



W H I T E P A P E R

Optimizing ISP Networks and Services with DNS Redirection

- **DNS REDIRECTION OPERATION AND BENEFITS**
- **LOAD BALANCING ALGORITHMS**
- **DNS SERVER HEALTH MONITORING**
- **OVERFLOW AND BACKUP SERVERS**
- **HIGH APPLICATION AVAILABILITY USING HOT STANDBY ACEDIRECTORS**
- **ACESWITCH/ACEDIRECTOR ARCHITECTURE**

Alteon WebSystems, Inc.

50 Great Oaks Boulevard
San Jose, California 95119
408-360-5500
408-360-5501 fax

<http://www.alteon.com>

Help desk costs for Internet Service Providers (ISPs) are skyrocketing. In fact, for many ISPs, help desk costs rival, or even exceed, WAN bandwidth costs as the single biggest expense item.

As ISPs consolidate, larger numbers of subscribers come online and the profile of the typical subscriber changes from technologically sophisticated to technologically challenged, ISPs are faced with an uphill battle. This battle consists of automating end user connectivity regardless of geographic location, optimizing infrastructure performance and the management and configuration of end user systems.

Many calls to ISP help desks have their roots in Domain Name Services (DNS) problems. DNS is a distributed database service that provides the mapping between IP addresses and hostnames.

Proper network operation requires that subscribers' computers be configured with the correct DNS server address. A DNS server address that is mis-configured, because the subscriber made a mistake entering it, the entry was accidentally changed or any other reason, will result in the subscribers losing Internet connectivity and, moments later, a call to the ISP's help desk.

Further, the need to configure subscribers' computers with the correct DNS server address can hamper an ISP's ability to implement DNS changes needed to keep up with subscriber demand. For example, an ISP may want to move from a centralized DNS server to a number of decentralized DNS servers to meet the needs of a growing user base.

Ideally, with decentralized DNS servers, subscribers would access the closest DNS server. However, because DNS servers in a decentralized architecture have unique IP addresses, this is not possible unless each subscriber reconfigures the DNS server address on their computer.

For any ISP, getting users to change the DNS server address on their computer can be an arduous and time-consuming task, fraught with potential problems. Still more difficult, is dealing with the ensuing help desk calls that result from subscribers mis-configuring their DNS server address during such a change.

Though the Dynamic Host Control Protocol (DHCP) may alleviate some of these problems, ISPs would have to convert all of their users to this automatic addressing solution.

DNS redirection, implemented on Web switches, can eliminate these problems. Web switches are a new and special class of LAN switch that front-ends individual servers or server farms, providing customized services that allow for increased scalability, availability and better server efficiency. The ability to load balance or redirect server-bound traffic is one such value-added service.

By redirecting DNS traffic, all DNS requests are directed to the DNS server of the ISP's choosing, regardless of where the requests are addressed. If a subscriber's computer uses an erroneous DNS server address for any reason, it doesn't matter. DNS requests are still directed to the DNS server chosen by the ISP and the subscriber's network connectivity is maintained—with no calls to the ISP's help desk.

Additionally, an ISP may want a subscriber to use a DNS server other than the one for which the subscriber's computer is configured, even if the configured address is correct for their local Point of Presence (PoP). A good example of this situation is mobile subscribers who have ventured beyond their home area when their ISP has implemented a distributed DNS architecture.

Without DNS redirection, DNS requests are sent to the user's home PoP's DNS server instead of to the DNS server for the PoP into which they've dialed. Again, DNS redirection comes to the rescue because the DNS server address used by the subscriber is ignored and DNS requests are directed to the DNS server designated by the ISP.

Finally, DNS redirection allows DNS requests to be dynamically spread over multiple DNS servers in a server farm. This scales processing power, reduces response time and increases DNS availability.

DNS REDIRECTION OPERATION AND BENEFITS

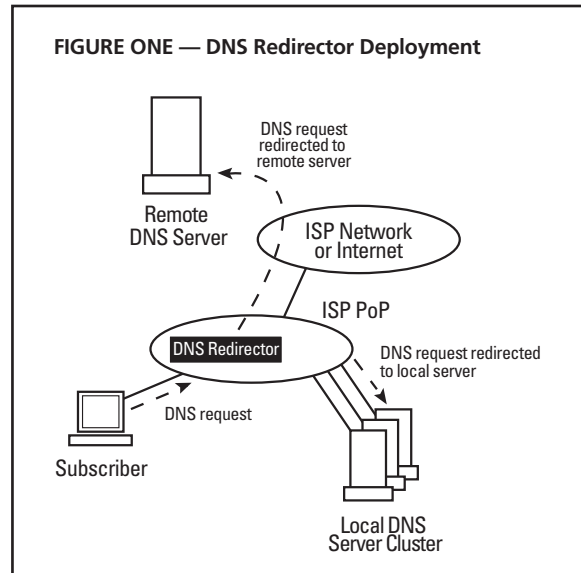
DNS redirection runs on Web switches such as the Alteon WebSystems' ACEswitch or ACEdirector products.

As shown in Figure 1, the Web switch sits in the data path between subscribers' computers and the Internet, generally in an ISP's PoP. The Web switch examines each packet, determining which are DNS requests. Using Layer 2 or Layer 3 switching, packets not identified as DNS requests are subsequently forwarded to their ultimate destination. DNS packets are intercepted and automatically redirected. For redirected packets, Web switches perform the required network address translations and send them to the DNS server specified by the ISP. This DNS server may be located locally, in the ISP's PoP, or at a remote location. The DNS server can also be a single device or a virtual DNS server created by a DNS server farm.

When multiple DNS servers are used in conjunction with DNS redirection, Web switches distribute DNS requests across the servers in the server farm based on a pre-configured algorithm. Choices for this algorithm are round-robin, least connections, hashing and minimum misses.

The Web switch performs health checks on configured DNS servers, sending DNS requests only to DNS servers that have passed these health checks. Redundant Web switches can also be deployed to eliminate any single point of failure in the system, yielding the ultimate in uptime.

When deploying DNS redirection, a Web switch must be topologically situated such that subscribers' DNS requests pass through it before going to the Internet. This allows for the interception and redirection of all DNS requests. If the network topology permits DNS requests to bypass the Web switch, they may be switched or routed around it and will not be redirected to the desired DNS server.



Fixing Mis-configured DNS Server Addresses

When a subscriber mis-configures the DNS server address in their computer, their DNS requests will either be sent to a system that is not a DNS server or dropped. In either case, they will not receive any responses to their DNS requests, will not be able to obtain IP addresses for Internet hostnames, thereby losing Internet connectivity (see Figure 2).

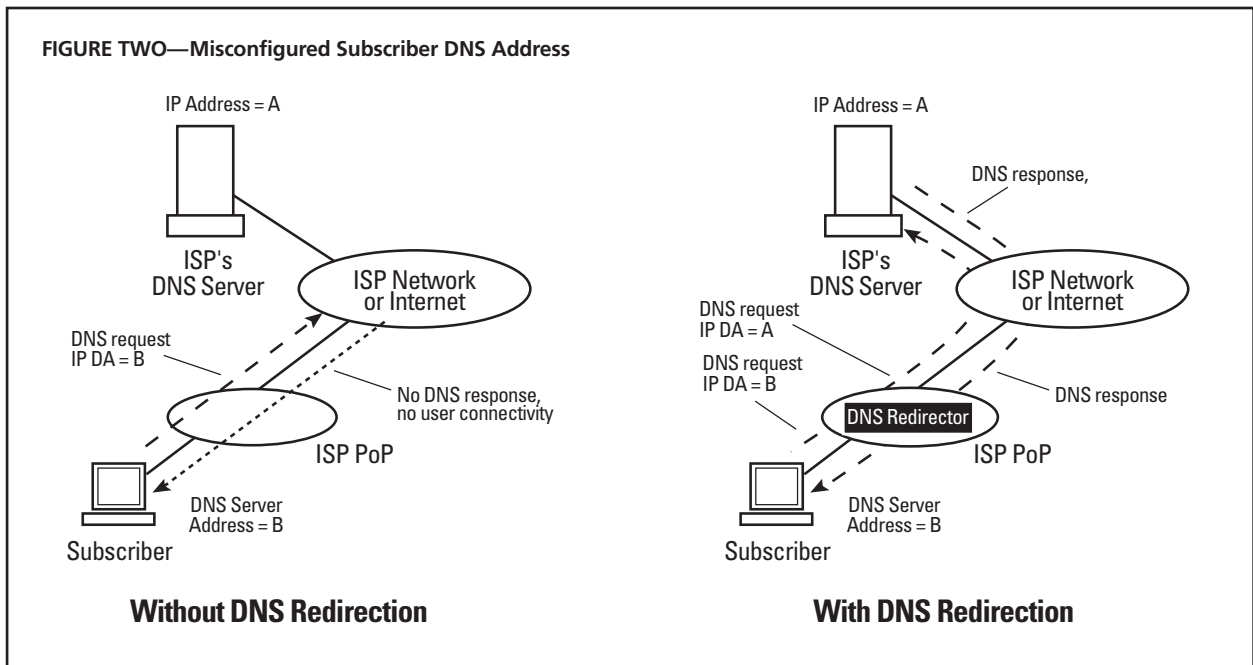
With DNS redirection, the DNS server address configured in subscriber's computer doesn't matter. The Web switch intercepts all DNS requests, regardless of the destination IP address and sends these requests to the DNS server designated by the ISP.

The Web switch also manipulates the DNS responses so they appear as though they came from the address configured for the subscriber's DNS server. Doing this, users' Internet connectivity is maintained, no matter how DNS server address have been configured on their computer.

Implementing a Distributed DNS Architecture

As subscriber bases grow, many ISPs who have implemented a centralized DNS server architecture see advantages in distributing the DNS function.

For a large subscriber base, a distributed architecture with many DNS servers - each located close to part of the subscriber population - offers faster, more efficient response than having a single centralized DNS server. Fast DNS responses are key to maintaining high application performance and subscriber satisfaction. Until the DNS function is completed, connectivity to the end system housing the application cannot take place. As a result, users must wait for DNS transactions to be completed until applications can be accessed.



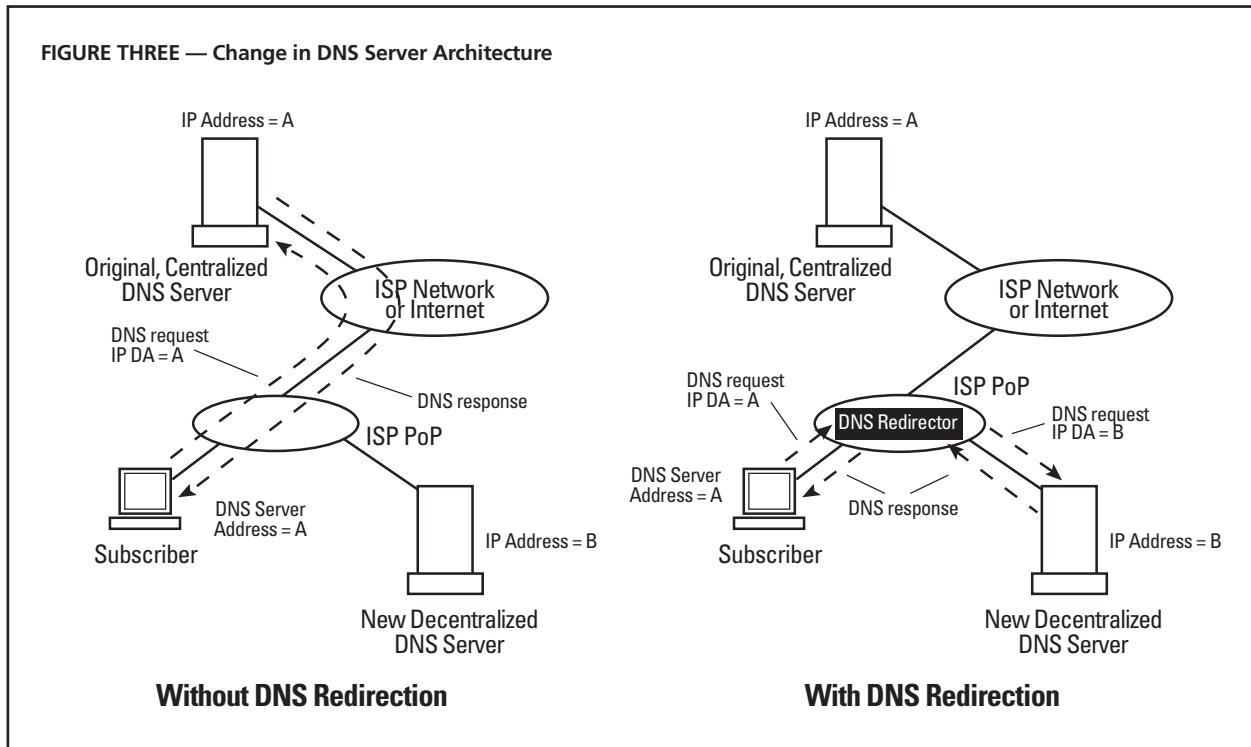
Further, a distributed DNS architecture allows ISPs to better manage their Internet traffic. For example, many portals support mirror sites across the country or world. ISPs using distributed DNS servers may configure those servers to direct traffic to specific mirror sites located close-by.

But moving to a distributed DNS server architecture brings the ISP face-to-face with some harsh realities. If they cannot get subscribers to reconfigure the DNS server address on their computers, the move to a distributed DNS server architecture will offer no benefits because subscribers will continue to send requests to the original, central DNS server (see Figure 3). Conversely, if the ISP gets its subscribers to change the DNS server address on their computers, a significant percentage of them will make mistakes, and the ISP is then faced with the mis-configured DNS server address problems described above.

DNS redirection resolves this dilemma because the DNS server address configured in subscribers' computers doesn't matter. All DNS requests are intercepted by the Web switch and sent to the desired DNS server.

In fact, with DNS redirection, subscribers retain the benefit of keeping the address of the original, central DNS server configured in their computers.

If the DNS server or all servers in the DNS server farm to which the Web switch sends DNS requests fail, the Web switch sends DNS requests to the host identified in the destination IP address specified by the subscriber's computer. If this address is the original, central DNS server, then the DNS request will be directed to that server, providing a backup to the failed server or server farm.



Coping with Mobile Subscribers

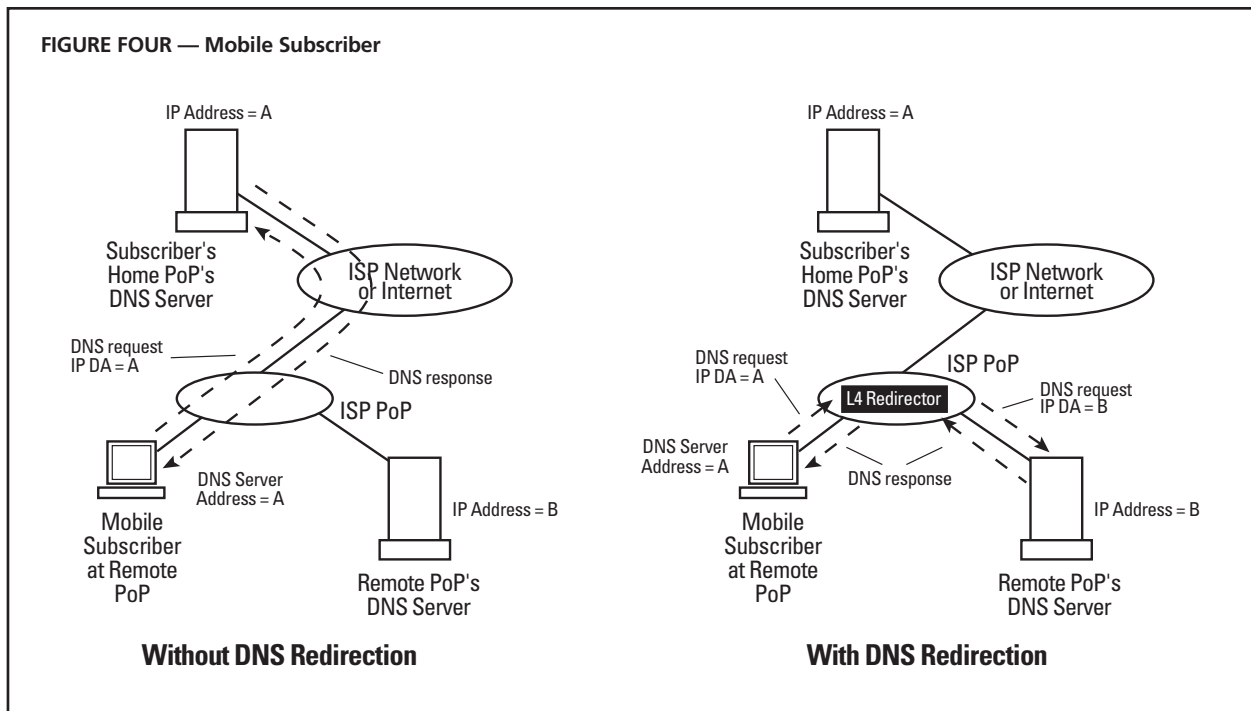
In distributed DNS server environments where all subscribers have their local DNS server address properly configured on their computers, mobile subscribers present a significant challenge. For a mobile subscriber, the DNS server that is local to their home PoP may not be local to the PoP they dial into away from home. While it's desirable for a mobile subscriber to use the DNS server local to the PoP into which they dial, that's not where their requests will go.

Instead, DNS requests will go to the subscriber's home PoP's DNS server, slowing responses (see Figure 4). As previously noted, fast DNS responses are key to maintaining high application performance and subscriber satisfaction.

With DNS redirection this problem is solved. A mobile subscriber's DNS requests are intercepted by a Web switch at each PoP and can be automatically redirected to the closest DNS server. When the subscriber uses a different PoP, they'll use a different DNS server, if that's how the ISP has configured the DNS service.

What's at stake here is not simply where DNS requests go but also how other traffic is handled. As noted above, portals may support mirror sites and ISPs may configure distributed DNS servers to direct traffic to local mirror sites. A mobile subscriber away from their home PoP but using their home PoP's DNS server will not send traffic to the best mirror site for their current location. If this happens, application performance will suffer.

Using DNS redirection to send the subscriber's requests to the DNS server for the PoP they're currently using eliminates this problem.



Increasing DNS Availability

Subscribers must be able to access a DNS server to resolve Internet hostnames to IP addresses. If a subscriber can't access a DNS server, they can't access the Internet. Because of this, maintaining high DNS server availability is key to maintaining high subscriber satisfaction (see Figure 5).

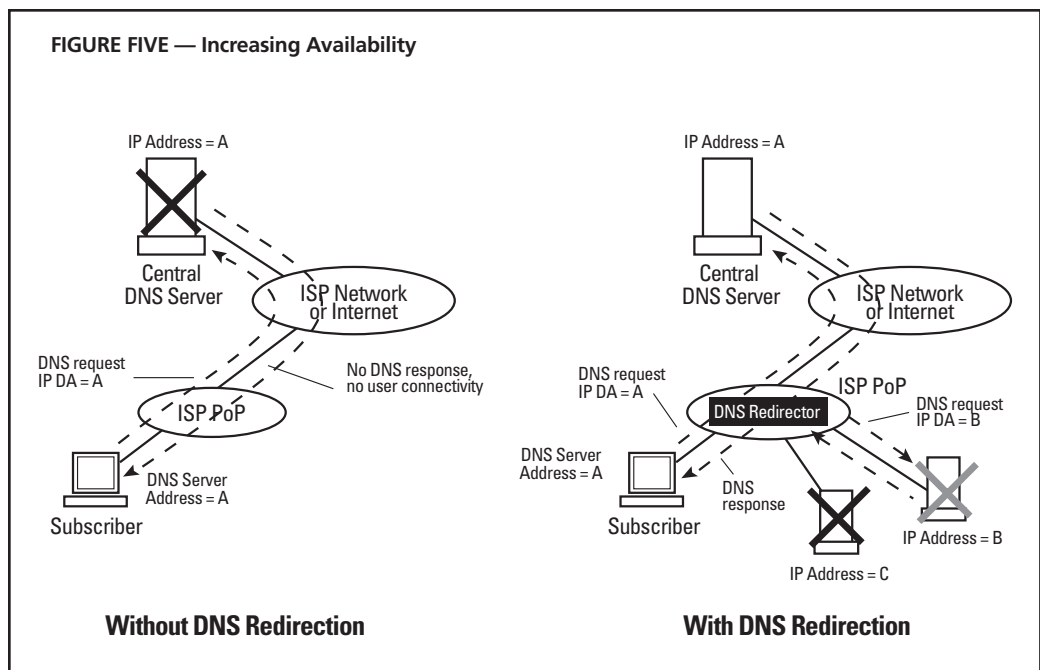
DNS redirection can increase DNS server availability. An ISP can deploy multiple DNS servers and configure Web switches to distribute DNS requests across the servers in the server farm.

The Web switch will perform health checks on the DNS servers and will send DNS requests only to DNS servers that pass the health checks. For example, in Figure 5, if the DNS server with IP address C fails, as represented by the black X, all DNS requests will go to the DNS server with IP address B.

The ISP can optionally configure the Web switch to send DNS requests to a backup server if any server in the server farm fails or if all of the servers in the server farm fail. For example, in a distributed DNS architecture, one PoP's DNS server(s) can both support local subscribers and act as backup for other PoPs.

As a last resort, if the configured DNS server or all configured DNS servers in a server farm fail and no backup server is available or configured, the Web switch will send the DNS request to the specified host using the destination IP address used by the subscriber's computer. ISPs who want maximum availability should have their subscriber's computers to use the address of a central DNS server that can be used as a fail-safe in the event that all other DNS servers fail. If they do, then the Web switch's last resort will result in subscribers' DNS requests being directed to the fail-safe DNS server.

This is shown in Figure 5, where both DNS servers in the local PoP have failed (as represented by the black X and gray X). In this case, DNS requests will be sent to the central server with IP address A.



LOAD BALANCING ALGORITHMS

When multiple servers are used to form a DNS server farm, the Web switch directs each DNS request to a specific server based on an ISP-configurable load-balancing algorithm. The ISP has the choice of round-robin, weighted round-robin, least connections, weighted least connections, hashing or minimum misses for this algorithm. In addition, weighting and a maximum-connections threshold can be configured for any DNS servers.

Round Robin

With this algorithm, new connection requests are forwarded to the DNS servers in a round robin fashion such that, over time, each DNS server in the server farm gets the same number of connection requests. This doesn't mean that each DNS server will have the same number of active connections – some servers will close connections faster than others will.

Weighted Round Robin

Weighted round robin load balancing is similar to round robin load balancing, but each server in the server farm is assigned a static weight based on some view of its capacity. The Web switch presents connection requests to servers in proportion to their weighting.

Least Connections

When this load-balancing method is selected, the number of active connections each DNS server is handling is tracked. As requests for new connections are received, they are forwarded to the DNS server in the server farm with the fewest active connections.

Weighted Least Connections

In weighted least connections, a weighting function is added to the least connections policy. The Web switch normalizes the number of connections to each server based on the server's static weighting. It directs DNS requests to the server with the fewest active normalized (as opposed to actual) connections.

Hashing

Hashing uses a mathematical algorithm to manipulate DNS requests' source and destination IP addresses to assign them to servers. Hashing has the property of assigning DNS requests from the same subscriber to the same DNS server.

Minimum Misses

Minimum misses also manipulates DNS requests' source and destination IP addresses to assign them to servers and assigns DNS requests from the same subscriber to the same DNS server. While hashing will generally offer better instantaneous distribution of load than minimum misses, minimum misses will suffer from less perturbation than hashing when DNS servers are added to or removed from the load balancing algorithm.

Choosing a Load-Balancing Metric

The choice of load-balancing metrics is not crucial to DNS redirection operation. In general, it is probably best to use hashing as it offers good instantaneous load distribution and directs requests from a given subscriber to the same DNS server. Directing requests to the same DNS server may offer some response time advantages because requests may be more likely be serviced directly from the local server, without it needing to make queries from servers higher in the DNS hierarchy.

In environments where only one switch redirects requests to the DNS servers, users may also want to weigh the advantages and disadvantages of using least connections.

Maximum Connections Option

The maximum connections option can be configured with any of the load balancing policies described above. It allows users to set the maximum number of active connections to be assigned to a particular DNS server. When a server reaches its maximum connections limit, the Web switch will not send any more DNS requests to it until it drops back below its maximum connections limit.

If all of the servers in the server farm reach their maximum connections limit, the Web switch will send DNS requests to the Internet using the destination IP address used by subscribers' computers until at least one server drops below its limit (there are exceptions to this statement, see Overflow and Backup Servers, below).

DNS SERVER HEALTH MONITORING

As earlier noted, DNS server availability is crucial to ensuring that ISP subscribers have continuous network connectivity. To facilitate this goal, Web switches monitor the health of DNS servers and direct packets only to healthy DNS servers.

DNS Request-Based Health Monitoring

Web switches monitor the health of DNS servers by sending requests to each DNS server in the server farm on a regular basis. These requests identify both failed servers and failed DNS services on a healthy server.

If DNS request-based testing indicates a failure, the Web switch places the DNS server in the "Server Failed" state. At this time, the Web switch stops redirecting DNS requests to the server, distributing them across the remaining healthy servers in the DNS server farm. If all servers in the DNS server farm are unavailable, and no backup server is configured and available, the Web switch sends DNS requests to the server to which the requests are addressed.

When a DNS server is no longer in the "Server Failed" state, the Web switch begins sending DNS requests to it. Once the Web switch receives a DNS response from server, it brings the previously failed DNS server back into service.

Physical Connection Monitoring

Web switches also monitor the physical link status of switch ports connected to DNS servers. If the physical link to a server goes down, the switch immediately places that server in the "Server Failed" state, taking the same actions as if the DNS server had failed request-based monitoring. When the Web switch detects that a failed physical link to a DNS server has been restored, it brings the server back into action in the same manner described earlier.

OVERFLOW AND BACKUP SERVERS

Web switches can also be configured to introduce overflow servers when any or all of the servers in a server farm hit their maximum connections threshold. It can also bring in backup servers when it detects that any or all of the servers in a server farm have suffered from a service or physical link failure.

Overflow Servers

As discussed previously, a maximum connections limit can be configured for a DNS server. When the number of active connections to a DNS server reaches the maximum connections limit, the Web switch will not send any additional connection requests to that server until its number of active connections falls below its maximum connections limit. Overflow servers can take over temporarily.

The idea behind overflow servers is that a server that does not normally supply DNS to a particular PoP can be configured to do so under special circumstances. Under normal circumstances, the Web switch will not forward any DNS requests to this server.

However, if one or all of the DNS servers in the server farm hit their maximum connections limits, the Web switch will introduce the overflow server into the load-balancing mix. When the load on the DNS server(s) falls to an appropriate level, the Web switch will remove the overflow server from the load balancing mix by ceasing to direct DNS requests to it.

Overflow servers may be configured to become active when an individual DNS server or when all of the DNS servers in the server farm exceed their maximum connections thresholds.

As it is called upon to process connections from the Web switch, the overflow server continues to support its normal functions. The intent is to have the overflow server temporarily add capacity to the hunt group.

If an overflow server is recruited into service too frequently and for prolonged periods of time, it may signal a need to increase the permanent capacity of the server farm by adding more servers.

Backup Servers

Application, server or link failure can cause Web switches to remove a server from the load-balancing mix.

Users may configure the Web switch to introduce a backup server into the load-balancing mix when any or all of the servers within a server farm fail.

In the case where the switch introduces the backup server into the load-balancing mix after the removal of a single server, the backup server is taken out of the mix when the Web switch determines that the server is again operational.

In the case where the Web switch introduces the backup server into the mix after the removal of all servers in a server farm, the backup server is taken out of the load-balancing mix when the Web switch determines that all servers in the server farm are once again operational.

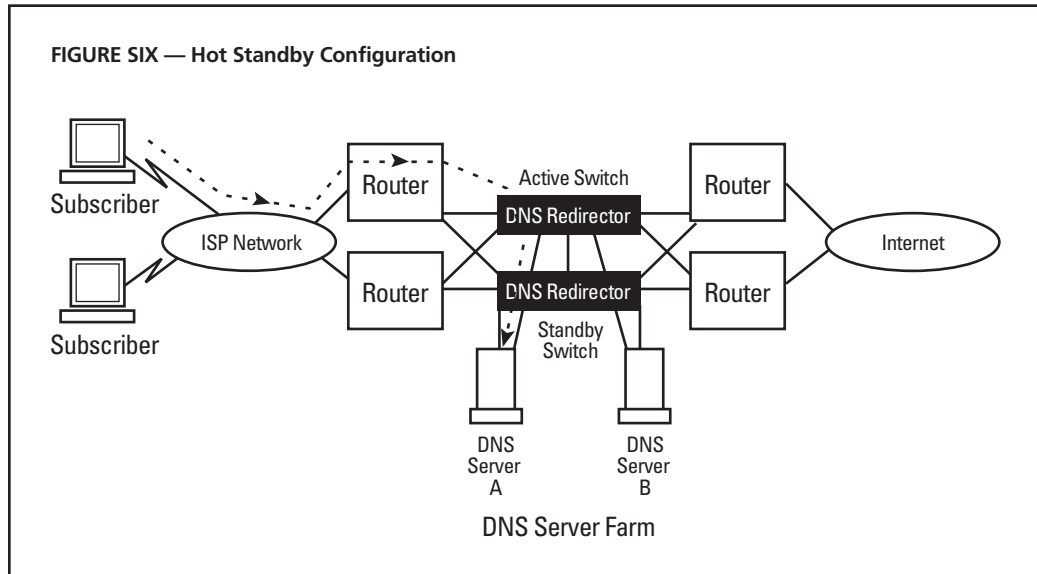
As with overflow servers, taking a backup server out of the mix means that the Web switch stops directing DNS requests to it.

HIGH APPLICATION AVAILABILITY USING HOT STANDBY WEB SWITCHES

Beyond DNS server health monitoring, even higher levels of application availability can be achieved by using hot standby Web switches.

Web switches can be used in pairs with one active and the other in hot standby mode to build network topologies with no system-wide single point of failure.

This means that the Web switches are not single points of failure and that their use does not force a single point of failure at some other point in the network. Eliminating single points of failure increases application availability. An example of a hot standby configuration is shown in Figure 6.



This topology supports use of active and standby Web switches, redundant network devices and redundant, load-balancing DNS servers. Note that no traffic flows through the standby Web switch. For example, all traffic from client 1 to DNS server A flows through the active Web switch only, as shown by the arrows.

When implementing a hot standby configuration, one of the Web switches is designated as the active switch, the other as the standby switch.

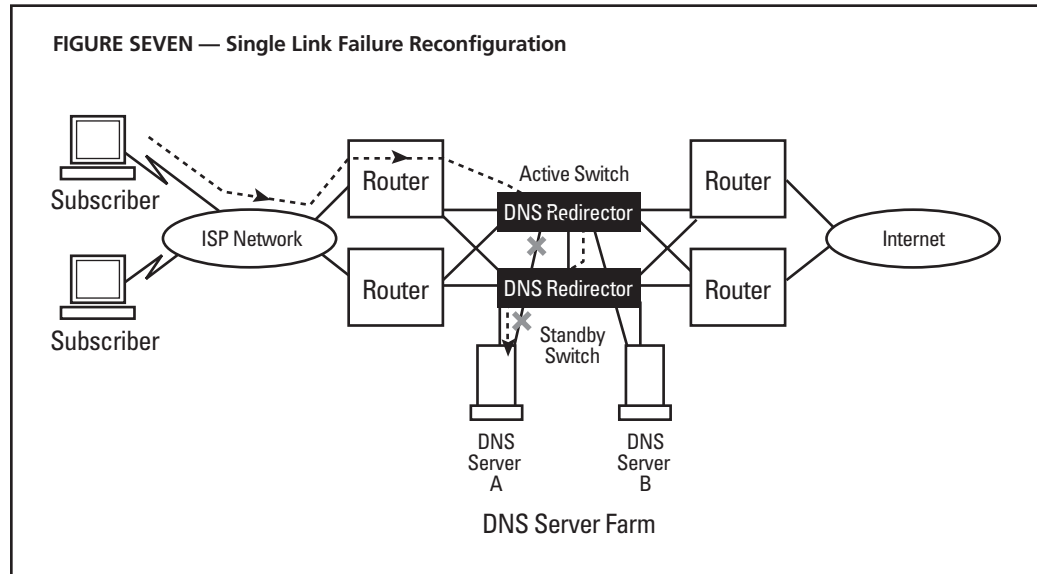
A direct link, known as the failover link, is configured between the active Web switch and the standby Web switch. The failover link is used to send keep-alive messages between the active and standby Web switches. Data traffic crosses the failover link only in the event that a port on the active switch has failed.

If the active Web switch detects a link failure, it communicates that information to the standby Web switch via the failover link. If the corresponding port on the standby Web switch is healthy, that port is activated.

For example, Figure 7 shows the case where the link from the active Web switch to DNS server A has failed. In this case, the port on the standby ACEdirector that connects to DNS server A becomes active. All traffic between client 1 and DNS server A now passes through the active Web switch, crosses the failover link, goes through the standby Web switch and traverses the newly activated link, as shown in the figure.

Where an entire active Web switch fails, the standby Web switch becomes active, as shown in Figure 8. After the router on the upper left determines (using OSPF, for example) that the originally active Web switch has failed and that the standby has become active, traffic will again flow from client 1 to DNS server A, as shown.

As noted earlier, the failover link is used to send keep-alive messages between the active and standby Web switches. If a Web switch fails to receive keep-alive messages from its counterpart, it may be an indication that the failover link has failed or that its counterpart has failed.



It is important to distinguish between these two cases. If the failover link has failed but both Web switches are healthy, this must be determined to avoid the “split brain” problem where the original standby Web switch attempts to become active while the original active Web switch is still active—a situation that could disrupt communications.

Conversely, if the standby Web switch stops receiving keep-alive messages because the original active Web switch actually has failed, it must determine this so it can become active.

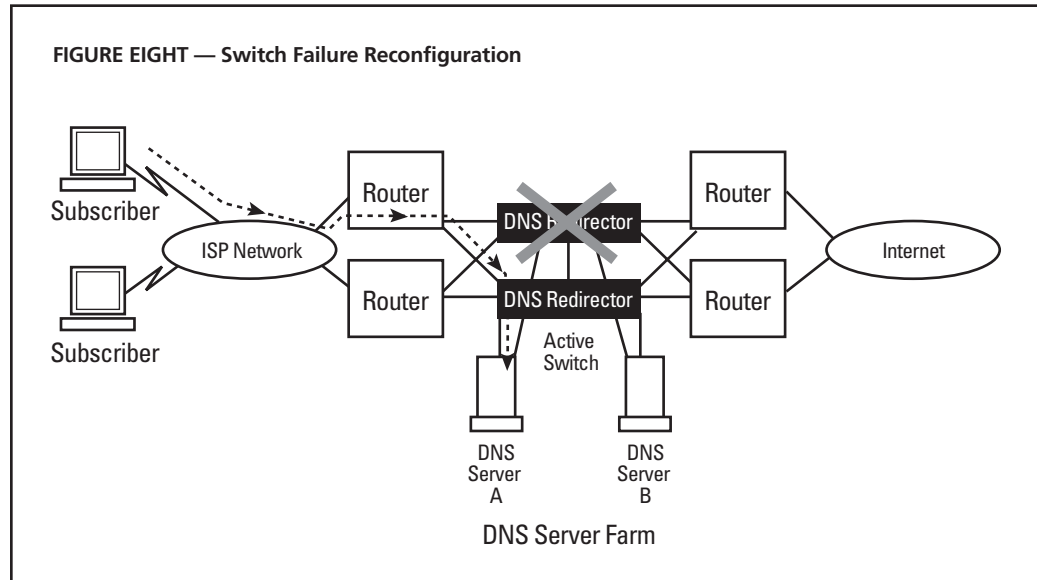
Distinguishing between the two cases is accomplished by combining Physical and Data Link Layer health checks used on all Web switch ports along with specialized messaging.

ACESWITCH/ACEDIRECTOR ARCHITECTURE

Alteon WebSystems pioneered the concept of Web switching. The fundamental idea behind Web switching is to optimize the interface between the network and the servers. Web switches provide high-performance Layer 2, 3 and 4-based switching services to attached devices. This multilayer switching functionality is combined with sophisticated redirection features and value-added packet processing functionality such as packet filtering and server load balancing.

Web switches scale application performance by acting as an intelligent front-end processor for server farms. This enables applications to run across multiple servers in a manner that is transparent to clients. It also improves application availability by ensuring that client requests are directed only to healthy servers.

DNS redirection is a perfect example of a Web switching function. Web switches running DNS redirection act as intelligent front-end processors to DNS server farms, providing network connectivity and intelligently directing DNS requests to servers in the DNS server farm.

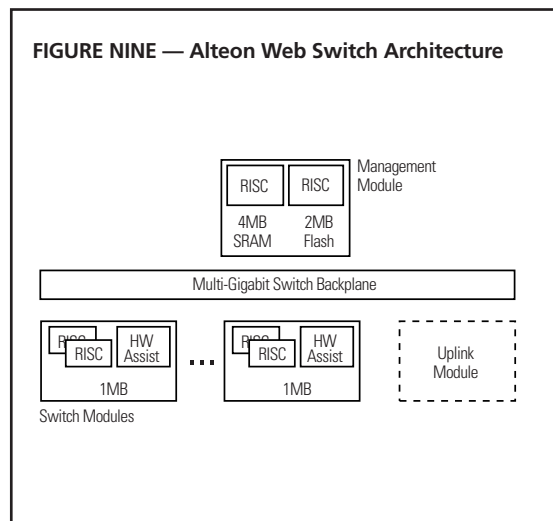


Examining thousands or tens of thousands of packets per second, determining which must be forwarded normally, which should be redirected to a DNS server (and to which DNS server, if they are clustered) and performing the necessary network address translation requires vast amounts of processing capacity and memory. Every incoming packet must be examined to determine if it is a DNS request. Packets that are not DNS requests must be sent toward their destinations using Layer 2 or Layer 3 switching.

If clustered DNS servers are used, the configured distribution algorithm must be executed for each DNS request to determine to which DNS server in the server farm the packet should be sent.

After this determination is made, the packet is manipulated so that the proper DNS server will receive it. Additional background processing is also necessary to perform tasks such as checking the health of the DNS servers, exchanging keep-alive messages when hot-standby switches are used and collecting and reporting statistics for network management.

Alteon's distributed processing Web switch architecture is ideally designed for processor-intensive packet examination and manipulation. Each port on these devices integrates a switching ASIC that comprises a hardware-assisted forwarding engine and dual, 90-MHz RISC processors. Two additional RISC processors provide support for switch-wide management functions (see Figure 9).



On each switch port, the processors in the switching ASIC handle packet examination, specified forwarding at Layer 2, 3 or 4, execution of the distribution algorithm and address substitution.

Background tasks such as DNS server health checking, keep-alive message exchange, and statistics gathering and reporting are handled by the central management processors. With this architecture, processing tasks for each session are distributed to different processors for parallel operations, increasing performance.

SUMMARY

DNS redirection implemented on Alteon WebSystems' Web switching products provides valuable benefits to ISPs and their customers.

It can improve DNS server availability, flexibility and response time. This will lead to improved subscriber satisfaction and lower support costs for ISPs.

By enabling distributed DNS server architectures, DNS redirection can also allow ISPs to direct subscribers to the closest site for mirrored content, reducing response times and Internet bandwidth usage.