both HTTP 1.0 and HTTP 1.1.

Each request is examined and handled as described below:

- If the request is a non-GET request like HEAD, POST, PUT, or HTTP with cookies, it doesn't get sent to the cache.
- If the request is an ASP or CGI request, or a dynamically generated page, it doesn't get sent to the cache.
- If the request is a cookie, it can optionally bypass the cache.

Network administrators can configure up to 32 URL expressions, each 8 bytes long, for non-cacheable content types. Up to 64 strings, comprising 40 bytes each, can be used for URL sub-string matching. As each URL web request is examined, non-cacheable items are forwarded to the origin server and requests with sub-string matches are redirected to the appropriate cache server.

NOTE – The term "origin server" refers to the server originally specified in the request.

Examples of sub-strings are:

- "/product" - matches URL that starts with "/product," including any information in the "/product" directory
- "product" - matches URL that has the string "product" anywhere in the entire URL

The switch is pre-configured with a list of thirteen non-cacheable items that the network administrator can add, delete, or modify via the user interface. These items are either known dynamic content file extensions or dynamic URL parameters, as described below:

- dynamic content file extensions: cgi (cgi files)
- cfm (cold fusion files), .asp (ASP files), bin (bin directory), cgi-bin (cgi-bin directory), shtml (scripted html), .htx (Microsoft HTML extension file), .exe (executable)
- dynamic URL parameters: +, !, %, =, &

By sending requests with specific URLs or URL sub-string matches to designated web servers, the network administrator can perform web server farm tuning.

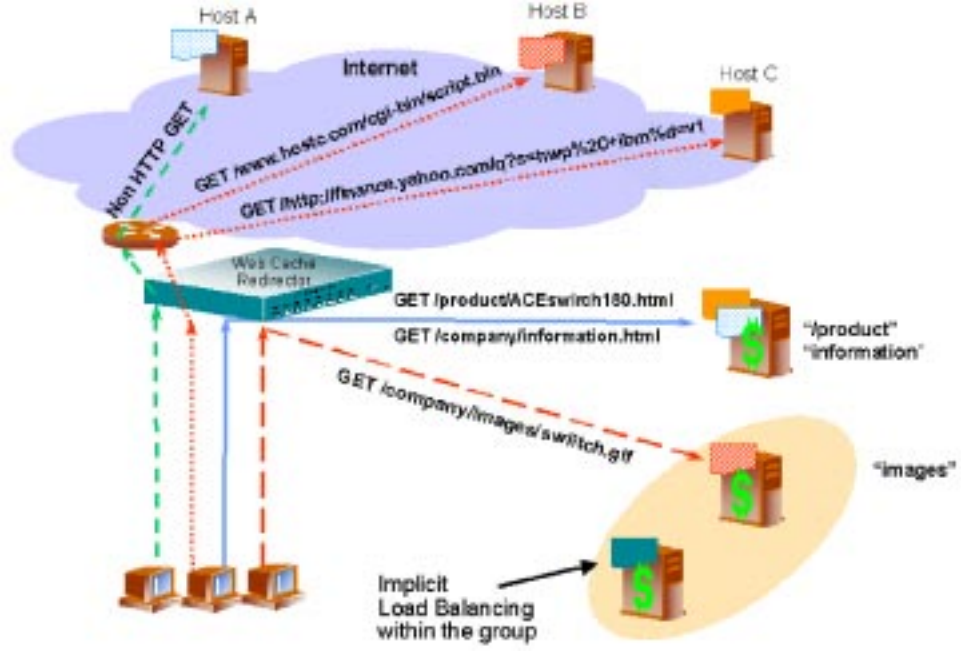

Figure 1 URL-Based Web Cache Redirection

An HTTP “Host” header, if present, will be used for hashing to determine content location. All requests for *www.alteon.com*, for example, will be forwarded to the same cache server.

Requests will be load balanced among the multiple servers matching the URL according to the metric specified for the server group (`leastConns` is the default).

Configuring URL-Based Web Cache Redirection

To configure URL-based web cache redirection, perform the following steps:

1. **Before you can configure URL-based web cache redirection, you must first configure real servers and services for basic server load balancing, as indicated below:**

   - Define each real server and assign an IP address to each real server in the server pool.
   - Define a real server group and set up health checks for the group.
   - Enable server load balancing on the switch.

   For information on how to configure your network for server load balancing, see Chapter 15, "Server Load Balancing," in the *WebOS 5.2 User's Guide*.

2. **Configure a switch to support web cache redirection.**

   For information on this topic, see Chapter 16, "Filtering," of the *WebOS 5.2 User's Guide*.

3. **Configure the parameters and file extensions that will bypass web cache redirection, using the commands described below:**

   Use the `/cfg/slb/url/redir` command to add or remove expressions that should not be cacheable. The switch is pre-configured with a list of thirteen non-cacheable items. Listed below, these items are either known dynamic content file extensions or dynamic URL parameters:

   - dynamic content file extensions: cgi (cgi files), .cfm (cold fusion files), .asp (ASP files), bin (bin directory), cgi-bin (cgi-bin directory), .shtml (scripted html), .htx (Microsoft HTML extension file), .exe (executable)
   - dynamic URL parameters: +, !, %, =, &

   Use the `/cfg/slb/url/redir/urlal` command to enable/disable cache redirection, as described below.

   - **`Enable`**: Switch will redirect all non-GET requests to the origin server.
   - **`Disable`**: Switch will compare all requests against the expression table to determine whether the request should be redirected to a cache server or the origin server.

Use the `/cfg/slb/url/redir/cooki` command to enable/disable cache redirection of requests that contain "`cookie:`" in the HTTP header, as described below.

- **`Enable`**: Switch will redirect all requests that contain "`cookie:`" in the HTTP header to the origin server.
- **`Disable`**: Switch will compare the URL against the expression table to determine whether the request should be redirected to a cache server or the origin server.

4. **Define the load balancing string(s) for URL or web cache server load balancing.**

   Use the `/cfg/slb/url/lb` command and the parameters listed below to define a string.

   - **`add`**: Add string or a path.
   - **`rem`**: Remove string or a path.

   A default string "`any`" means the particular server can handle all URL or web cache requests. A string that starts out with a slash (/) such as "`/images`" means if this string is applied to a particular server, that server can only handle requests that start out with the "`/images`" string.

   Example: With the "`/images`" string, the server will handle these requests:

   ```
   /images/product/b.gif
   /images/company/a.gif
   /images/testing/c.jpg
   ```

   This server will not handle these requests:

   ```
   /company/images/b.gif
   /product/images/c.gif
   /testing/images/a.gif
   ```

   A string that doesn't start out with a slash (/) means that if this string is applied to a particular server, that server can handle any requests that contain the defined string.

   Example: With the "`images`" string, the server will handle these requests:

   ```
   /images/product/b.gif
   /images/company/a.gif
   /images/testing/c.jpg
   /company/images/b.gif
   /product/images/c.gif
   /testing/images/a.gif
   ```

For easy configuration and identification, each defined string has an ID attached, as shown in the following example:

Example: Number of entries: 6

ID	SLB String
1	any
2	.gif
3	/sales
4	/xitami
5	/manual
6	.jpg

5. **Configure a real server to handle URL-based load balancing or web cache redirection.**

   If you don't add a defined sub-string (or add the defined sub-string "`any`"), the server will handle any request.

   Use the `/cfg/slb/real 2/addlb` *ID* command to add a defined sub-string.

   where *ID* is the identification number of the defined string.

   Example:

```
>> /cfg/slb/real 2/addlb 3
```

   Use the `/cfg/slb/real 2/remlb` *ID* command to remove a defined sub-string.

   Example:

```
>> /cfg/slb/real 2/remlb 3
```

   The server can have multiple defined sub-strings, as shown below:

   - "/images"
   - "/sales"
   - ".gif"

   With these defined strings, the server can handle requests that start out with "/images" or "/sales" and any requests that contain "`.gif`"

6. **Configure a filter to support normal web cache redirection.**

   For instructions on how to configure filters for web-cache redirection, see "Example Configuration for the Web-Cache Solution" in Chapter 16, "Filtering," of the *WebOS 5.2 User's Guide*.

7. **Enable URL-based web cache redirection on the filter.**

   Use the `/cfg/slb/filt` *filter-number*`/url enable` command to turn on URL-based web cache redirection for a particular filter.

8. **On the switch, create a default filter (for non-cached traffic) and assign the filter to the client port.**

9. **Enable and apply the configuration.**

Statistics for URL-Based WCR

Use the `/stats/slb/url/redir` command to show the number of hits to the cache server or origin server.

Example:

Total Cache Server Hits	73942
Total Origin Server Hits	2244

URL-Based Server Load Balancing

By separating static and dynamic content requests via HTML parsing, URL-based server load balancing allows the network administrator to optimize resource access and server tuning. Content dispersion can be optimized by basing load balancing decisions on the entire path/file-name of each URL. This WebOS 6.0 feature supports both HTTP 1.0 and 1.1.

URL-based load balancing operates in a manner similar to URL parsing for web cache redirection, except that the switch virtual IP address (VIP) is the target of all IP/HTTP requests.

Network administrators can configure up to 64 strings, comprising 40 bytes each, for URL matching. Each URL web request is then examined against the URL strings defined for each real server, as described under "URL-Based Web Cache Redirection" on page 10. URL requests will be load balanced among the multiple servers matching the URL, according to the metric specified in the server group (`leastConns` is the default).

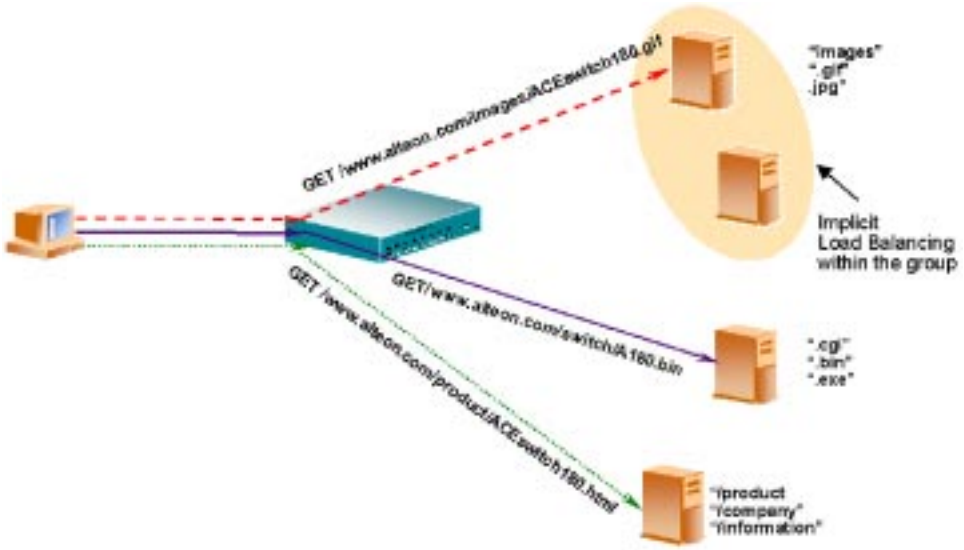

Figure 2 URL-Based Server Load Balancing

Example: The network administrator specifies the following criteria for load balancing:

- Requests with ".cgi" in the URL: forwarded to servers RIP1, RIP2, RIP5.
- Requests with the sub-string "images" in the URL: sent to servers RIP3, RIP4 and RIP6.
- Requests with URLs starting with the sub-string "/product:" sent to servers RIP2, RIP3 and RIP5.
- Requests containing URLs with anything else: sent to servers RIP1, RIP2, RIP3. These servers have been defined with the "any" string.

Configuration Issues When Using URL-based SLB

URL-based SLB has the following limitations:

- When global server load balancing (GSLB) is enabled, URL-based server load balancing is not supported.
- When URL based SLB is used in an active/active redundant setup, use a proxy IP address (PIP) instead of Direct Access Mode (DAM) to enable URL parsing feature.

  Because Direct Access Mode imposes a topology restraint on your network, i.e., that all frames coming back through the switch server port must egress the switch via the same client port where they were originally received, it does not allow you to use URL server load balancing with active-active redundancy.
- Firewall load balancing (FWLB) and URL-based SLB by the same switch on the clean side of the network does not work.

Configuring URL-Based Server Load Balancing

To configure URL-based server load balancing, perform the following steps:

1. **Configure real servers and services for basic server load balancing, as indicated below:**
   - Define each real server and assign an IP address to each real server in the server pool.
   - Define a real server group and set up health checks for the group.
   - Define a virtual server on virtual port 80 (HTTP) and assign a real server group to service it.
   - Enable server load balancing on the switch.
   - Enable client processing on the port connected to the client.

   For information on how to configure your network for server load balancing, see Chapter 15, “Server Load Balancing,” in the *WebOS 5.2 User’s Guide*.
2. **Define the load balancing string(s) for URL server load balancing.**

   Use the `/cfg/slb/url/lb` command and the parameters listed below to define a string.
   - **`add`**: Add string or a path.
   - **`rem`**: Remove string or a path.

   A default string ’any’ means the particular server can handler all URL or web cache requests. A string that starts out with a slash (/) such as ’/images’ means if this string is applied to a particular server, that server can only handle requests that start out with the ’/images’ string.

NOTE – For examples related to this step, see Step 4 under "Configuring URL-Based Web Cache Redirection" on page 12.

3. **Configure a real server to handle URL-based load balancing.**

   If you don't add a defined sub-string (or add the defined sub-string "any"), the server will handle any request.

   Use the `/cfg/slb/real 2/addlb ID` command to add a defined sub-string.

   where *ID* is the identification number of the defined string.

   Example:

```
>> /cfg/slb/real 2/addlb 3
```

   Use the `/cfg/slb/real 2/remlb ID` command to remove a defined sub-string.

   Example:

```
>> /cfg/slb/real 2/remlb 3
```

   The server can have multiple defined sub-strings, as shown below:

   - '/images'
   - '/sales'
   - '.gif'

   With these defined strings, the server can handle requests that start out with '/images' or '/sales' and any requests that contain '`.gif`'

4. **Enable URL parsing on the client port.**

   For URL server load balancing to work, request processing must be done on the client port. You can enable client-side processing using either method described below:

   - Enable Direct Access Mode, using the `/cfg/slb/direc ena` command.
   - Configure a proxy IP address on the client port (e.g., `/cfg/slb/port` *port-number*`/pip` 12.12.12.12). If you use this method, ensure that Direct Access Mode is disabled (the default setting), using the `/cfg/slb/direc dis` command.

NOTE – Direct Access Mode imposes a topology restraint on your network. It requires that all frames coming back through the switch server port must egress the switch via the same client port where they were originally received. When DAM is enabled, it can create problems in address re-mapping for networks that have more than one active ingress router.

5. **Enable URL-based SLB on the virtual server(s), using the `/cfg/slb/virt` *virtual-server-number*`/uslb ena` command.**

Statistics for URL-Based SLB

Use the `/stats/slb/url/lb` command to show the number of hits to the SLB or cache server.

Example:

ID	SLB String	Hits	ID	SLB String	Hits
1	any	73881	4	/xitami	162102
2	.gif	0	5	/manual	0
3	/sales	0	6	.jpg	0

VRRP for Active/Active Redundancy

WebOS Release 5.2 and above includes Virtual Router Redundancy Protocol (VRRP) for redundancy to routers within a LAN. In addition, Alteon WebSystems has extended VRRP to include virtual servers as well, allowing for full active/active redundancy between its Layer 4 switches. This allows for more efficient network resource allocation than the old hot-standby method. It also supports more complex failover topologies.

VRRP Active/Active Synchronization

The old hot-standby failover required the primary and secondary switches to have identical configurations and port topology. With VRRP and active/active failover, this is optional.

If desired, each switch can be configured individually with different port topology, Server Load Balancing, and filters.

If you would rather force two active/active switches to use identical settings, you can synchronize their configuration using the following command:

`/oper/slb/sync` *IP_address*

The `sync` command copies the following settings to the switch at the specified IP interface address:

- VRRP settings
- Server Load Balancing settings (including SLB port settings)
- Filter settings (including filter port settings)

If you perform the `sync` command, you should check the configuration on the target switch to ensure that the settings are correct.

NOTE – In WebOS version 5.2.21, the `sync` command also copies IP proxy settings to the target switch. This creates duplicate IP addresses on your network. To correct this problem, you must reconfigure each IP proxy on the target switch to use a unique IP address.

VRRP, STP, and Failover Response Time

VRRP active/active failover is significantly different from the hot-standby failover method in previous releases. One important difference is that VRRP generally requires Spanning-Tree Protocol (STP) to be enabled in order to resolve bridge loops that usually occur in cross-redundant topologies like the one shown below.

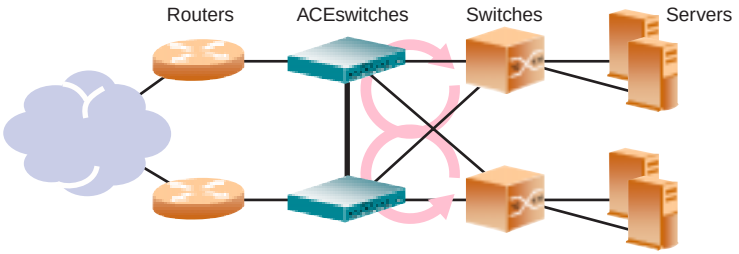

Figure 3 Cross-redundancy creates loops, but STP resolves them

In this example, a number of loops are wired into the topology. STP resolves loops by blocking ports where looping is detected.

One drawback to using STP with VRRP is the failover response time. STP could take as long as 45 seconds to reestablish alternate routes after a switch or link failure.

When using VRRP in WebOS Release 5.2, you can decrease failover response time by using VLANs instead of STP to separate traffic into non-looping broadcast domains. For example:

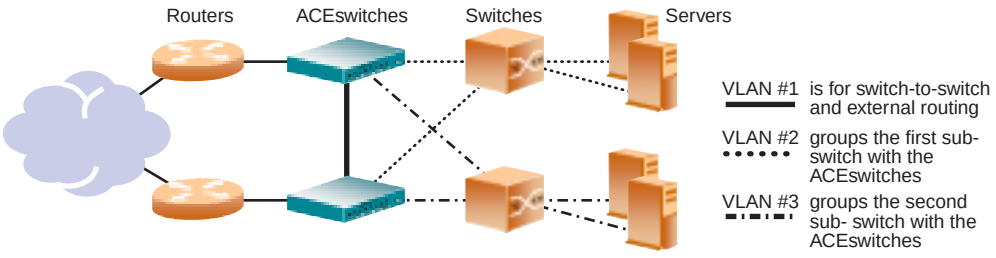

Figure 4 VLANs can be used to create non-looping topologies.

The topology above allows STP to be disabled. On the ACEswitches, IP routing allows traffic to cross VLAN boundaries. The servers use the ACEswitches as default gateways. For port failure, traffic is rerouted to the alternate path within one health-check interval (configurable between 1 and 60 seconds, with a default of 2 seconds).

VRRP Virtual Router ID Numbering

During the software upgrade process (see below), VRRP virtual router IDs will be automatically assigned if failover is enabled on the switch. When configuring VRRP virtual routers at any point after upgrade, virtual router ID numbers (`/cfg/vrrp/vr #/vrid`) must be assigned in accordance with the following restrictions:

- The virtual router ID may be configured as ***any number*** between 1 and 255 when the virtual router IP address is not assigned the same value as a virtual server IP address.
- The virtual router ID must be configured as an ***odd number*** between 1 and 255 under the following circumstance:
  - The virtual router uses Layer 4 services (its virtual router IP address is the same as assigned to a virtual server), *and*...
  - Layer 3 binding is turned on for the virtual server (by enabling the `layr3` option on the virtual server menu: `/cfg/slb/virt`)
- The virtual router ID must be configured as an ***even number*** between 2 and 254 under the following circumstance:
  - The virtual router uses Layer 4 services (its virtual router IP address is the same as assigned to a virtual server), *and*...
  - Layer 3 binding is turned off for the virtual server (by disabling the `layr3` option on the virtual server menu: `/cfg/slb/virt`)

Persistence

When a particular server has data associated with a specific user, *persistence* enables successive connections to be sent to that server. In WebOS Release 5.2 and earlier versions of Alteon WebSystems' switch software, TCP/IP session persistence is achieved using the source IP address as the key identifier.

There are two major problems associated with persistence based on a packet's IP source address:

- **No SLB:** Proxied clients will appear to the switch as a single IP source address and will not be able to take advantage of server load balancing on the switch. When many individual users behind a firewall use the same IP proxy source address, requests will be directed to the same server, without the benefit of load balancing the traffic across multiple servers. Persistence is supported without the capability of effectively distributing traffic load.
- **No Persistence:** When individual users share a pool of IP source addresses, persistence for any given request cannot be assured. Although each IP source address will be directed to a specific server, the address itself is randomly selected, thereby making it impossible to predict which server will receive the request. Server load balancing is supported, without true persistence for any given user.

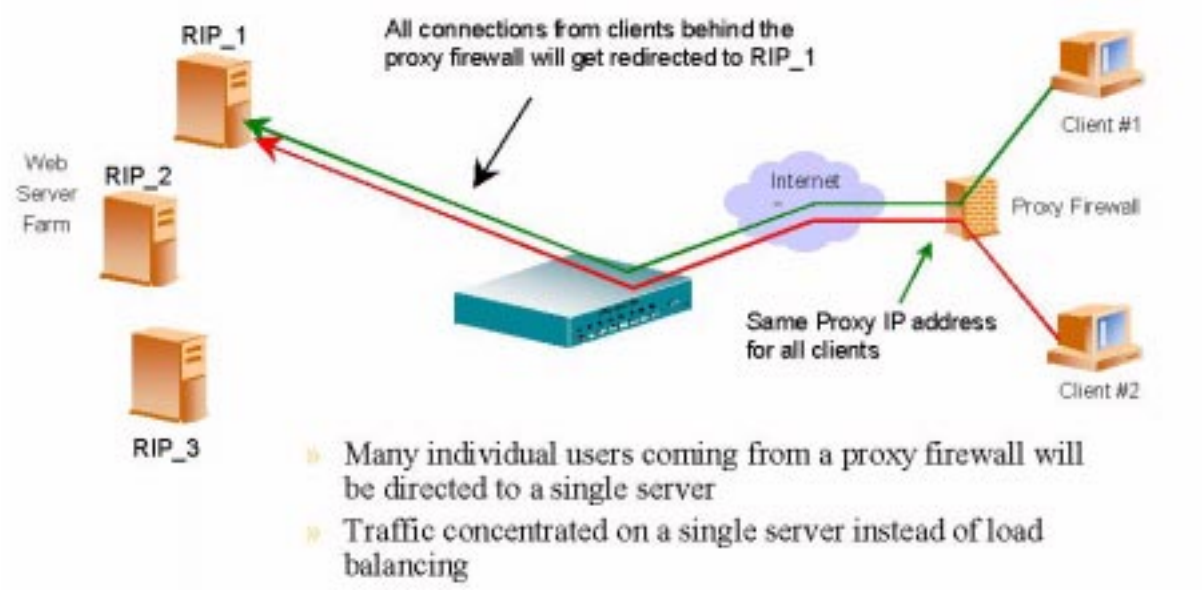

Figure 5 Example 1 - Why IP Source Persistence Doesn't Work: No SLB

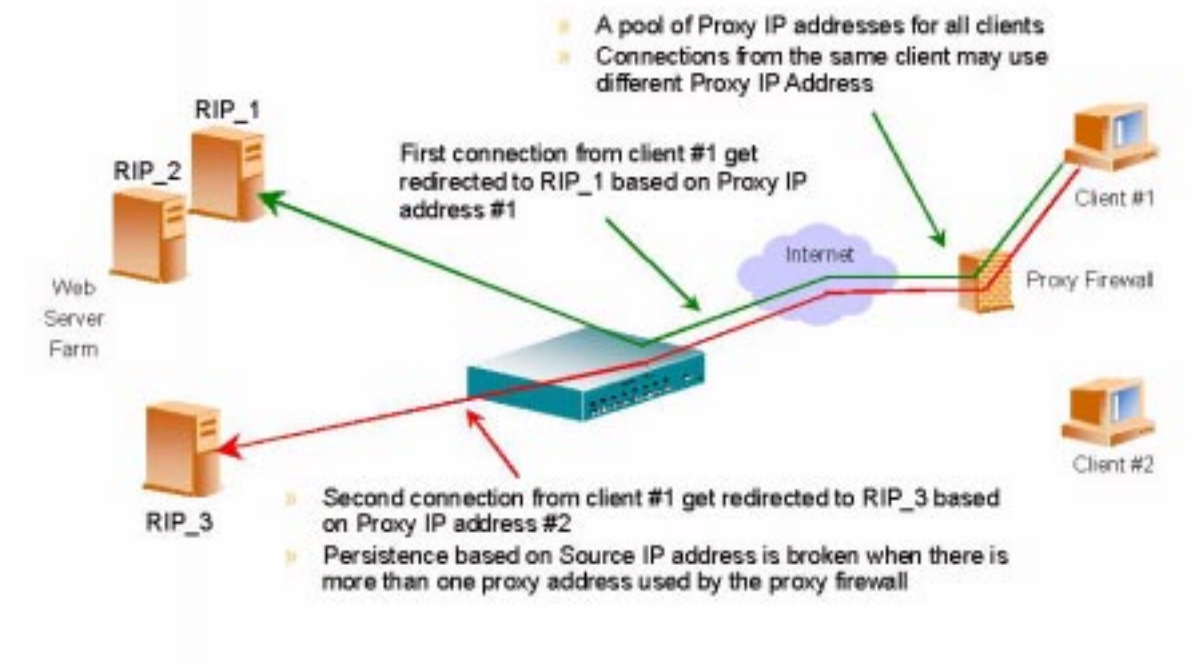

Figure 6 Example 2 - Why IP Source Persistence Doesn't Work: No Persistence

WebOS Release 6.0: Rather than using the current method based on the IP source address, WebOS Release 6.0 allows you to select between the following methods of persistence:

- URL cookie-based persistence in balancing HTTP requests. For more information, see "URL Cookie-Based Persistence" on page 25.
- SSL session-based persistence in SSL/HTTPS load balancing. For more information, see "SSL Session ID-Based Persistence" on page 28.

URL Cookie-Based Persistence

“Cookies” are a general mechanism for maintaining state information between a client and server. When used, the server issues a token (the cookie) to the client upon receiving the initial request. The client will then send the cookie back to the server on each subsequent request as a means of identification.

This cookie may represent a customer ID, a token of trust showing that authentication has already been performed, or an index to client information stored on the server (such as shopping basket contents). In more complex applications, the cookie can be encoded with special information, allowing the client access to customized information or preferential content.

When cookies are used, persistence is required to ensure that the client connects to the same server each time. To prevent persistence problems such as those shown on page 23 (loss of Server Load Balancing or the failure of persistence), the URL cookie-based persistence feature in WebOS Release 6.0 uses cookie content (such as real server IP address or other user distinguishing value) to maintain persistence. Because this supports client connections to the same server while allowing sessions with different cookies to be effectively distributing traffic across multiple servers, it provides one of the best persistence methods for HTTP traffic.

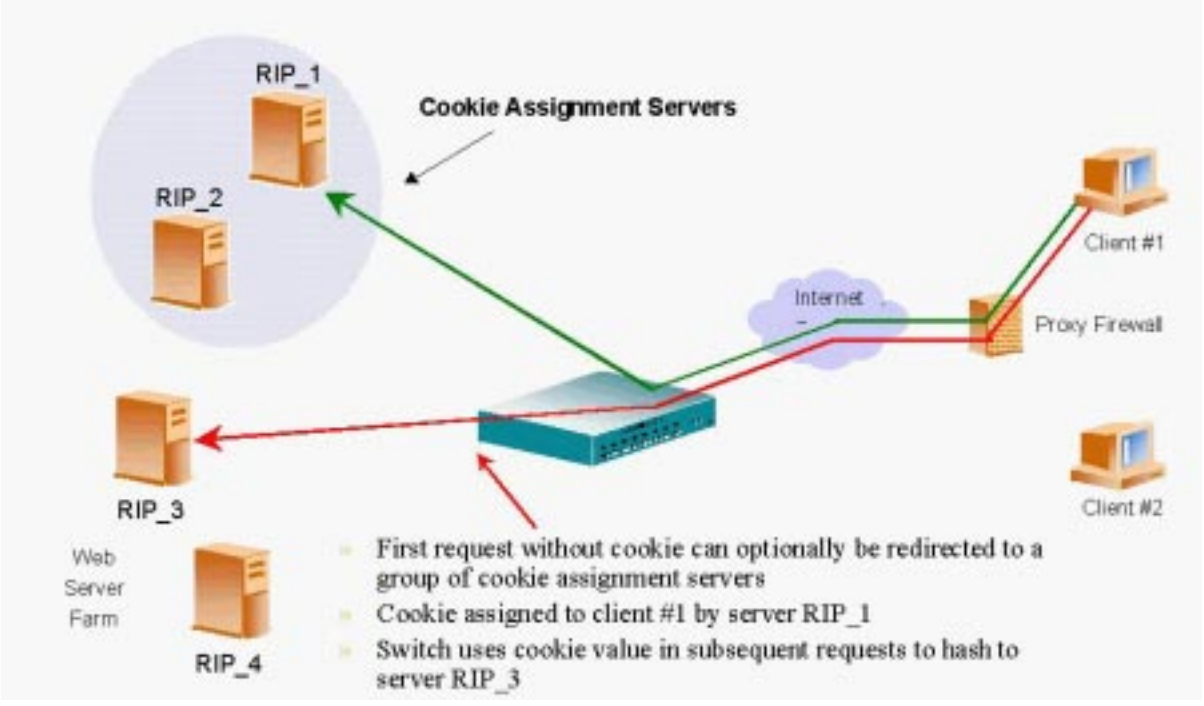

Figure 7 Cookie-Based Persistence

How Cookie-Based Persistence Works

- Network manager specifies cookie name, offset and length as a key for persistence
- Cookies can be generated by a cookie assignment server or the real server itself
- Cookies can be dynamically generated or permanently embedded in client browser
- Cookies are stored with connection binding information
- Subsequent requests with same cookie are sent to the same server

Configuring Cookie-Based Persistence

1. **Enable URL Server Load Balancing in the virtual server. An example is given below:**

```
>> /cfg/slb/virt virtual-server-number# cooki enable
```

2. **Configure the group SLB metric to use HASH. An example is given below:**

```
>> /cfg/slb/group virtual-server-number# metrc hash
```

3. **Configure a cookie name, offset byte, and extracting length. An example is given below:**

```
>> /cfg/slb/virt virtual-server-number# cname sid 2 4
```

Example:

HTTP HEADER:

```
Cookie:  sid=0123456789abcdef; name1=value1;...
```

We want to use cookie name ’`sid`’ and ’`789a`’ of the value as a hashing key to the real server.

The configuration will be: `/cfg/slb/virt 1/cname sid 8 4`

If we want the whole value of ’`sid`’, the configuration will be:

```
/cfg/slb/virt 1/cname sid 1 16
```

Directing Cookie Client to a Specific Server

By embedding the real server's IP address (in hexadecimal format) in the cookie value, the cookie assignment server or real server can tell the client exactly which server to go back to in the subsequent connections.

1. **The server (cookie assignment server or real server) converts the client IP address from dotted format into hexadecimal. An example is given below:**

```
205.178.15.4 --> cdb20f04
```

2. **Configure the cookie assignment server to embed the converted address into the cookie value.**

```
sid=1234cdb20f043243
```

3. **Configure the switch to read in the correct cookie value. An example is given below:**

```
>> /cfg/slb/virt virtual-server-number# cname sid 5 8
```

Assigning a Server to Be the Cookie Assignment Server for Client Requests Without the Specified Cookie

Configuration: **`/cfg/slb/real 1/nocok enabled`**

If Real Server Group 1 contains servers 1,2,3,4,5, and 6, and only Real Servers 5 and 6 have `/nocok` enabled, when client requests come in for the first time and don't have the specified cookie, servers 5 and 6 will be load balanced to assign a specified cookie to the server. When a client request comes in with the specified cookie, it will get hashed into either real server 1,2,3, or 4.

SSL Session ID-Based Persistence

Secure Sockets Layer (SSL) is a set of protocols built on top of TCP/IP that allow an application server and user to communicate over an encrypted HTTP session, providing authentication, non-repudiation, and security. The SSL protocol "handshake" is performed using clear text; the content data is then encrypted, using an algorithm exchanged during the "handshake," prior to being transmitted.

Session persistence allows you to re-establish a user's connection to a particular server. This is an important consideration for administrators of business web sites, where a server may have data associated with a specific user that is not dynamically shared with other servers at the site.

NOTE – When global server load balancing (GSLB) is enabled, SSL-based server load balancing is not supported. Because the session state is held by the SSL web servers, Global SLB would not be able to recover sessions in progress if the current site were to crash. Subsequent requests would be forwarded to the nearest GSLB server, as is done with other applications.

Using the SSL session ID, the switch forwards the request to the real server that it bound the user to during the last session. Using the session ID will enhance network performance. Negotiating the type of encryption used during data transmission requires a lengthy protocol exchange that is bypassed with session persistence.

How SSL Session ID-Based Persistence Works

- All SSL sessions which present the same session ID (32 random bytes chosen by the SSL server) will be directed to the same real server.

NOTE – The SSL session ID is only "visible" to the switch after the TCP 3-way handshake. In order to make a forwarding decision, the switch must terminate the TCP connection to examine the request.

- New sessions are sent to the real server based on metric (`Hash`, `roundrobin`, `leastconns`, and `minmisses`).
- If no session ID is presented by the client, the switch picks a real server based on the metric for the real server group and waits until a connection is established with the real server and a session ID is received.

- The session ID is stored in a session hash table. When a subsequent connection comes in with the same session ID, it is sent to the same real server.

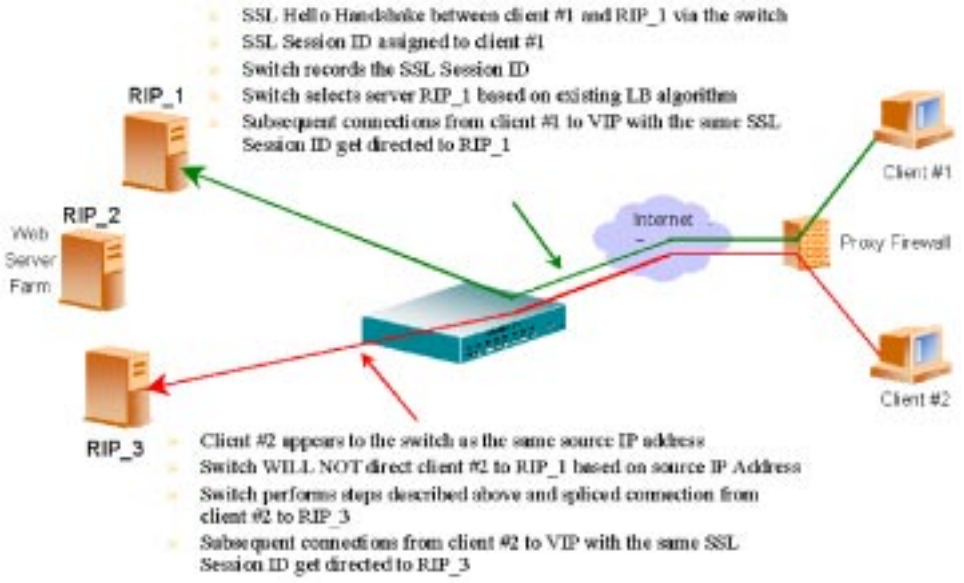

Figure 8 SSL Session ID-Based Persistence

Configuring SSL Session ID-Based Persistence

To configure session ID-based persistence for a real server, perform the following steps:

1. **Configure real servers and services for basic server load balancing, as indicated below:**
   - Define each real server and assign an IP address to each real server in the server pool.
   - Define a real server group and set up health checks for the group.
   - Define a virtual server on virtual port 443 (HTTPS) and assign a real server group to service it.
   - Enable server load balancing on the switch.
   - Enable client processing on the port connected to the client.

   For information on how to configure your network for server load balancing, see Chapter 15, “Server Load Balancing,” in the *WebOS 5.2 User’s Guide*.

2. **If a proxy IP address is not configured on the client port, use the `/cfg/slb/direc ena` command to enable Direct Access Mode for real servers.**

3. **Select the persistent binding type for the virtual port to configure session ID-based persistence. An example is given below:**

```
>> /cfg/slb/virt virtual-server-number# pbind 443 sessid
```

NOTE – Session ID-based persistence does not change the use of the **pbind** `enabled`/`disabled` parameters.

HTTPS/SSL Health Check

In WebOS Release 6.0, a health check option has been added to the Real Server Group Menu (`/cfg/slb/group/healt/sslh`) that allows the switch to query the health of the SSL servers by sending an SSL client "Hello" packet and then verify the contents of the server's "Hello" response.

The SSL enhanced health check behavior is summarized below:

- The switch sends a SSL "Hello" packet to the SSL server.
- If it is up and running, the SSL server responds with the "Server Hello" message.
- The switch verifies various fields in the response and marks the service "UP" if the fields are OK.

During the handshake, the user and server exchange security certificates, negotiate an encryption and compression method, and establish a session ID for each session.

Service Group Failover

In WebOS Release 5.2 and earlier versions of Alteon WebSystems' switch software, a real server with multiple services configured for load balancing will be taken out of rotation if any single service on the server fails a health check.

On Alteon WebSystems' switches running WebOS Release 6.0, a service can fail a health check and the server running that service will remain available to load balanced requests from the switch for other services running on that server.

To provide service group failover, several real server groups are configured with the same real servers and then assigned to support a virtual server with different services. When a service health check fails on one server, that server is placed out of service in one real server group only. Because it is in more than one real server group, the server can still respond (in another server group) to load-balanced requests for other services it is running.

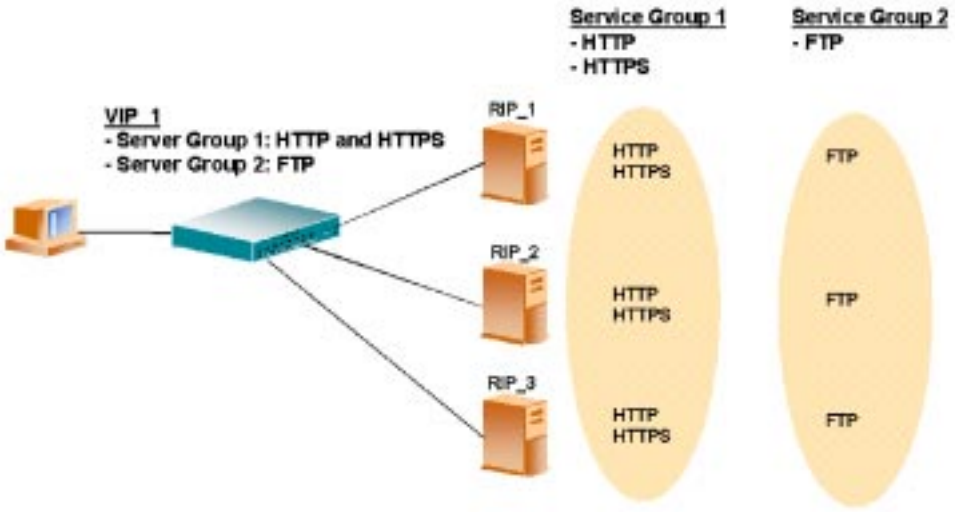

Figure 9 Service Group Failover

Example: Service group failover on Alteon WebSystems' switches running WebOS 6.0:

1. **Real servers #1, #2, and #3 are created. All servers can provide HTTP, HTTPS, and FTP service.**
2. **Real servers #1, #2, and #3 are added to Real Server Groups 1 and 2.**
3. **Virtual Server #1 is created. Real Server Group 1 is assigned to service HTTP and HTTPS. Real Server Group 2 is assigned to service FTP.**
4. **If a HTTP health check sent to Real Server #1 in Real Server Group #1 fails, the switch places Real Server #1 out of service for that real server group only.**

   Server load balancing on the switch will forward HTTP requests to the next available server, either in that real server group or in another group.
5. **Although the HTTP service on Real Server #1 is down, the server will still be able to respond to FTP requests load balanced to Real Server Group 2.**

Trunking Extended to Six Ports

Trunk groups can provide super-bandwidth, multi-link connections between Alteon WebSystems' WebOS switches or other trunk-capable devices. A "trunk group" is a group of ports that act together, combining their bandwidth to create a single, larger virtual link.

WebOS Release 5.2 and earlier versions of Alteon WebSystems' switch software supported up to four trunk groups per switch, each with 2 to 4 links. In WebOS Release 6.0, trunking support has been extended to a maximum of six ports in a trunk group.

NOTE – The additional two ports are supported only on multi-link connections between Alteon WebSystems' switches.

Prior to configuring the switch for port trunking, you must first connect the switch ports which will comprise the trunk group. Use the options of the Trunk Group Configuration Menu, (`/cfg/trunk/`) to configure trunking parameters.

Weighting for GSLB Sites

When GSLB is configured on a switch running WebOS Release 5.2 and earlier versions of Alteon WebSystems' switch software, the switch will respond to DNS inquiries with a ranked list of sites in the global pool. Ranking is based on geographical region and each site's response time.

In WebOS Release 6.0, Global SLB sites can be "weighted" to give preference to local servers. The `wght` option on the Global SLB Menu (`/cfg/slb/gslb`) is used to set the weighting value (1 to 48). Then, when making GSLB decisions, the local site's response time is divided by the weight value, determining the degree to which the local site should be favored.

By default, the local GSLB site has a weight setting of 1, giving it no preference over remote sites and resulting in classic GSLB behavior. Higher weighting values reduce the adjusted response time value and result in a higher ranking for the local site.

Alternately, since remote sites are configured under the SLB Real Server Menu (`/cfg/slb/real`), remote sites can be weighted using the SLB Real Server Menu's `wght` option. Increasing the remote site's real server weight increases it's standings in the GSLB response. This real server weighting system can be used together with the GSLB weight to tune GSLB performance.

NOTE – The GSLB local site weight is ignored for client requests that originate outside the local geographic region as defined by the IANA field of the client's IP address.

UDP Stateless Load Balancing

A new UDP stateless option has been added to Server Load Balancing:

`/cfg/slb/virt` *virtual_server_ID*`/udp` *UPD_port* **`enable`**|**`disable`**|**`stateless`**

Normally, session time-out is governed using the real server time-out option (`/cfg/slb/real` *real_server_ID*`/tmout`) where the default is 10 minutes. When the `stateless` option is set, UDP sessions time-out immediately, ignoring the real server time-out value. This is useful for quick request/response traffic such as DNS and RADIUS where traffic flows are not required.

NOTE – When the `stateless` option is used, the switch does not record the current number of sessions. As a result, the `leastcons` Server Load Balancing metric (configured under the Real Server Group Menu) cannot be used. Instead, use `roundrobin`, `hash`, or `minmisses`.

Persistent Mask for SLB

A new persistent mask option has been added to Server Load Balancing:

`/cfg/slb/pmask` *IP_mask*

Where *IP_mask* is an IP address mask in dotted decimal notation. The default value is 255.255.255.255 (off).

When a persistent mask is configured, all clients in the masked range will be load balanced to the same real server. This is useful in situations where client proxies are being used.

GSLB Minimum Available Connections

A minimum available connections option has been added to Global Server Load Balancing:

`/cfg/slb/glsb/minco` *minimum_connections*

Where *minimum_connections* is a value between 0 and 65535, with 0 as the default.

When a value is specified, remote global sites will stop redirecting (via HTTP) or handing-off (via DNS) clients to this site once the available connections on this site drops below the configured minimum.

GSLB Real Server Name Redirection

A new real-name options has been added to Global Server Load Balancing:

`/cfg/slb/gslb/usern enable|disable`

When this option is enabled, the real server name and virtual server domain name are used in the HTTP redirect to remote sites. This option is cookie friendly. When disabled, the real server IP address is used instead and may not work with all cookies.

Server Fragment Remap Disabling

A new fragment remap disable command has been added to Server Load Balancing:

`/cfg/slb/virt` *virtual_server_ID*`/frag e|d`

Specifying **`d`** disables remapping server fragments. The default is **`e`** (enabled). This option should be enabled when the switch is expected to load balance UDP applications that generate large UDP datagrams which are fragmented by the servers. When load balancing UDP applications that are generally unfragmented, such as DNS or RADIUS, this option should be disabled.

32-Bit Low/High Access to 64-Bit Counters

New SNMP MIB variables provide 32-bit access to 64-bit counters. All 64-bit counters now have equivalent 32-bit low and 32-bit high counters so that 32-bit SNMP applications can utilize them. These new counters are detailed within the MIB itself.

Addendum, Errata, and Limitations

Defining Host Names for Virtual Services

The following text clarifies material regarding host names that appears in your *WebOS Release 5.2 User's Guide* on pages 7-40, 17-10, and 17-14.

Host names can be assigned to services on virtual servers through the `hname` option on the Virtual Server Configuration Menu (`/cfg/slb/virt`). Host names are used in conjunction with the domain name (also assigned through the Virtual Server Configuration Menu) and provide support to the Global Server Load Balancing (GSLB) system.

For each virtual server, any particular host name can be assigned to only one service. If multiple services under the same virtual server require the same host name (for example "www" for both HTTP and HTTPS services), the host name in question should be defined for only one of those services, and the host name for the other should be left blank. Leaving duplicate host names blank does not harm the efficiency of the GSLB system.

NOTE – If you try to configure duplicate host names for any particular virtual server, the switch will not allow you to apply your configuration changes until the duplicate entries are cleared. You can clear a host name by defining it as "`none`".

Health Checking in Large Server Farms

By default, health checking for each real server is performed every two seconds. When a very large number of real servers (over 250) is connected to the switch, attempting health checks on each real server every two seconds can degrade performance and health checking accuracy. If this becomes a problem, you can increase the time between health checks in order to gain system performance.

The following command is used for changing the health checking interval:

`/cfg/slb/real` *real_server_ID*`/intr` *time_interval*

Where *time_interval* is an integer between 1 and 60 seconds. In large server farms, the recommended value is at least 10 seconds.

Changing Filters in an Active Configuration

If when assigning a new filter to a switch port, the new filter has a higher order of precedence than filters currently firing on the port (the filter number of the new filter is lower than the filter number of existing filters), the new filter will not take effect until the binding table for the switch port is cleared. You can clear the binding table for a given switch port with the following command:

`/oper/slb/clear` *port_number*

NOTE – This command also clears all sessions cached on the specified port.

Restrictions on Direct Access Mode

Direct Access Mode imposes the following restrictions:

- Topology: All frames coming back through the switch server port must egress the switch via the same client port where they were originally received. When DAM is enabled, it can create problems in address re-mapping for networks that have more than one active ingress router.
- VRRP: Sharing should be disabled.
- URL SLB with active-active redundancy: Because of the topology restrictions imposed by Direct Access Mode, you cannot use it to support URL server load balancing with active-active redundancy. To support URL server load balancing with active-active redundancy, you need to set up an IP proxy address.

Remote Gateways Are Not Supported

When setting default gateways on the switch, the gateway device must be on the same IP subnet as one of the switch's IP interfaces. The WebOS switch software does not support default gateways that must be reached remotely through an intermediate router.

Port Trunking with Cisco 3.2.2 Not Supported

When STP is enabled on your WebOS switch, port trunking between the switch and a Cisco Catalyst with version 3.3.2 software is not supported.

TFTP Software Downloads to Active Image

When performing software upgrades using the `/boot/tftp` command, you should not replace the active software image. If booting from `image1`, replace `image2`. Alternately, if you wish to replace `image1`, first boot using `image2`.

Use the `/boot/image` command to change the image from which you wish to boot, and use the `/boot/reset` command to reboot the switch using the specified software image.

Serial Download of Non-Serial Software

Serial download requires a software image designated specifically for serial download. The regular WebOs software files installed via TFTP are not compatible with the serial download method. Serial download of TFTP files will corrupt the switch software bank and require an additional serial download of the proper software.

WebOS Browser-Based Interface

When you configure virtual server IP address ***after*** a VRRP virtual router with the same IP address has been configured, the switch's browser-based interface may occasionally be displayed to a client accessing a virtual server. To prevent this, first configure the virtual server IP address, then the VRRP virtual router.

Listing of Known Bugs and Fixes

Up-to-date information about the status of known software bugs, fixes, and work-arounds for each release of the WebOS switch software is available online. The following URL leads to our main web page:

`http://www.alteon.com`

Follow the "Support" link to the "Field Notices," where you will find the "Known Bug List."

This information is available as part of your support contract and will require you to enter your support access name and password.

Late-Breaking News and Support

Web access: http://www.alteon.com

Questions? Check the URL for Alteon WebSystems. This website includes product information, software updates, release notes, and white papers. The website also includes access to Alteon WebSystems Customer Support for accounts under warranty or that are covered by a maintenance contract.