Data Center: Load Balancing Data Center Services SRND
Solutions Reference Nework Design
March, 2004

Corporate Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
     800 553-NETS (6387)
Fax: 408 526-4100

Customer Order Number: 956980
Preface

Target Audience

Document Organization

Document Conventions

CHAPTER 1

Application Optimization and Load Balancing

Introduction to Application Optimization

The Benefits of Application Optimization

Information Technology Initiatives

Data Center and Server Consolidation

Application and Infrastructure Architectures

Achieving Application SLAs

Web and N-tier Migration

Content Switching Design

Scaling DNS Services

Scaling Radius Services

Data Center Networking Architecture

Network Infrastructure

Cisco Storage Networking

Data Center Security

Business Continuance Networking

CHAPTER 2

Scaling Server Farms: Content Switching Design

Fundamental Challenges to Scaling Server Farms

Benefits of Content Switching

What is the CSM?

CSM Requirements

Design Details

Data Center Network Infrastructure

Content Switching Design Goals

Interoperability

Scalability

High Availability

Performance
Preface

This Solution Reference Network Design (SRND) provides a description of the design issues related to optimizing server and application environments. Server farms are built to provide a highly available, scalable infrastructure that on which applications run. As enterprises grow, so do their application needs, which implies that more and more expenditures are allotted for additional application-specific software and server requirements. This increase in bottom line expenses heightens the sensitivity to overall return on investments. Cisco provides networking solutions to mitigate the impact of these expenses by offering network-based services that enhance the overall security, manageability, scalability, and high availability of these environments. Many of these services reside in the data center environment, hence the collection of topics found in this SRND.

Target Audience

This publication provides solution guidelines for enterprises implementing Data Centers with Cisco devices. The intended audiences for this design guide include network architects, network managers, and others concerned with the implementation of secure Data Center solutions, including:

- Cisco sales and support engineers
- Cisco partners
- Cisco customers

Document Organization

This document contains the following chapters:

<table>
<thead>
<tr>
<th>Chapter or Appendix</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chapter 1, “Application Optimization and Load Balancing”</td>
<td>Provides an overview of the benefits of optimizing your server and application environments.</td>
</tr>
<tr>
<td>Chapter 2, “Scaling Server Farms: Content Switching Design”</td>
<td>This design chapter provides guidance on how to scale your server farm using the content switching module.</td>
</tr>
<tr>
<td>Chapter 3, “Scaling DNS Services”</td>
<td>Describes how to use load balancing to ensure availability and improve performance for DNS servers.</td>
</tr>
<tr>
<td>Chapter 4, “RADIUS Server Load Balancing”</td>
<td>Describes how to use load balancing to ensure availability and improve performance for RADIUS servers.</td>
</tr>
</tbody>
</table>
Document Conventions

This guide uses the following conventions to convey instructions and information:

Table 1  Document Conventions

<table>
<thead>
<tr>
<th>Convention</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>boldface font</strong></td>
<td>Commands and keywords.</td>
</tr>
<tr>
<td><strong>italic font</strong></td>
<td>Variables for which you supply values.</td>
</tr>
<tr>
<td>[ ]</td>
<td>Keywords or arguments that appear within square brackets are optional.</td>
</tr>
<tr>
<td>{x</td>
<td>y</td>
</tr>
<tr>
<td><strong>screen font</strong></td>
<td>Examples of information displayed on the screen.</td>
</tr>
<tr>
<td><strong>boldface screen font</strong></td>
<td>Examples of information you must enter.</td>
</tr>
<tr>
<td>&lt; &gt;</td>
<td>Nonprinting characters, for example passwords, appear in angle brackets.</td>
</tr>
<tr>
<td>[ ]</td>
<td>Default responses to system prompts appear in square brackets.</td>
</tr>
</tbody>
</table>
Application Optimization and Load Balancing

This paper describes the benefits of application optimization and summarizes the way load balancing in the data center can help to achieve these benefits. It includes the following sections:

- Introduction to Application Optimization, page 1-1
- Content Switching Design, page 1-4
- Scaling DNS Services, page 1-5
- Scaling Radius Services, page 1-6
- Data Center Networking Architecture, page 1-7

Introduction to Application Optimization

Application optimization is one of the key solution areas within the Data Center Networking Architecture. The objective of application optimization is to ensure high performance and high availability for applications running in the enterprise data center. Optimization increases application availability and scalability using intelligent application-aware network technology. Figure 1-1 illustrates how application optimization fits into the Data Center Networking Architecture and identifies some of the key component technologies used.

The network technologies that help optimize application performance include the following:

- Caching improves application response time.
- Content switching and load balancing to consolidate applications and increase application scalability and availability.
- Secure Sockets Layer (SSL) offloading allows servers to increase the number of SSL transactions supported.
Caching, and in particular reverse proxy caching, offloads serving static content from servers, which increases scalability of the server farm. This process of offloading occurs transparently for both the user and the server farm.

Content switching optimizes application services by providing a front-end device that balances incoming requests to the available servers. You can use the built-in Cisco IOS software Server Load Balancing (SLB) feature or use the Cisco Content Switching Module (CSM). CSM can balance requests based on Layer 4 or Layer 5 information, and this allows you to partition server farms based on content.

SSL offloading assigns the processing of SSL traffic to a dedicated device. This increases the scalability of servers, allows the centralized management of SSL services on a single device, and allows content switches to load balance SSL-encrypted traffic after it has been decrypted into clear text.

### The Benefits of Application Optimization

Application optimization helps to achieve the overall business goals of the Data Center Networking Architecture, which are:

- Lower total cost of ownership (TCO)
- Business resilience
- Business responsiveness

Application optimization reduces TCO by enabling server consolidation and supporting the migration to web-based applications. Optimizing your applications for high availability improves business resilience by improving the application infrastructure and helping your organization meet its service-level agreements (SLAs). Optimizing your applications to run efficiently improves the responsiveness of your data center, future-proofing it from changes in technology or increased demands for performance.
Information Technology Initiatives

The information technology initiatives that help applications run efficiently and without interruption in the data center include the following:

- Data center and server consolidation
- Application and infrastructure architectures
- Achieving application service level agreements (SLAs)
- Web and N-tier migration

Data Center and Server Consolidation

Data center and server consolidation reduces operational costs, decreases application complexity, and helps to centralize your computing environment. It requires standardizing server platforms, increasing the number of servers, and upgrading the CPU capacity and memory of your existing servers. Your data center network must move toward a homogeneous application environment and improve the manageability of distributed server farms. Application optimization provides important tools to help achieve these objectives, including the following:

- Application-aware network services
- Application scalability

Application-aware network services include IOS Server Load Balancing (SLB), content caching, and Secure Sockets Layer (SSL) offloading. SLB and SSL offloading also help applications become scalable.

Application and Infrastructure Architectures

Improving your application and infrastructure architecture is a strategic investment rather than a tactical investment, which helps you get control of the total cost of ownership. A well-designed data center infrastructure protects your investment, provides flexibility to meet future needs, and improves utilization. Part of a good infrastructure is a standardized application environment that simplifies application development, deployment and maintenance, and enables integration with web-based and other innovative applications.

Your infrastructure can be defined by vendor initiatives, such as HP UDC, IBM On-Demand, or Sun N1, or based on standards like Java/J2EE. You may implement architecture implementations from specific vendors, such as Oracle 9iAS, .NET, Web Logic, or WebSphere, or have a infrastructure custom designed for your specific requirements. Whatever your network architecture, application optimization improves its flexibility and efficiency and provides application-aware networking capabilities.

Achieving Application SLAs

Achieving application SLAs is a goal that includes uninterrupted application availability, improving service levels, and incorporating operational best practices. It requires developing processes, expertise, and tools to achieve operational excellence, and consistent management across functional areas. Optimizing applications for high availability and load balancing eliminates single points of failure and improves manageability of services.
Web and N-tier Migration

Web and N-tier migration refers to the ongoing evolution of the enterprise application environment to increase end-user productivity and lower application complexity. It creates separate web, application, and database tiers and relies on a standard thin client (web browser) and a standard server (web server), which provides independence from changes in client/server operating systems. To make this work, you need intra-tier scalability, security, availability, and awareness. Application optimization helps ensure scalability through providing tools and designs for balancing and offloading services, such as SLB, SSL offloading, and content caching.

Content Switching Design

Content switching maximizes the efficiency of individual servers residing in a server farm, which increases the overall performance and resiliency of application services. The Cisco Content Switching Module (CSM) provides a high-performance modular solution for scaling Layer 4 and Layer 5 data center server farm services.

For instance, you can partition a video streaming server farm into servers supporting MPEG and servers supporting Quicktime or Windows Media. The CSM determines the type of request by inspecting the URL and then forwards the request to the appropriate server. A content switch can also represent multiple servers to DNS by a single IP address. When the content switch receives a client request, it can use a variety of load balancing methods, including round robin. The content switch also performs server and application health checks before forwarding client requests to the server. This is shown in Figure 2.

![Figure 1-2 Content Switches Provide Load Balancing and High Availability](image)

The CSM is a content switch on a blade available for the Catalyst 6500 or 7600 platforms. It is capable of maintaining multiple Gigabit throughput rates (4 Gbps maximum inbound and outbound) while providing Layer 4-7 inspection and load balancing services. The CSM provides performance for approximately 75,000–80,000 cps (connections per second) when configured to function at Layer 5, and approximately 125,000 cps when configured to function at Layer 4.
The CSM requires the use of Native IOS in the Catalyst switch and does not function with CatOS. Native IOS requires the presence of a MultiLayer Switch Feature Card (MSFC) on the supervisor module. The following table presents the software and hardware recommendations for the CSM and the Catalyst switch.

<table>
<thead>
<tr>
<th>CSM Software Release</th>
<th>Hardware Part Number</th>
<th>Software Part Number</th>
<th>Hardware Requirements</th>
<th>IOS Release</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2(1)</td>
<td>WS-X6066-SLB</td>
<td>SC6K-2.2-CSM</td>
<td>Supervisor 1A w/ MSFC &amp; PFC - Or - Supervisor 2 Module with MSFC 2</td>
<td>12.1(11b)E</td>
</tr>
</tbody>
</table>

Chapter 2, "Content Switching Design" provides guidelines, procedures, and configuration examples for implementing content switching with a CSM.

### Scaling DNS Services

DNS clients send queries in a single packet to a DNS proxy, which is a local agent that relays requests to DNS servers. A DNS server, if authoritative for the name requested, responds with the matching IP address. The complete DNS transaction is known as a type A query.

Figure 3 illustrates using a content switch to provide load balancing and improve availability for DNS servers. To eliminate a possible bottleneck or single point of failure, you can just add additional servers to the DNS server cluster. The content switch assumes the IP address of the name server (10.10.10.150), which is configured as the virtual IP address (VIP) and is advertised through DNS. The content switching mechanism can be provided by the IOS-SLB feature available in IOS or a CSM card in the Catalyst 6500 family aggregation layer switch.
Deploying content switches to load balance DNS servers provides the following benefits:

- Up to 65,000 UDP queries per second on the CSM 2.2(2a)
- Up to 40,000 UDP queries per second on the IOS-SLB
- High availability for the DNS servers thanks to DNS probes (available on both the CSM and IOS-SLB)

For information about how to implement load balancing for DNS servers, refer to Chapter 3, “Scaling DNS Services.”

Scaling Radius Services

Because Remote Access Dial-In User Service (RADIUS) servers are increasingly important for user authentication and session accounting, it is essential to ensure RADIUS server capacity and availability. You can use IOS SLB to balance RADIUS requests for authorization, authentication, and accounting among redundant RADIUS servers.

RADIUS is a general-purpose access server technology providing for authentication and accounting in a variety of applications. Cisco implements RADIUS into its Cisco Secure Access Control Server (ACS) software. You can use IOS SLB for load balancing RADIUS servers. In an 802.1x support scenario, user authentication is required for network connectivity and VLAN assignment. IOS SLB provides some RADIUS-specific extensions for supporting mobile wireless, such as a sticky database of usernames.

Figure 6 illustrates an IOS SLB network connection to a server farm. The switch, which is running IOS SLB, interconnects a router and a number of servers, and balances the load among the servers. An IP address for the virtual server is injected as a host entry into the routing table of the switch. Propagation of this IP address is controlled by the enterprise routing protocol.
To achieve final authentication, the RADIUS authentication process may require multiple packet exchanges between client and server. RADIUS client-initiated accounting requests may occur hours or days later during the RADIUS client/server session. For this reason, the RADIUS load-balancing solution should be able to direct RADIUS traffic from a specific client to the same RADIUS server.

For information about how to implement load balancing for RADIUS servers, refer to Chapter 4, “Scaling RADIUS Servers.”

**Data Center Networking Architecture**

The Data Center Networking architecture includes a suite of advanced solutions in the following areas:

- Data center IP network infrastructure
- Storage networking
- Application optimization
- Data center security
- Business continuance networking

As shown in Figure 1-5, data center services are related and interdependent. The storage networking and network infrastructure services are the foundation because they provide the fundamental building blocks used by every network service. After the infrastructure is in place, you can build server farms to support the application environments. These environments should be protected using network security technologies and optimized using load balancing and other application optimization technologies.

Once the data center is functioning in an efficient and secure way, you should ensure that the entire data center does not provide a single point of failure through the use of distributed data centers, site selection, SAN extension and other business continuance technologies.
Network Infrastructure

The Cisco intelligent switching infrastructure consolidates network components and resources by supporting distinct application and server environments on the same physical infrastructure, while maintaining their virtual separation for security and availability purposes. The term *infrastructure* refers to the Layer 2 and Layer 3 configurations that provide network connectivity to the server farm as well as the network devices that provide security and application-related functions. Data centers are composed of devices that provide the following functions:

- Network connectivity, including switches and routers
- Network and server security, including firewalls and intrusion detection systems (IDSes)
- Availability and scalability of applications, including load balancers, SSL offloaders and caches

The data center infrastructure must provide port density and Layer 2 and Layer 3 connectivity, while supporting security services provided by access control lists (ACLs), firewalls and intrusion detection systems (IDS). It must support server farm services such as content switching, caching, SSL offloading while integrating with multi-tier server farms, mainframes, and mainframe services (TN3270, load balancing and SSL offloading). For detailed information about designing and building your network infrastructure, see the following website:


Cisco Storage Networking

Direct-attached storage is expensive, difficult to manage, and inefficient, requiring very large amounts of unused capacity to ensure availability. Storage-area network (SAN) and network-attached storage (NAS) systems are widely implemented to consolidate storage, increase its availability, simplify management, and reduce capital and operational expenditures. Unfortunately, many traditional SAN solutions are limited, resulting in multiple SAN “islands” that lack the scalability and intelligence to deliver on the potential promised by storage networking.

Cisco provides fully integrated, multilayer, intelligent storage networking solutions, built with products such as the Cisco MDS 9000 Family, that scale to meet the needs of a SAN environment of any size. Cisco’s innovative solutions combine advanced storage switching functions such as virtual SANs, traffic management, and diagnostics with network-hosted storage services to provide unparalleled ease of management, scalability, and intelligence. For more information about Cisco storage networking, see the following website:


Data Center Security

Data center security protects the network infrastructure from internal and external threat and ensures data privacy and integrity. Cisco provides the technologies and products that allow you to effectively enforce your network security policy. Your security policy must accurately define access and connection requirements or the best technology will not provide the results you want.

You must protect your data center network both between functional and administrative regions within the data center and at its perimeter against external threats. Cisco delivers a powerful set of integrated network security technologies including access controls, firewalls, extranet VPN termination, network-based and host-based intrusion detection and prevention, and trust and identity services. These solutions can be deployed as standalone appliances or as modules for the Cisco Catalyst 6500 Series. For more information about data center security solutions, see the following website:
Business Continuance Networking

Business continuance is a top priority because customers expect continuous availability to an organization’s products and services, regardless of circumstances. Business continuance keeps essential applications running and protects valuable data during and after a disruption or failure.

Cisco networking solutions support a portfolio of business continuance strategies required to meet the different recovery point objectives (RPOs) and recovery time objectives (RTOs) of the enterprise applications. These networking solutions include site selection between distributed data centers and Storage Area Network (SAN) extension for mirroring mission-critical session traffic and data, and cost-effective WAN solutions for remote replication of data to offsite backup and storage locations.

The goal of disaster recovery and business continuance plans is guaranteed accessibility to data anywhere and at any time. Meeting this objective is all but impossible with a single data center, which is a single point of failure if a catastrophic event occurs. In a disaster scenario, with a single data center, the business comes to a standstill until it is rebuilt and the applications and data are restored. Cisco site selection solutions overcome this single point of failure, while providing additional benefits, such as application scalability, high availability, and load distribution.

SAN extension increases the geographic distance allowed for SAN storage operations, in particular for data replication and copy operations. By replicating or copying data to an alternate site, an enterprise can protect its data in the event of disaster at the primary site.

For more information about site selection and SAN extension business continuance solutions, see the following website:

Scaling Server Farms: Content Switching Design

Content switching is fast becoming a widely accepted technology for optimizing application services residing in the data center. Content switching deployments utilize intelligent services to maximize the efficiency of the individual servers residing in the server farm, which, in turn, increases the overall performance and resiliency of the application services located on these servers. The Content Switching Module (CSM) introduces a high performance module-based solution for scaling Layer 4 and Layer 5 data center server farm services.

Fundamental Challenges to Scaling Server Farms

Often, in order to increase the scalability and resiliency of server farm application services, each individual application is mirrored across multiple servers within the server farm. For this resiliency to be relayed to the requesting clients, each server’s IP address must be entered into a DNS server as an A record. This, in turn, creates a large number of IP addresses to be entered into DNS. When the DNS server receives a client request, it simply performs round robin load balancing on these configured A records and replies to the client. As the DNS server forwards the A record to the client, it has no knowledge of the current server or application availability. This, in turn, creates the possibility of user requests being forwarded to unavailable or non-responsive servers and applications. Figure 2-1 displays a logical overview of this process.
Benefits of Content Switching

By adding content switching services to the previous architecture, each of the challenges previously mentioned are addressed and alleviated. With the addition of the content switch, now multiple servers can be represented in DNS by a single IP address. When the content switch receives the client request, it is capable of performing server load distribution through a variety of load balancing algorithms, and is not limited to just round robin. The content switch is also capable of performing both server and application health checks which verifies the availability of these services prior to forwarding any client requests to the server. If the server or services are unavailable, the content switch does not forward the client request. This eliminates the likelihood of a user request being forwarded to a non-responsive server. Figure 2-2 displays the addition of a content switch to the previous network and includes its capabilities for representing multiple servers through a single IP address and performing server and application health checks.
What is the CSM?

The CSM is a content switch on a blade available for the Catalyst 6500 or 7600 platforms. It is capable of maintaining multiple Gigabit throughput rates (4 Gbps max inbound and outbound) while providing Layer 4-7 inspection and load balancing services.

The CSM provides performance for approximately 75,000–80,000 cps (connections per second) when configured to function at Layer 5, and approximately 125,000 cps when configured to function at Layer 4.

CSM Requirements

The CSM requires the use of Native IOS in the Catalyst switch and does not function with CatOS. Native IOS requires the presence of an MSFC on the supervisor card. The following table presents the software and hardware recommendations for the CSM and the Catalyst switch.

Table 2-1  CSM Hardware and Software Requirements

<table>
<thead>
<tr>
<th>CSM Software Release</th>
<th>Hardware Part Number</th>
<th>Software Part Number</th>
<th>Hardware Requirements</th>
<th>IOS Release</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2(1)</td>
<td>WS-X6066-SL B-APC</td>
<td>SC6K-2.2-CS M</td>
<td>Supervisor 1A w/MSFC &amp; PFC -Or- Supervisor 2 Module with MSFC 2</td>
<td>12.1(11b)E</td>
</tr>
</tbody>
</table>
Design Details

This design section discusses the basic data center network infrastructure requirements, content switching deployment goals, and the specific design information regarding deployment of the CSM.

Data Center Network Infrastructure

In order to provide optimal content delivery through the use of expanded Layer 4-7 services on the CSM, the content must first be accessible through lower layer network protocols. As a prerequisite to adding content switching services to the data center, the existence of a highly available and scalable Layer 2 and Layer 3 network infrastructure is of utmost importance. Figure 2-3 illustrates a data center infrastructure designed around these requirements.

Figure 2-3  Layer 2 and Layer 3 Data Center Network Infrastructure

By leveraging existing Layer 2 and Layer 3 protocols on Cisco switches and routers, it is possible to provide a network infrastructure that supports high availability and low convergence times, which in turn, increases the likelihood that any service failures are made transparent to the end user.
Content Switching Design Goals

Interoperability

The deployment scenarios covered in this chapter focus on the possible interoperability modes between a Catalyst 6500 and the CSM. While the available features vary between deployment modes, the main difference is in the support of the server default gateway. Offering multiple deployment modes expands the possibility of creating a viable solution to meet a variety of network design requirements. The common goal with each of these modes is providing an effective means for deploying and provisioning content switching services, while requiring little or no change to the existing data center network infrastructure.

The three deployment modes discussed in this design chapter include:

- Single subnet bridge mode
- Secure router mode
- One-arm mode

Scalability

Content switching deployments must be scalable. The logical and physical scalability must be able to increase, while requiring little or no change to the existing network infrastructure. The following factors limit scalability on a CSM:

- Connections per second (CPS), Concurrent connections (CC), and Packets per second (PPS)
- Total number of VIPs
- Client and server VLANs
- HSRP groups
- Keepalives
- Policies
- Real servers
- FT groups

Before deploying a content switch, you must consider the following:

- Total number of servers supported
- Total number of client and server VLANs required
- Total number and types of applications supported
- Types of rules and load balancing algorithms required
- Amount of acceptable failover, recovery, and downtimes

Traffic flow statistics should be gathered prior to deploying a content switch. This ensures the content switch can maintain the necessary performance numbers for CPS, CC, and PPS. These metrics are important because they determine how many connections the content switch is capable of setting up and the amount of connections it can maintain at any given point in time. These performance characteristics are further explained in an upcoming section.
Design Details

High Availability

The introduction of content switching services must not compromise existing network infrastructure high availability (HA). HA for content switching deployments is broken down into two main categories: HA for the server default gateway and HA for the content switch itself. While the options vary between deployment modes, the requirement for each remains constant. You can deploy HA for the CSM in an active-standby configuration. The active-standby configuration assists in maintaining a predictable traffic flow and decreases the chance for over subscription to occur upon a failure. For example, suppose two content switches were deployed in an active-active mode and each was running at 80% of its overall capacity. Upon failure of one of the content switches, the second content switch would not be able to maintain 80% of both its capacity as well as the capacity of the failed content switch, the result would be traffic loss.

HA is also available for applications services by using stateful failover. Stateful failover retains all connection session information between devices. The end-user experiences no “disconnect” upon failure. Maintaining stateful failover for session persistence information in an e-commerce environment is also important. The loss of session information in this environment results in the loss of transaction information like that associated with a user’s shopping cart. Due to the nature of web applications, end-users can directly perceive any loss of session information involved in that session.

Performance

Performance for a content switch is measured through three metrics: CPS, CC and PPS.

CPS defines the number of connections a content switch is capable of setting up on a per second basis. This is typically limited by the hardware capabilities of the switch. CPS performance varies based on the depth of packet lookup. The further into the packet the content switch must read, the slower the CPS setup rate. Hence, connection setup rates for Layer 4 rule are much faster than for a Layer 7 rule.

CC defines the total amount of active connections that can be open on a content switch at any given time. Once the content switch reaches the limit of CCs, the new connection requests are dropped. Although the CC number is high (1 million in the CSM’s case) a combination of high connection setup rates and long-lived flows can make the CC limit reachable. The CSM idle timer can be configured to age out old existing connections when they reach their time limit in order to make room for newer connections.

PPS defines the total amount of throughput capabilities for the content switch. Packet size has a direct effect on throughput capabilities. The capability of a content switch to perform hardware switching at Layer 2 is a requirement to achieve acceptable throughput performance results.

How the MSFC Communicates with the CSM

The MSFC forwards inbound traffic to the CSM through a four-Gigabit trunk logically configured on the Catalyst 6500 backplane. To ensure traffic is sent to the correct VLAN, all traffic traversing the trunk has an associated 802.1q tag. Once traffic leaves the Catalyst 6500, these VLAN tags are stripped from the packet as depicted in Figure 2-4.
Configure the MSFC to belong to either the client-side or server-side VLANs of the CSM, but never both. Configuring the MSFC to belong to both the client and server side VLANs can cause traffic to bypass the CSM.

### CSM Modes of Operation

The CSM operates in either one of two modes: RP (route processor) mode or CSM mode. It is recommended that you use the RP mode of operation because it allows you to utilize more than one CSM in the same chassis at the same time. After configuring the CSM to run in RP mode, all commands for configuring the CSM begin with “mod csm x”, where x is the slot in which the CSM resides.

```
switch1(config)#mod csm ?
<1-6>  slot where the CSM module resides
```

Another benefit of running the CSM in RP mode is that it allows for the simultaneous use of IOS-SLB while the CSM is in operation. IOSSLB refers to the server load balancing features available in certain software releases for the 7200 and 6500 series platforms. All commands relating to configuring IOS-SLB begin with the “ip slb”.

```
switch1(config)#ip slb ?
dfp  configure Dynamic Feedback Protocol manager
entries  initial and maximum SLB entries
firewallfarm  configure an SLB firewall farm
mode  configure SLB system mode
natpool  define client nat pool
probe  configure an SLB probe
route  SLB route
serverfarm  configure an SLB server farm
static  configure server initiated connection behaviour
vserver  configure an SLB virtual server
```

The other possible mode of operation is CSM mode. This mode is outdated and is not recommended because it only allows for the use of a single CSM in a chassis and additionally does not provide support for IOS-SLB.

### CSM Deployment

This section discusses two ideal locations for deploying CSMs within the enterprise data center; the data center aggregation switches and service switches.
Aggregation Switches

To ensure that traffic is properly routed through the CSM, place the CSM in a Catalyst 6500 that is in a direct path of traffic flows. The ideal location for this is in the data center aggregation switches. By placing the CSMs in the data center aggregation switches, they reside in a direct path for all traffic flows in and out of the data center. Management tasks also become easier because all Layer 2-3 and 4-7 services are housed in a central location. Additional content networking appliances, such as content engines, SSL offloading, and content transformation devices; are directly connected to these aggregation switches as shown in Figure 2-5.

Figure 2-5  CSMs Deployed in the Data Center Aggregation Switches

Packet Flow

Inbound

When the MSFC receives the inbound client request, it forwards the request to the CSM based on the destination IP address. The MSFC forwards all packets with a VIP as the destination IP address to the CSM. Each packet has an 802.1q tag associated with it for the incoming VLAN. This allows the CSM to associate all packets arriving on this VLAN with the corresponding client-side VLAN. These packets are then forwarded to the server VLAN and sent back over the internal trunk to the appropriate real server.
Outbound

The return traffic from the server is considered part of the same flow. Traffic from the server is set with a destination MAC of the CSM and a destination IP address of either the client or the server-side CSM VLAN, depending on the deployment mode used. Inbound and outbound traffics flows are illustrated in Figure 2-6.

**Figure 2-6 Inbound and Outbound Traffic Flows Through the CSM**

Service Switches

In some cases, it may not be possible to place the CSMs in the aggregation switches. This is typically due to either software version constraints and/or the lack of open slots and ports in the aggregation switches. Service switches solve these problems. Layer 2 trunks provide physical connectivity between the service switches and the aggregation switches. This is similar to the connectivity between the aggregation switches and the data center access switches. However, unlike the access switches, the service switch must have an MSFC. The MFSC is required on the service switch in order to run Native IOS. The CSM and other service devices are configured just as if they reside on the aggregation switches. The following figure shows the addition of services switches to the data center architecture.
Design Details

Figure 2-7  The CSMs Deployed in Data Center Service Switches

These switches are called service switches because they introduce a means for aggregating all services to one location. In Figure 2-7, the CSM, IDS, firewalls, content engines, SSL offloading, and content transformation devices are all directly connected to the service switches.

The addition of these switches is often beneficial for a number of reasons. You may sometimes need to place restrictions on the software versions running on the core data center network infrastructure. Because of this restriction, it is not possible to run a software version on the aggregation switches capable of supporting the CSM. By adding services switches to the data center infrastructure, it is possible to run the necessary IOS software versions to support the CSM without interfering with the IOS software currently running on the data center aggregation switches. Low, or nonexistent failover times for Layer 4-7 services are other added benefits associated with the addition of service switches. If all Layer 4-7 devices are connected directly to a aggregation switch, and that switch fails, all attached services must also failover. Because the service switches are connected through Layer 2 links to both aggregation switches, they allow for the failure of a aggregation switch without the need to fail all Layer 4-7 services.
Packet Flow

**Inbound**

When the MSFC on the active aggregation switch receives an inbound request with a VIP as the destination address, it forwards the request to the CSM client-side VLAN over a Layer 2 trunk to the service switch. Once the CSM receives the packet, the CSM forwards it to the server-side VLAN, load balances it to a real server, and forwards it to the server VLAN over the same trunk on which it was received. Because both the service and access switches are Layer 2 adjacent to the aggregation switch, when the packet reaches the aggregation switch, it simply performs a Layer 2 lookup and forwards the packet to the correct port as shown in Figure 2-8.

**Outbound**

The server forwards the return traffic to the MAC address of the CSM. When the packet from the server reaches the aggregation switch, the switch performs a Layer 2 lookup and forwards the packet over the trunk connected to the service switch. The CSM then forwards the packet to the client-side VLAN, where it is sent to the Layer 3 interface of the MSFC providing the default gateway to the CSM client-side VLAN.

**Deployment Modes**

The CSM provides content switching services through a variety of deployment modes. Each of these modes is differentiated through the location of the server default gateway. Each also supports different features and functionality, providing a variety of means for meeting most network design requirements. This section covers the theoretical concepts associated with each mode, and the implementation section
of this design chapter provides the details of how to deploy each mode. For more information regarding global server load balancing, see the Enterprise Distributed Data Center Solutions Reference Network Design (SRND).

**Single Subnet Bridge Mode**

In single subnet bridge mode, the CSM bridges traffic flows between the client-side and server-side VLANs, rewriting the destination Layer 2 MAC address. In this deployment mode, the Catalyst 6500 MSFC provides the default gateway for the servers. This mode also allows you to leverage Cisco IOS high availability features, like Cisco’s Hot Standby Router Protocol (HSRP), to support redundancy for the server default gateways. The following diagram displays a logical overview of the CSM operating in single subnet bridge mode.

**Figure 2-9  Overview of the CSM Configured for Single Subnet Bridge Mode**

---

**Secure Router Mode**

When configured in secure router mode, the CSM routes traffic flows between client-side and server-side VLANs belonging to different subnets, and rewrites both the destination MAC address and IP address. In this mode, the server’s default gateway changes location, moving from the MSFC to the CSM. Support for high availability for the default gateway is through the `alias` command on the CSM. The `alias` command provides functionality very similar to HSRP by supplying a “floating” IP and a virtual MAC address which servers point to as the default gateway. The following figure represents a logical overview of the CSM secure router mode.
One Arm CSM Mode

The one arm CSM mode provides a means for optimizing backend server-to-server communication. Often, servers residing in the data center need to communicate with each other or with databases residing within the data center. These communications do not necessarily need to be sent to the content switch for load balancing. This deployment mode utilizes policy based routing (PBR) to provide a means for configuring the Catalyst 6500 to only route certain traffic flows to the content switch, therefore alleviating unnecessary traffic flows from being forwarded to the content switch. The mode has other advantages as well. It allows you to configure the server default gateway on the MSFC allowing high availability to be provided by HSRP. It also allows you to use certain Cisco IOS features, such as Private VLANs, for the server farm.

High Availability

Both the MSFC and the CSM provide several high availability options, which differ between CSM deployment modes. The MSFC uses HSRP to provide redundancy for Layer 3 interfaces that often serve as default gateways for the CSM client-side VLAN and real servers. The CSM uses a fault tolerant (FT) VLAN configuration to provide redundancy between each pair of CSM modules residing in either the same or separate chassis.

Because the high availability configuration for the Catalyst switch and CSM is done separately, it is possible to have a scenario where an active Catalyst switch is housing a standby CSM. To maintain an easier environment for managing both physical devices and traffic flows, the recommendation is that all active Layer 4-7 devices are connected to the active Catalyst switch.
The high availability options for each deployment mode are covered in detail in the Implementation Details section of this chapter.

## Network Address Translation (NAT)

### Client NAT

Client NAT on the CSM operates in a very similar fashion to the client NAT on Cisco routers. All incoming client addresses are translated to an IP address that is part of a pool of internal addresses. This pool usually consists of IP addresses belonging to the RFC 1918 private address space.

### Server NAT

Configure server NAT on the CSM either through the use of a one-to-one static map or by using port address translation (PAT) to overload the Vserver IP address (VIP). When server NAT is enabled on the CSM, clients are unable to reach the real server’s IP address. The following diagram displays traffic flows with client and server NAT enabled on the CSM.

*Figure 2-11  Traffic Flows with Client and Server NAT Enabled on the CSM*
Recommendations

The recommended deployment mode must be based directly on the requirements for the network. Because these requirements vary widely from one network to another, it is not practical to recommend one deployment mode for all scenarios. Each deployment mode has its own set of requirements and caveats. The following implementation section covers the configuration details and caveats associated with each of these scenarios. Use this section to gather the information regarding each scenario to assist you in making the proper deployment mode selection.

Implementation Details

The following diagram is a reference for all deployment modes discussed in this section.

Figure 2-12  CSM Implementation Details

Single Subnet Bridge Mode

Initial Configuration

The CSM bridges traffic flows between the client side and server side VLANs in single subnet bridge mode. Perform the following steps to create a basic configuration for single subnet bridge mode.

Step 1  Add the CSM client and server VLANs to the VLAN database.
### Implementation Details

#### Step 2
Create the CSM client and server VLANs.

#### Step 3
Create a server farm of real servers.

#### Step 4
Create a vserver (VIP).

#### Step 5
Verify status.

Prior to configuring the client and server VLANs on the CSM, you must first add each VLAN to the VLAN database. In this example VLANs 5 and 10 are added to the VLAN database.

```
switch1(config)#vlan 5
switch1(config-vlan)#
switch1(config)#vlan 10
switch1(config-vlan)#
```

Once the VLANs are operational, you can add them to the CSM configuration. It is important to note that the IP addresses for both the CSM client-side and server-side VLANs are exactly the same. This portion of the configuration enables the CSM to bridge traffic flows between the client and server VLANs.

```
switch1(config)#mod csm 4
switch1(config-module-csm)#vlan 5 client
switch1(config-slb-vlan-client)# ip address 192.168.2.9 255.255.255.0
switch1(config-slb-vlan-client)# gateway 192.168.2.1

switch1(config)#mod csm 4
switch1(config-module-csm)#vlan 10 server
switch1(config-slb-vlan-server)# ip address 192.168.2.9 255.255.255.0
```

Notice the gateway command under the VLAN 5 client configuration mode. This command provides the CSM with a default gateway for all traffic destined for VLAN5. Because a default gateway must be on the same subnet as the client-side VLAN, you must configure a Layer 3 interface for Vlan 5 on the MSFC. In single subnet bridge mode, this Layer 3 interface also provides the default gateway support for any real servers belonging to this subnet.

All services are defined in the server farm command sub-mode. These services may include real servers, Content Engines, SSL Offloading, and Content Transformation devices. You can assign multiple server farms to each vserver. Here, each real server is assigned an IP address and brought into service. At least one of the configured real servers must be brought into service for each server farm. If not, the vserver (VIP) to which the server farm is assigned does not become operational.

```
switch1(config)#mod csm 4
switch1(config-module-csm)#serverfarm s1
switch1(config-slb-sfarm)#real 192.168.2.10
switch1(config-slb-real)#inservice
switch1(config-slb-sfarm)#real 192.168.2.11
switch1(config-slb-real)#inservice
```

You must create a vserver (VIP) and at least one server farm with at least one operational, defined real server assigned. A vserver is defined by an IP address with an associated default mask of 255.255.255.255. When the vserver is advertised to the rest of the network, it is advertised as a /32 host route. A vserver can provide matching for traffic to be load balanced on any IP protocol or, for increased granularity, on a specific TCP or UDP port. You can also assign policies to perform any necessary Layer 5 load balancing. Once a server farm is assigned to the vserver, it must be brought in service to become operational.

```
switch1(config)#mod csm 4
switch1(config-module-csm)#vserver vs1
switch1(config-slb-vserver)#virtual 192.168.2.100 tcp 80
```
In some cases, IP address allocation restrictions do not allow you to configure a VIP that resides on the same subnet as the CSM client side VLAN. This creates a routing problem because the VIP is now not configured as a directly attached subnet, and the MSFC has no way of knowing how to reach it. To solve this problem, simply add a static route to the Vserver on the MSFC. In the example below, the CSM client-side VLAN has been configured for the 192.168.2.0/24 subnet and the VIP has been configured with an IP address of 192.168.100.100.

```
vlan 5 client
ip address 192.168.2.9 255.255.255.0
gateway 192.168.2.1

vserver VS1
virtual 192.168.100.100 tcp 80
serverfarm S1
persistent rebalance
inservice

The following static route allows the MSFC to reach the VIP.

ip route 192.168.100.100 255.255.255.255 192.168.2.9

Use the following show command to determine if the vserver is operational.

switch1#sh mod csm 4 vservers
```

Figure 2-13 illustrates the CSM deployed in single subnet bridge mode.

**Figure 2-13 The CSM Configured for Single Subnet Bridge Mode**
Additionally, to verify the status of the real servers, use the following command.

`switch1#sh mod csm 4 reals`

```
<table>
<thead>
<tr>
<th>real</th>
<th>server farm</th>
<th>weight</th>
<th>state</th>
<th>conns</th>
</tr>
</thead>
<tbody>
<tr>
<td>192.168.2.10</td>
<td>S1</td>
<td>8</td>
<td>OPERATIONAL</td>
<td>0</td>
</tr>
<tr>
<td>192.168.2.11</td>
<td>S1</td>
<td>8</td>
<td>OPERATIONAL</td>
<td>0</td>
</tr>
</tbody>
</table>
```

The following command is also useful for verifying connectivity of the real servers.

`switch1#ping mod csm 4 reals`

```
<table>
<thead>
<tr>
<th>IP address</th>
<th>Reachable</th>
</tr>
</thead>
<tbody>
<tr>
<td>192.168.2.10</td>
<td>Yes</td>
</tr>
<tr>
<td>192.168.2.11</td>
<td>Yes</td>
</tr>
</tbody>
</table>
```

**Default Gateway High Availability**

The MSFC provides default gateway support for the servers when the CSM is configured in the single subnet bridge mode. Use HSRP on the MSFC to provide high availability for the Layer 3 interfaces acting as the default gateway.

*Figure 2-14 Default Gateway High Availability with CSM in Single Subnet Bridge Mode*
VLAN 5 provides the CSM client-side and real server default gateway and was configured with HSRP. The HSRP tracking feature is used to detect a link failure for GigabitEthernet port 1/1.

```plaintext
interface Vlan5
  ip address 192.168.2.2 255.255.255.0
  no ip redirects
  standby 2 ip 192.168.2.1
  standby 2 priority 110
  standby 2 track GigabitEthernet1/1
```

Use multi-group HSRP (MHSRP) when multiple server default gateway support is required.

### Server Initiated Connections

You must add specific commands to the configuration for the CSM to keep track of server-initiated connections when the CSM is configured in the bridging mode. The CSM allows server initiated connections by default, but does not list them in the CSM connection table. This is because the CSM simply forwards outbound server initiated flows to the client-side VLAN while in single subnet bridge mode. Use the "sh mod csm 4 conns" command to verify if there are any active server initiated connections in the CSM. The output does not list any connections as being active.

```
switch1#sh mod csm 4 conns
  prot vlan source                destination           state
  -----------------------------------------------------------------------------
  switch#
```

Take the following steps to ensure that the CSM correctly displays server-initiated connections in the connection table.

**Step 1** Create a server farm that does not include any real servers and use the predictor forward algorithm.

**Step 2** Create a vserver to accept all traffic for any protocol.

**Step 3** Permit only the server VLAN to access this vserver.

**Step 4** Add the server farm to the vserver configuration.

By creating a server farm that uses predictor forward with no real servers configured, all server-initiated traffic that belongs to the allowed VLAN is simply forwarded to the client-side VLAN.

To create the server farm, perform the following.

```
switch1(config)#mod csm 4
switch1(config-module-csm)#serverfarm s3
switch1(config-slb-sfarm)#predictor forward
```

Now create the vserver, any allowed server VLANS, and assign the server farm.

```
switch1(config)#mod csm 4
switch1(config-module-csm)#vserver vs3
switch1(config-slb-vserver)#virtual 0.0.0.0 0.0.0.0 any
switch1(config-slb-vserver)#vlan 10
switch1(config-slb-vserver)#serverfarm s3
switch1(config-slb-vserver)#inservice
```

In this configuration, the vserver is set to receive incoming traffic for any protocol arriving from VLAN 10. The CSM is now able to inspect each flow and lists it in the connection table. Using the "sh mod csm 4 conns" command now lists any active server initiated connections from VLAN 10.

```
switch1#sh mod csm 4 conns
  prot vlan source                destination           state
```

Client NAT

Client NAT helps ensure a predictable traffic flow when the CSM is configured in single subnet bridge mode. Because the CSM translates both the Layer 2 and Layer 3 source information, all of the outbound return packets point to the CSM for both Layer 2 and Layer 3 destination addresses. This decreases the likelihood of outbound packets bypassing the CSM.

The use of client NAT does introduce some limitations on traffic logging. When client NAT is enabled, server transaction and access logs do not reflect the real source IP of the client. Because of this, network administrators at times choose not to enable client NAT.

There are two steps to configuring client NAT on the CSM.

**Step 1** Create a pool of internal IP addresses to use for translation.

**Step 2** Enable client NAT in serverfarm configuration submode.

First, create the pool of addresses.

```
switch1(config-module-csm)#natpool clients 192.168.2.120 192.168.2.129 netmask /24
```

Now enable client NAT on the server farm.

```
switch1(config-module-csm)#serverfarm s1
switch1(config-slb-sfarm)#nat client clients
```

Server NAT

In order for server NAT to operate properly while the CSM is configured in single subnet bridge mode, you must first complete the configuration to allow server initiated connections.

Because all outbound server initiated flows are normally just bridged, all rules are also bypassed, including server NAT. To enable server NAT while the CSM is running in bridge mode, perform the following steps.

**Step 1** Enable active server initiated connections to be listed on the CSM by performing the previous steps.

**Step 2** Configure a client NAT pool of addresses to be used for translation.

**Step 3** Enable client NAT for the server farm.

Beginning with step two, create a pool of addresses.

```
switch1(config-module-csm)#mod csm 4
switch1(config-module-csm)#natpool server 192.168.2.101 192.168.2.105 netmask 255.255.255.0
```

Enable client NAT for the server farm, referencing the pool of addresses.

```
switch1(config-module-csm)#serverfarm s3
switch1(config-slb-sfarm)#nat client server
```
Now, all server-initiated connections from VLAN 10 are translated to one of the IP addresses belonging to this pool. The real servers IP address does not appear as the source on the client.

```
switch1#sh mod csm 4 conns
         prot vlan source        destination          state
In  TCP  10  192.168.2.10:1053       10.15.0.6:23          ESTAB
Out TCP  5    10.15.0.6:23          192.168.2.101:8193      ESTAB
```

Verify that all server-initiated connections are translated to one of the IP addresses belong to the pool by using the netstat command on the client.

```
[root@client3 /root]# netstat -a
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0          138     10.15.0.6:telnet        192.168.2.101:8193 ESTABLISHED
```

**Caveats**

Because the CSM is incapable of performing routing while configured in single subnet bridge mode, all servers must be Layer 2 adjacent to the CSM. This also means that all “noise” and broadcast traffic is bridged through the CSM. This may hinder CSM performance as the amount of “noisy” traffic increases. ESE has not tested CSM performance issues associated with bridge mode. The testing is scheduled to occur in the near future with the results documented in future revisions of this chapter.

The details associated with configuring server NAT and server initiated connections are more cumbersome than with secure router mode, which may discourage some administrators from deploying the CSM in bridge mode.

**Secure Router Mode**

When you configure the CSM in secure route mode, it routes between the client-side and server-side VLANs. Because the CSM is now configured to route, you can now place servers more than one Layer 3 hop from the CSM. The CSM supports the server default gateway in this mode. High availability for the server default gateway is provided through the CSM alias command.

**Initial Configuration**

The initial configuration for the client and server side VLANs instruct the CSM to operate in either bridge mode or secure router mode. The difference between bridge mode and secure router mode is the IP addresses assigned to the client and server VLANs of the CSM. Otherwise, the configuration tasks required for real servers and server farms are identical to those while running in single subnet bridge mode.

To enable secure router mode on the CSM, configure the client and server VLANs with different IP addresses.

```
switch1(config-module-csm)#vlan 5 client
switch(config-slb-vlan-client)#ip address 192.168.1.9 255.255.255.0
switch(config-slb-vlan-client)#gateway 192.168.1.1
switch1(config-module-csm)#vlan 10 server
switch(config-slb-vlan-server)#ip address 192.168.2.2 255.255.255.0
switch(config-slb-vlan-server)#alias 192.168.2.1 /24
```
Configure the client-side VLAN (VLAN 5) and the server-side VLAN (VLAN 10) on different subnets, thereby enabling the CSM to function in secure router mode. The gateway command, under the client VLAN configuration, points to a Layer 3 interface for VLAN 5 on the MSFC. Multiple server-side VLANs can be routed to either single or multiple client-side VLANs.

The server-side VLAN includes the use of an alias IP address. This alias IP address is the default gateway for the servers residing on VLAN 10. It provides a similar function to HSRP. The alias IP address and its virtual MAC address “float” between two redundant CSMs, providing a fault tolerant default gateway for servers as shown in Figure 2-15. The “sh mod csm 4 arp” command displays the alias IP address and its associated MAC address.

```
switch1#sh mod csm 4 arp
Internet Address  Physical Interface  VLAN      Type       Status
--------------------------------------------------------------------
192.168.1.1        00-01-64-F9-1A-01   10         -ALIAS-    local
```

**Figure 2-15 The CSM Configured in Secure Router Mode**

**Server-Initiated Connections**

To allow server initiated connections while running in router mode, you must configure either static NAT or a static route to the server-side VLAN. This is because the server-side VLAN is configured on the CSM with no corresponding Layer 3 interface on the MSFC, making the CSM unknown to the MSFC.

**Client NAT**

Configure client NAT for secure router mode the same way as for single subnet bridge mode. See the client NAT configuration under single subnet bridge mode for more details.
Server NAT

You can configure static NAT to provide translation either through a single IP address, or by performing PAT (Port Address Translation) for the VIP. To configure static NAT to use PAT on the vserver, perform the following IOS commands.

```
switch1(config)#mod csm 4
switch1(config-module-csm)#static nat virtual
switch1(config-slb-static)#real 192.168.2.10
```

Now all server initiated connections appear to the client as being sourced by the VIP.

```
[root@client3 /root]# netstat
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address Foreign Address State
tcp 0 126 10.15.0.6:telnet 192.168.1.100:8198 ESTABLISHED
```

An alternative to enabling static NAT is to create a static route to the CSMs server-side VLAN. The following static route tells the MSFC to forward all traffic destined for the CSM server VLAN to the VIP address.

```
switch1(config)# ip route 192.168.2.0 255.255.255.0 192.168.1.100
```

Caveats

Running in secure router mode dictates that the server default gateway must be placed on the CSM. This is considered to be a caveat by some because it does not allow the use of HSRP to support the server default gateway.

CSM High Availability

High availability for redundant CSM modules is provided through a Fault Tolerant (FT) VLAN configuration. When the CSMs are configured for high availability, they act as an active-standby pair. Current software releases do not allow the CSM to use the FT VLAN configuration in an active-active setup. Each pair of CSMs must be configured with a separate FT VLAN.

The CSM with higher priority assigned for the FT group becomes active, while the CSM with the lower priority takes on a standby role. By default, the hello interval of this FT information between CSMs is set to one second, and the default failover time (dead timer) is set to three seconds. Use the “`show mod csm x ft`” command to verify assigned VLAN, current state, priority, and timer information. The following displays the FT information for both the active and standby CSMs.

```
switch1#sh mod csm 4 ft
  FT group 1, vlan 200
  This box is active
    priority 30, heartbeat 1, failover 3, preemption is off
switch2#sh mod csm 4 ft
  FT group 1, vlan 200
  This box is in standby state
    priority 20, heartbeat 1, failover 3, preemption is off
```

You can exchange CSM FT information over either a dedicated trunk or a Layer 2 EtherChannel. If you use a trunk to exchange the FT information, it is recommended that only the FT VLAN be allowed on the trunk. This is to ensure that other traffic does not overrung the link causing the loss of FT hello packets. If you use a Layer 2 EtherChannel, it is recommended that you enable QoS on the link guaranteeing priority to the FT hello packets. This is to ensure that a delay is not experienced with the
exchange of hello packets causing both CSMs to become active. By default QoS (COS 7) is enabled on all Supervisor and MSFC combinations except for the Supervisor 2 MSFC 2 combination. If you are using this combination, you must manually enable QoS.

You can enable stateful failover on the CSM for both session and sticky information. When you enable stateful replication, state table information is replicated to the standby CSM. This ensures the standby CSM is aware of all current state information if a failure occurs. To enable replication of both state and sticky (session persistence) information perform the following.

```
switch1(config-module-csm)#vserver vs1
switch1(config-slb-vserver)#replicate csrp connection
switch1(config-slb-vserver)#replicate csrp sticky
```

To verify stateful failover is operating correctly, open a telnet connection from the client to a server through a VIP on the CSM. You can verify the current state for the active connection on the primary CSM by issuing the following command.

```
switch1#sh mod csm 4 conns
```

<table>
<thead>
<tr>
<th>prot</th>
<th>vlan</th>
<th>source</th>
<th>destination</th>
<th>state</th>
</tr>
</thead>
<tbody>
<tr>
<td>In TCP 5</td>
<td>10.15.0.6:1034</td>
<td>192.168.1.100:23</td>
<td>ESTAB</td>
<td></td>
</tr>
<tr>
<td>Out TCP 10</td>
<td>192.168.2.10:23</td>
<td>10.15.0.6:1034</td>
<td>ESTAB</td>
<td></td>
</tr>
</tbody>
</table>

This command establishes a telnet connection between the client and server. When the primary CSM fails, stateful replication ensures that this connection is carried over to the secondary CSM when it becomes active.

The following example displays the CSM on switch 2 is now the active CSM for FT group 1.

```
switch2#sh mod csm 4 ft
FT group 1, vlan 200
This box is active
priority 20, heartbeat 1, failover 3, preemption is off
The telnet connection was also maintained and uninterrupted between the client and server as verified in the example below.
```

```
switch2#sh mod csm 4 conns
```

<table>
<thead>
<tr>
<th>prot</th>
<th>vlan</th>
<th>source</th>
<th>destination</th>
<th>state</th>
</tr>
</thead>
<tbody>
<tr>
<td>In TCP 6</td>
<td>10.15.0.20:1070</td>
<td>192.168.1.100:23</td>
<td>ESTAB</td>
<td></td>
</tr>
<tr>
<td>Out TCP 10</td>
<td>192.168.2.10:23</td>
<td>10.15.0.20:1070</td>
<td>ESTAB</td>
<td></td>
</tr>
</tbody>
</table>

The total length of time for the failover to occur was approximately 3.5 seconds. It should be noted that this failover time was achieved on a CSM with a very light traffic load. On a CSM with heavier traffic loads, it is possible for failover times to become longer. The results of heavy traffic loads have not been tested thoroughly by ESE. The tests are scheduled for the near future with the results published in a future revision of this chapter.

---

**Scalability**

ISBU has defined the following scalability limits. ESE scalability and performance testing results will be published to the ESE website once completed.

**Configuration Scalability**

- 256 VLANs (total for both client and server)
- 4000 virtual servers
• 4000 server farms
• 16,000 real servers
• 4000 probes
• 16,000 access control list

Performance Summary

Connections
• 1,000,000 concurrent TCP connections
• ~125,000 connection setups per second - Layer 4
• ~80,000 connection setups per second - Layer 5

Throughput

• 4 Gigabits-per-second total combined (client-to-server and server-to-client) throughput
Scaling DNS Services

DNS traffic in an intranet or Internet is mainly made of DNS queries and DNS responses. Clients send queries in a single packet to a DNS proxy. The DNS proxy is an agent local to the client that relays the requests to DNS servers. A DNS server, if authoritative for the name requested, responds with the matching IP address. This is a complete DNS transaction known as a query type A.

The client with IP address 10.0.0.5 needs to communicate with the web-server www.foo.com in Figure 3-1. In order to retrieve the IP address for www.foo.com, the client sends a DNS query to its local DNS server, which in turn sends a DNS query to the name server called nameserver.foo.com. Nameserver.foo.com replies with the IP address of the web server www.foo.com 10.10.10.80. The traffic exchange occurs using UDP as the transport protocol.
Normally, there are multiple name servers for the same zone (a domain such as foo.com): a primary (also called master) and multiple secondary servers (also called slaves). Primary and secondary name servers have the same level of authority in a zone. The designation of master or slave depends on how you update your DNS servers configuration files, known as zone files. Use the master server to periodically update your name server’s zone files. The slave servers then pull the updated configurations from the master in a process known as a zone transfer.

Figure 3-2 depicts master to slave communication. The master is located inside the Enterprise network while the secondary is in the Demilitarized Zone (DMZ). The communication between the primary and secondary servers is over UDP when comparing zone file revision numbers and over TCP during the actual configuration file transfer between the primary to secondary DNS servers.

The DNS design shown in the Figure 3-2 is a simplified one. It shows the DNS server in the DMZ as the secondary server. When you update the internal name server, you also update the secondary DNS in the DMZ. A more advanced DNS design takes into account the threat of exposing internal DNS host names to the outside world, but this is outside the scope of this chapter.
When the traffic on nameserver-2.foo.com reaches overload, the name resolution for www.foo.com slows down. Adding additional servers in a DNS server cluster and placing a content switch in front of the cluster alleviates the bottleneck. The content switch assumes the IP address of nameserver-2.foo.com (10.10.10.150). That same IP address is configured as the VIP and is advertised as the DNS service. Figure 3-3 displays an example of a data center with a number of secondary servers behind a content switch.

The content switch described in this chapter is either the IOS-SLB feature available in IOS or a CSM card in the Catalyst 6500 family aggregation layer switch.
Benefits

The advantages of deploying content switches to load balance DNS servers include:

- Increased number of queries on the CSM
  - Up to 65,000 UDP queries per second on the CSM 2.2(2a)
  - Up to 90,000 UDP queries per second on a future CSM release
- Increased number of queries on the IOS-SLB
  - Up to 40,000 UDP queries per second
- High availability for the DNS servers thanks to DNS probes (available on both the CSM and IOS-SLB)
System Components

Hardware Requirements

Choose one item from the following equipment list to support DNS load balancing:

- A pair of Catalyst 6500s with CSMs
- A pair of Catalyst 6500s with MSFC2 and the IOS-SLB feature set

The configuration described in this chapter uses the IOS `ip default next-hop` option. This option is not a requirement. This design also supports the basic modes of operation on the CSM (router or bridge mode).

If you choose IOS-SLB, use the MSFC2 on the aggregation switches for performance reasons.

Software Requirements

For the testing of this application note, the following releases of code were used:

- Supervisor IOS version 12.1(11b)E2 for the aggregation switches
- Release 2.2(2a) of the CSM software

Features

Idle Timeout

Content switches cache “established connections” between clients and servers in a session table used for lookup purposes. When the lookup process returns a match, the content switch sends the packet without additional processing. Each entry in the session table is specific to a single client-server communication. Content switches reclaim cached connections when the client and server close them.

An idle connection is an entry that stays open, but unused, for a long period. The idle timeout feature allows you to specify how long a content switch keeps an idle connection in its cache.

While TCP explicitly creates and closes a connection, UDP traffic does not use connections. Therefore, content switches must age UDP entries based on a timer. Because DNS client-server exchanges are short, Cisco recommends that you set the idle timeout to the minimum value of 4 seconds. On the CSM and IOS-SLB, the idle timeout is set on a per virtual server basis. The bigger the idle timeout value, the faster the cache reaches its maximum limits under sustained connection setups.

For example, on the CSM, the maximum number of concurrent connections is 1,000,000. Using the default idle timeout of 3600s, the CSM sustains an average of 1,000,000 /3600s, which equals 278 requests per second. Using a timeout of 20s, the CSM sustains 1,000,000 / 20s, which equals 50,000 requests per second. Therefore, decreasing the idle timeout value on your CSM/IOS-SLB lowers the speed at which the Content Switch reaches the limit of concurrent connections. Keep in mind the speed at which the concurrent connections increase depends on the connection per second rate.
Static NAT

Network address translation (NAT) allows server originated connections to leave the data center using a single Public IP address. Port address translation (PAT), a special case of NAT, is a more accurate description of this process. In this configuration, all server originated traffic uses a single IP address. The “static” command in both CSM and IOS-SLB configurations defines NAT. Server originated connections are crucial in a correct DNS configuration because secondary DNS servers must perform zone transfers and source of authority (SOA) queries to the master DNS server.

Per-Packet (IOS-SLB Only)

The “per-packet” feature is an option available on IOS-SLB to increase performance for load balancing protocols that do not require state connection information maintenance. DNS is one of these protocols. Normally, IOS-SLB maintains connection “state” information even for UDP: when a UDP request comes in from a client, IOS-SLB creates an entry in the cache for the traffic in the reverse direction.

DNS queries do not require this intelligence from a Content Switch. All the Content Switch really needs to do is rewrite IP addresses (source or destination depending on the direction of the traffic), which is called NATing. This is exactly what the “per-packet” option does.

The per-packet option disables the caching of UDP entries: the virtual server becomes a NAT device. Both virtual servers and “static” use per-packet. “Static” is the configuration keyword to define a NAT pool.

Per-packet is enabled on a per virtual server (vservers) and a per “static” basis:

- Under the “vservers” configuration, NAT translates IP addresses on packets from the client to the server
- Under the “static” configuration, NAT translates IP addresses on packets from server to client

DNS Probes

Content switches use probes to check the availability of the servers being load balanced. DNS probes are DNS query type “A” for a specific fully qualified domain name (FQDN) that you configure. You can use the probe responses to either compare IP addresses, or just assume that any positive DNS response means that the server is healthy. A negative response or no response is an indication of server failure, which forces the content switch to mark the server as unavailable, therefore removing it from the load distribution rotation.

Design Description

Topology

The Cisco multi-layer network design is the basis for the network topology described in this chapter. You can find Cisco’s multi-layer network design documented at http://www.cisco.com/warp/public/cc/so/nso/lnso/cpsg/cnld_wp.htm. For the purpose of this discussion, a number of DNS servers were front-ended by a Catalyst 6500 switch running either IOS-SLB or the CSM as the content switch.
DNS Server Design

There are two possible designs that influence the load balancing of DNS servers:

- Having a Primary (or master) name server in the server farm
- Having only Secondary (or slave) name servers in the server farm

In an enterprise, there is typically a primary or master DNS server placed in the internal network and a secondary DNS server located in the DMZ. You place the secondary DNS server in the DMZ for security reasons. Depending on the number of requests that each of these DNS servers receives, you could place a content switch in front of either DNS farm. On DMZ DNS server farms, typically there are only secondary DNS servers behind the content switch. On internal DNS server farms, there is either a single or multiple primary servers behind a content switch.

The key difference between a primary and a secondary name server is how the zone transfer takes place. Secondary name servers pull updated configurations from the primary name server based on revision number. Secondary name servers first send a special request to retrieve the revision number of the configuration file. This request is an SOA request sent over UDP and destined for port 53. If the revision number is higher than the local number, the secondary DNS server opens a TCP connection to port 53 on the primary, and requests the actual zone transfer, an AXFR query, to retrieve the configuration.

Virtual Primary Name Server Provided by the Server Farm

You can use a content switch to create a “virtual” primary name server or a “virtual” secondary name server. This section describes the first option.

There are two different ways to design a “virtual” primary name server:

- Using a server farm of one primary and multiple secondary name servers
- Using a server farm of primary name servers

The first design is one in which there is a primary name server in the server farm and all other servers are secondary. The secondary name servers use zone transfers to obtain copies of the zone file from the primary.

The second design consists of a server farm made only of primary name servers. This creates the restriction that you must manually update all of your name servers each time you add new records, which could add significant management overhead.

Both designs allow you to perform zone transfers with any server in the server farm: a secondary name server can act as a master for a zone transfer. Having the primary name server in your server farm requires configuring a virtual server for TCP traffic to port 53 to allow incoming requests for zone transfers, see Figure 3-4.
Virtual Secondary Name Server Provided by a Server Farm

This section describes how to create a “virtual” secondary name server. In this scenario, there are only secondary name servers in the server farm. The secondary servers pull the zone files from a primary name server that is outside the server farm, as depicted in Figure 3-5.
DNS Record Configuration

When configuring either the primary or secondary name servers, it is important to remember that, to the external client, the server farm looks like a single DNS server. Therefore, even if you set the IP address of your servers to a private address, remember that the IP address of nameserver-2.foo.com is the Virtual IP (10.10.10.150). This requires modification of the NS records that define nameserver-2.foo.com.

Traffic Paths

DNS queries are either recursive or non-recursive. This categorization defines how the server reacts when it does not have the answer to a query.

Non-Recursive Queries

A client sending a “non-recursive” request to a server is asking the server to supply the information only if available. If the server does not have the information, it sends back a response with an error code. “Non-recursive” queries are the simplest type of query.

A DNS query type “A” requests a server to return an IP address for a fully qualified domain name. An example is requesting the IP address for www.foo.com.
A DNS query type “NS” requests the IP address of a Name Server that knows about a specific domain. For example, a local DNS proxy trying to resolve www.foo.com on behalf of a client must know the address for the name server responsible for foo.com. In Figure 5, nameserver-2.foo.com (10.10.10.150) is the authoritative name server for foo.com that the local DNS proxy returns to the client.

The basic DNS server farm must be able to answer DNS non-recursive queries. These queries are UDP packets destined for port 53. The client for this request is typically a local DNS server. The source of these requests may be the Internet or intranet.

**DNS Recursive Queries**

A recursive DNS query identifies the behavior of a name server when it must contact other name servers to resolve the query on behalf of the client. A DNS resolver (software running on the client PC) generates this query and sends it to the local DNS server. Since the local DNS server performs the proxy DNS function for the client, this function is often referred to as DNS proxy.

For this type of request, the name server becomes the client to a number of DNS queries sent out to other name servers. The DNS server farm must be able to perform recursive queries. The environment using content switches must support server-originated requests from inside the server farm. The source port for server-originated queries is random, the destination port is 53, and the protocol used is UDP.

**SOA Queries**

A SOA query is a DNS query that allows a secondary name server to retrieve the revision number for the zone files on the primary name server. Based on this number, the DNS server decides whether or not to pull down the zone file by means of a zone transfer. SOA queries use UDP with the destination port of 53. On some older Bind versions, the source port is also 53.

If there are no primary name servers in the server farm, the secondary DNS servers inside the server farm send SOA queries to an outside name server. This is another reason to ensure that your design with content switches allows server-originated queries.

If the primary name server is inside the server farm, local secondary servers send SOA queries to this primary name server, without leaving the server farm.

External secondary name servers send SOA queries to the server farm: any secondary server belonging to the server farm can respond to these requests reporting its local revision number. This can potentially be a problem if you updated the configuration on the single primary name server and the update has not propagated to the secondary servers. The only solution to this problem is running multiple primary name servers with the drawback that you have to manually update all of them.

**Zone Transfers**

A secondary name server wishing to update its zone files first sends an SOA query using UDP to a primary name server, which in this design, could be a virtual primary name server. If the secondary server finds that the revision number is higher than its own, it starts a zone transfer. The zone transfers occur on TCP port 53. The following trace is a sample of a real zone transfer between a secondary (192.168.10.102) and a primary (192.168.10.101) name server:

```
192.168.10.102        192.168.10.101        DNS-UDP  Standard query SOA
192.168.10.101        192.168.10.102        DNS-UDP  Standard query response SOA
192.168.10.102        192.168.10.101        TCP      1025 > domain [SYN] Handshake
[...]
192.168.10.102        192.168.10.101        DNS-TCP  Standard query SOA
192.168.10.101        192.168.10.102        DNS-TCP  Standard query response SOA
192.168.10.102        192.168.10.101        DNS-TCP  Standard query AXFR
```
[...] TCP continues...

Note

There are two SOA queries, one carried on UDP used to compare the revision numbers, and another SOA carried on TCP followed by an AXFR query (also TCP). The AXFR query is the query that starts the zone transfer.

In this example, the primary server is a real server, not a virtual server. In the case of a primary virtual server, the SOA UDP query could go to a server different from the one that performs the actual zone transfer. This is not an issue because the secondary server requesting the update resends the SOA query on the top of TCP before the AXFR query (which uses the same TCP connection). A preferable solution is to configure the Content Switch to make sure that all UDP and TCP requests from the same client within a very short time window go to the same server (this behavior is usually called session persistence or stickiness in content switching terminology).

The primary virtual server is possibly made of a single primary real and multiple secondary real name servers. Both the primary and secondary real servers perform zone transfers. The only caveat is, as previously pointed out, the risk of having the secondary servers configuration being out of sync with the primary servers configuration.

Performance Considerations

If you are considering scaling DNS services by using content switches to load balance multiple DNS servers, it is likely that your reason for doing so is the need to increase the performance of your DNS services. Content switching performance numbers give you a reference for how well your content switch keeps up with a continuous burst of requests. In a live environment, you often have limited bursts of traffic that exceed the content switch teardown rate and eventually fall to a rate well below the teardown rate.

Content Switching Module

Three performance-limiting factors must be considered on the CSM:

- The setup rate: currently around 160,000 to 170,000 cps in release 2.2.(2a)
- The teardown rate: currently 65,000 cps in release 2.2(2a) and higher in the a forthcoming release
- The maximum number of concurrent connections: currently 1,000,000

If incoming requests exceed the 200,000 cps rate, more than one CSM is required per chassis. The CSM considers UDP traffic as connections and it expects the traffic to be bi-directional. In brief, the CSM keeps status information about UDP traffic just as it does for TCP. The main difference between TCP and UDP from a connections cache point of view is the fact that TCP explicitly closes connections while UDP does not. This requires the Content Switch to age out the UDP connections. The CSM ages UDP entries out with a timer called the “idle” timeout. A higher idle timeout value translates into a larger number of concurrent connection overtime. To check the connection table, issue the `show module csm <slot> connections` command. To check the number of simultaneous open connections (flows) to a specific VIP, issue the `show module csm <slot> vserver` command. To check the total number of simultaneous open connections in the CSM (takes all the VIPs in account), issue the `show module csm <slot> stats` command.
If you set the idle timeout to the minimum value of 4 seconds, connections are torn down after 4 seconds. The maximum teardown rate is 65,000 cps in the 2.2(2a) version of IOS. If the rate of incoming connections is below 65,000, you notice that, after 4 seconds, the number of concurrent connections has not grown. On the left of Figure 3-6, you see a continuous burst of UDP requests at a steady rate of 50,000 cps. After 4 seconds, the connection table increases to 200,000 concurrent connections.

If the connection setup rate is higher than the tear down rate (which in the current release 2.2(2a) is 65,000 cps), the number of concurrent connections increases over time. This is shown on the right of Figure 3-6. If there is a continuous burst of UDP requests at a rate of 85,000 cps, after 4 seconds, the number of concurrent connections is 340,000. The connection table continues to increase after 4 seconds, as the tear down rate is lower than the setup rate. The number of connections in the table grows at a rate of 20,000 cps until it reaches the maximum number of concurrent connections in 33s (1,000,000 = 4s x 85,000 cps + 33s x 20,000 cps).

The maximum number of concurrent connections the CSM holds is 1,000,000, regardless of whether the connections are TCP or UDP based. This is why you see an entry in the cache table for each UDP “flow.” Once you reach the maximum number of concurrent connections, the CSM drops new incoming UDP or TCP requests.

In conclusion, a single CSM running 2.2(2a) can keep up with an infinite number of continuous bursts of 65,000 UDP requests per second. Higher rates could be processed for limited amounts of time, as shown in Figure 6. For instance, a burst of 85,000 cps could last 33s before saturating the CSM. In real life environments, steady bursts of continuous connections are highly unlikely. The current range is in the 25K to 35K cps per DNS server on the Internet.

**IOS-SLB**

The main scaling limitation in the IOS-SLB for load balancing UDP traffic is the connection setup rate. The IOS-SLB code provides the “per-packet” option to increase this number. The “Per-packet” option allows you to load balance UDP traffic without keeping state information. This process NATs the traffic to the destination IP address without recording any information in the connection table. By doing this, you are able to achieve 40,000 connections per second with a CPU utilization of 50%.
# Configuration Description

## CSM Configuration

The following configuration steps are required for a virtual secondary DNS server:

1. **Step 1** Configure the CSM in either secure or one-arm mode.
2. **Step 2** Define a DNS probe for DNS servers.
3. **Step 3** Define a server farm to answer DNS query and assign the DNS probe to this server farm.
4. **Step 4** Define a virtual server (vserver) for UDP DNS requests and assign the DNS server farm to this vserver.
5. **Step 5** Configure NAT for server-originated connections.

In addition to the previous steps, the following configuration steps are required for a virtual primary DNS server:

- Define a TCP probe for the DNS servers.
- Define a server farm to perform zone transfers and assign the TCP probe to this server farm.
- Define a vserver to perform zone transfers and assign the zone transfers server farm to this vserver.

The server farm for zone transfers use the same real servers as the server farm used for regular DNS queries. The reason for a separate server farm is to support the use a different probe.

To ensure that zone transfers occur on the same server that received the original `SOA` query, you must configure sticky groups as follows:

- Define a sticky group and a mask for the client IP (/32).
- Assign the sticky group to the vserver for DNS queries.
- Assign the sticky group to the vserver for Zone Transfers.

## Probe

The following examples displays a DNS probe configuration. Replace “name” with a domain name configured on your DNS server.

```
probe DNS dns
   name www.foo.com
   interval 3
   failed 5
```

To assess the health of servers performing zone transfers, define a TCP probe as follows:

```
probe TCP_ZONE tcp
   interval 3
   failed 5
```

## Vserver

The key parameter that you must tune for your DNS servers is the idle timeout, as pointed out in the example:
vserver DNS_TEST
  virtual 10.10.10.150 udp dns
  serverfarm DNS1
  idle 4
  persistent rebalance
  inservice

NAT

Remember, you must configure source NAT on the content switch for two reasons:

1. DNS queries originated by your DNS servers (UDP)
2. Zone transfers requested by secondary name servers in your server farm (TCP)

Configure NAT to use the same IP as the Virtual IP address as follows:

   static nat virtual
       real 192.168.10.0 255.255.255.0

Configuring a subnet forces source NAT (and PAT, since it’s a many-to-one relation) for all the real servers that match that specific subnet.

IOS-SLB Configuration

The following configuration steps are required for a virtual secondary DNS server:

<table>
<thead>
<tr>
<th>Step 1</th>
<th>Configure the IOS-SLB in directed mode.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 2</td>
<td>Define a DNS probe for DNS servers.</td>
</tr>
<tr>
<td>Step 3</td>
<td>Define a server farm to answer DNS queries and assign the DNS probe to this server farm, enable “nat server.”</td>
</tr>
<tr>
<td>Step 4</td>
<td>Define a per-packet virtual server (vserver) for UDP DNS requests and assign the DNS server farm to this vserver.</td>
</tr>
<tr>
<td>Step 5</td>
<td>Configure per-packet NAT for UDP return traffic with the same IP address as the Virtual IP address.</td>
</tr>
<tr>
<td>Step 6</td>
<td>Configure NAT for server-initiated connections. Per-packet NAT cannot be reused because zone transfers require TCP.</td>
</tr>
</tbody>
</table>

In addition to the previous steps, the following configuration steps are required for a virtual primary DNS server:

- Define a TCP probe for the DNS servers.
- Define a server farm to perform zone transfers and assign the TCP probe to this server farm.
- Define a vserver to perform zone transfers and assign the zone transfers server farm to this vserver.

Note

The server farm for zone transfer uses the same real servers as the server farm for regular DNS queries. The reason for a separate server farm is to support a different probe.

To ensure that zone transfers occur on the same server that received the original SOA query, you must configure sticky groups with the restriction that you have to use ping probes as opposed to DNS or TCP probes. This requirement exists because for sticky to work, you must use the same server farm across
vservers. Therefore, having two vservers, one for TCP and one for UDP both using the same server farm makes it impossible to configure different probes: you would have to choose either TCP probes or UDP probes at which point the correct choice is an ICMP probe.

The sticky configuration is as follows:

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>Define a “ping” probe.</td>
</tr>
<tr>
<td>Step 2</td>
<td>Define a single server farm (as opposed to one for DNS queries and one for Zone Transfers).</td>
</tr>
<tr>
<td>Step 3</td>
<td>Define a sticky group to the vserver for DNS queries with a mask of 32.</td>
</tr>
<tr>
<td>Step 4</td>
<td>Define the same sticky group to the vserver for zone transfers with a mask of 32.</td>
</tr>
</tbody>
</table>

**Probe**

To ensure stickiness between SOA queries and AXFR queries (zone transfers), use a simple “ping” probe. See the “CSM Configuration” section for more information on probes.

**Server Farms**

You must configure “nat server” in the server farm to use directed mode.

To ensure stickiness between the DNS server farm and the zone server farm, you must use a single server farm for both DNS and zone transfers, which requires you to use the same probe. In this instance, ICMP probes are recommended.

**Vserver Per-Packet**

The virtual server configuration for DNS queries is purposely setup as stateless, which enables a higher connection setup rate. Use the” per-packet” option to create the stateless vserver. When this virtual server receives a DNS request, the IOS-SLB code selects a server and performs NAT. Since the stateless configuration is used, you must configure server NAT to ensure the responses are sent back to the correct server.

```
vserver DNS_TEST
  virtual 10.10.10.150 udp dns service per-packet
  serverfarm DNS1
  persistent rebalance
  inservice standby SERVER_VLAN
```

**NAT Per-Packet**

The following configuration is used to translate server responses back to the virtual IP; it is not meant to allow server initiated connections. These responses are to the queries directed to the virtual IP. This configuration is required only to take advantage of the higher performance gains achieved by using the “per-packet” option so unless the per-packet option is used, the NAT to the VIP address process is not required.

```
ip slb static nat 10.10.10.150 per-packet
  real 192.168.10.104 53
  real 192.168.10.103 53
  real 192.168.10.102 53
  real 192.168.10.101 53
  !
```
Note

Port 53 ensures that NAT applies only to DNS responses.

NAT for Server Initiated Connections

Configuring NAT for server-originated connections is not a one-line configuration. Currently, as of 12.1(1lb)E2, the IOS-SLB configuration requires a virtual server and a server farm in addition to NAT. Omitting the virtual server results in NAT operating only on outgoing connections. See the caveat section at the end of this chapter for details on NAT discards on return traffic (CSCdx62349). Remember, you need NAT for server initiated connections for the following reasons:

- DNS queries originated by the DNS servers (UDP)
- Zone transfers requested by secondary servers (TCP)

Since the NAT configuration requires a different set of virtual servers for TCP and UDP, you must select a different IP address per VIP per protocol.

Configuring NAT for IOS-SLB requires the following steps:

---

**Step 1** List all of your real servers under the “static nat” configuration.

```plaintext
ip slb static nat 10.10.10.151
real 192.168.10.104
real 192.168.10.103
real 192.168.10.102
real 192.168.10.101
```

**Step 2** Configure a server farm with the same real servers as in the static NAT configuration.

```plaintext
ip slb serverfarm JUST-FOR-NAT
nat server
real 192.168.10.101
inservice
!
real 192.168.10.102
inservice
!
real 192.168.10.103
inservice
!
real 192.168.10.104
inservice
```

**Step 3** Configure a vserver with the NAT IP address.

```plaintext
ip slb vserver JUST-FOR-NAT
virtual 10.10.10.151 tcp 0
serverfarm JUST-FOR-NAT
inservice standby SERVER_VLAN
```

---

Vserver for Zone Transfers

Configure the vserver for zone transfers the same as you did on the CSM. Do not use the “per-packet” option for the following two reasons:

- TCP load balancing cannot be done by NATing the IP addresses
State information must be maintained

```plaintext
ip slb vserver ZONE-TRANSFERS
virtual 10.10.10.150 tcp 53
serverfarm ZONE_TRANSFERS1
inservice standby SERVER_VLAN
```

**Redundancy**

It is out of the scope of this chapter to explain redundancy with IOS-SLB. However, to clarify the configurations, notice that “**inservice standby SERVER_VLAN**” binds the status of a vserver to the status of the backend VLAN that provides the default gateway to the servers. If the MSFC is no longer the default gateway for the servers, the vserver gives up mastership.

Stateful failover is configured only for the zone transfer vserver because the state information for the UDP requests is not maintained.

**Sample Configurations**

This section provides the relevant portions of the configurations used for testing the design presented in this chapter. The details of the infrastructure are outside of the scope of this chapter.

**CSM Configuration**

```plaintext
!
static nat virtual
 real 192.168.10.0 255.255.255.0
!
probe DNS dns
 name www.foo.com
 interval 3
 failed 5
!
probe TCP_ZONE tcp
 interval 3
 failed 5
!
serverfarm DNS1
 nat server
 no nat client
 real 192.168.10.101
 inservice
 real 192.168.10.102
 inservice
 real 192.168.10.103
 inservice
 real 192.168.10.104
 inservice
 probe DNS
!
serverfarm ZONE_TRANSFERS1
 nat server
 no nat client
 real 192.168.10.101
 inservice
 real 192.168.10.102
```
in service
real 192.168.10.103
in service
real 192.168.10.104
in service
probe TCP_ZONE
!
sticky 5 netmask 255.255.255.255
!
vserver DNS_TEST
virtual 10.10.10.150 udp dns
serverfarm DNS1
idle 4
sticky 60 group 5
persistent rebalance
in service
!
vserver ZONE_TRANSFERS
virtual 10.10.10.150 tcp 53
serverfarm ZONE_TRANSFERS1
sticky 60 group 5
persistent rebalance
in service
!
ft group 1 vlan 100
priority 20
preempt
!

**IOS-SLB configuration**

!

ip slb probe DNS dns
lookup www.foo.com
interval 3
faildetect 5
!
ip slb probe TCP_ZONE tcp
interval 3
faildetect 5
!
ip slb serverfarm DNS1
nat server
probe DNS
!
real 192.168.10.101
in service
!
real 192.168.10.102
in service
!
real 192.168.10.103
in service
!
real 192.168.10.104
in service
!
ip slb serverfarm JUST-FOR-NAT
nat server
real 192.168.10.101
in service
!
real 192.168.10.102
  inservice
!
real 192.168.10.103
  inservice
!
real 192.168.10.104
  inservice
!
  ip slb serverfarm ZONE_TRANSFERS1
  nat server
  probe TCP_ZONE
!
real 192.168.10.101
  inservice
!
real 192.168.10.102
  inservice
!
real 192.168.10.103
  inservice
!
real 192.168.10.104
  inservice
!
  ip slb vserver JUST-FOR-NAT-TCP
  virtual 10.10.10.151 tcp 0
  serverfarm JUST-FOR-NAT
  inservice standby SERVER_VLAN
!
  ip slb vserver JUST-FOR-NAT-UDP
  virtual 10.10.10.151 udp 0
  serverfarm JUST-FOR-NAT
  inservice standby SERVER_VLAN
!
  ip slb vserver DNS_TEST
  virtual 10.10.10.150 udp dns service per-packet
  serverfarm DNS1
  inservice standby SERVER_VLAN
!
  ip slb vserver ZONE_TRANSFERS
  virtual 10.10.10.150 tcp dns
  serverfarm ZONE_TRANSFERS1
  replicate casa 10.6.0.2 10.6.0.3 65001
  inservice standby SERVER_VLAN
!
  ip slb static nat 10.10.10.150 per-packet
real 192.168.10.104 53
real 192.168.10.103 53
real 192.168.10.102 53
real 192.168.10.101 53
!
  ip slb static nat 10.10.10.151
real 192.168.10.104
real 192.168.10.103
real 192.168.10.102
real 192.168.10.101
!

**IOS-SLB with Stickiness**
ip slb probe ICMP ping
  interval 3
  faildetect 5
!
ip slb serverfarm DNS1
  nat server
  probe ICMP
!
  real 192.168.10.101
    inservice
!
  real 192.168.10.102
    inservice
!
  real 192.168.10.103
    inservice
!
  real 192.168.10.104
    inservice
!
ip slb vserver JUST-FOR-NAT-TCP
  virtual 10.10.10.151 tcp 0
  serverfarm DNS1
  inservice standby SERVER_VLAN
!
ip slb vserver JUST-FOR-NAT-UDP
  virtual 10.10.10.151 udp 0
  serverfarm DNS1
  inservice standby SERVER_VLAN
!
ip slb vserver DNS_TEST
  virtual 10.10.10.150 udp dns service per-packet
  serverfarm DNS1
  sticky 60 group 5
  inservice standby SERVER_VLAN
!
ip slb vserver ZONE_TRANSFERS
  virtual 10.10.10.150 tcp dns
  serverfarm ZONE_TRANSFERS1
  sticky 60 group 5
  replicate casa 10.6.0.2 10.6.0.3 65001
  inservice standby SERVER_VLAN
!
ip slb static nat 10.10.10.150 per-packet
  real 192.168.10.104 53
  real 192.168.10.103 53
  real 192.168.10.102 53
  real 192.168.10.101 53
!
ip slb static nat 10.10.10.151
  real 192.168.10.104
  real 192.168.10.103
  real 192.168.10.102
  real 192.168.10.101
!

Testing the Setup

Test this setup with a client that sends DNS requests to the virtual server defined on the CSM or IOS-SLB. There are two commands available to test the setup from your client:
• dig (available on Unix machines)
• nslookup (also available at the Windows DOS prompt)

In these examples, the root zone configuration was loaded into the DNS servers so the Name Server could be queried for the records “com.”, “it.”, etc.

**DNS Query Type NS, A, SOA**

From the Windows prompt, launch nslookup and perform the following steps:

**Step 1**
Type “server” followed by the virtual server address (in our case was 10.10.10.150).

**Step 2**
Set the query as an NS query “set q=ns” or an A query “set q=a” or any type of query you want to perform.

**Step 3**
Type the domain you want to resolve.

Nslookup sends the query to the virtual server. In this example, “it” was selected as the domain (do not forget the trailing dot).

```plaintext
> it.
Server: [10.10.10.150]
Address: 10.10.10.150
Non-authoritative answer:
it nameserver = NS.RIPE.NET
it nameserver = SERVER2.INFN.it
it nameserver = DNS2.IUNET.it
it nameserver = NAMESERVER.CNR.it
it nameserver = DNS.NIC.it
it nameserver = DNS2.IT.NET
it nameserver = NS2.PSI.NET
it nameserver = NS.EU.NET
NS.RIPE.NET internet address = 193.0.0.193
SERVER2.INFN.it internet address = 131.154.1.3
DNS2.IUNET.it internet address = 192.106.1.31
NAMESERVER.CNR.it internet address = 194.119.192.34
DNS.NIC.it internet address = 193.205.245.5
DNS2.IT.NET internet address = 151.1.2.1
NS2.PSI.NET internet address = 38.8.50.2
NS.EU.NET internet address = 192.16.202.11
```

You can do the same test with “dig,” the syntax is:

• dig [@server] [q-type] [domain]

For an example, query the NS records for “.com”:

```
[admin@localhost ~]$ dig @10.10.10.150 ns com.
; <<>> DiG 8.2 <<>> @10.10.10.150 ns com.
; (1 server found)
; res options: init recurs defnam dnsrch
; got answer:
; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 6
; flags: qr rd ra; QUERY: 1, ANSWER: 13, AUTHORITY: 0, ADDITIONAL: 13
; QUERY SECTION:
; com, type = NS, class = IN

; ANSWER SECTION:
com. 2D IN NS J.GTLD-SERVERS.NET.
com. 2D IN NS K.GTLD-SERVERS.NET.
```
Repeat the same type of test to verify that A and SOA queries work. These queries provide verification that the virtual servers defined for UDP work correctly.

Verify that the UDP “connections” are setup on the CSM console.

```
AGG1#show module csm 4 conns
prot vlan source         destination           state
----------------------------------------------------------------------
In  UDP  6    10.15.0.32:1027       10.10.10.150:53       ESTAB
Out UDP  6    192.168.10.104:53     10.15.0.32:1027       ESTAB
```

### Testing Zone Transfers

Test zone transfers by using either a query type AXFR or by removing the zone files from `/var/named` and reloading named (“ndc reload”). Verify that named restarts and pulls the configuration from the primary server (which is your virtual server). The following example is of the first type of test.

```
[admin@localhost ~]$ dig @10.10.10.150 axfr | more
; <<>> DiG 8.2 <<>> @10.10.10.150 axfr
; (1 server found)
$ORIGIN ..

  1D IN SOA       A.ROOT-SERVERS.NET (...
  6D IN NS        A.ROOT-SERVERS.NET.
A.ROOT-SERVERS.NET. 6D IN A         192.168.10.11
@                       6D IN NS        H.ROOT-SERVERS.NET.
H.ROOT-SERVERS.NET.     6D IN A         128.63.2.53
@                       6D IN NS        C.ROOT-SERVERS.NET.
```

### Caveats

There are some caveats to setting up DNS load balancing in your data center.

1. The IOS-SLB does not support the restoration of the configuration of TCP port 53 (used for zone transfers) after a reload. The release notes for bug id CSCdx61788 define the workaround.

2. In addition, when using IOS-SLB, setting up NAT for server-originated connections requires the configuration of a virtual-server. Setting up NAT opens the door for unwanted UDP/TCP traffic to the server farm. The bug id is CSCdx62349.

3. When using IOS-SLB, implementing stickiness between SOA and AXFR queries requires using “ping” probes instead of Layer 4 probes (like DNS or TCP probes). This application note explains both configurations.
CHAPTER 4

RADIUS Server Load Balancing

This chapter describes how to use Cisco IOS Server Load Balancing (SLB) for load balancing Remote Access Dial-In User Service (RADIUS) servers. It pays special attention to the requirements for demanding environments, such as an 802.1x support scenario, where user authentication is required for network connectivity and VLAN assignment. This chapter also provides some information about the RADIUS-specific extensions included in IOS SLB for supporting mobile wireless, such as a sticky database of usernames. However, a mobile wireless deployment has its own set of RADIUS requirements that are not fully described in this document. This chapter includes the following sections:

- The RADIUS Protocol
- The 802.1x Protocol
- IOS SLB
- IOS SLB in the Data Center
- Conclusion

The RADIUS Protocol

In the current Information Technology environment, RADIUS servers have become increasingly important for user authentication and session accounting. As such, it is paramount to provide sufficient RADIUS server capacity and ensure the high availability of RADIUS services. You can use Cisco IOS SLB in conjunction with multiple RADIUS server instances, to satisfy these requirements. In addition, it is oftentimes desirable to direct RADIUS clients to the same RADIUS server for their authentication and accounting services. IOS SLB is capable of satisfying this requirement also.

RADIUS is a general-purpose access server technology providing for authentication and accounting in a variety of applications. The original specification and implementation of RADIUS was completed in 1992 by Steve Willens of Livingston Enterprises, Inc. Cisco implements RADIUS into its Cisco Secure Access Control Server (ACS) software.

The RADIUS authentication protocol is documented in RFC 2058, while the accounting protocol is documented separately in RFC 2059. However, you can use them together for a comprehensive solution. RADIUS is a connectionless service, which means that issues related to server availability, retransmission, and timeouts are handled by RADIUS-enabled devices rather than by the transmission protocol. The RADIUS protocol is transaction-based and uses UDP port 1812 for authentication and UDP port 1813 for accounting. Previous versions of RADIUS used UDP ports 1645 and 1646.

The RADIUS authentication process may require multiple packet exchanges between client and server to achieve final authentication. Subsequent RADIUS client-initiated accounting requests are typically completed through a single two-packet exchange, but this may occur hours or days later during the
RADIUS client/server session. For this reason, RADIUS traffic from a specific client should be directed to the same RADIUS server, and a RADIUS load-balancing solution must take this requirement into account.

Figure 4-1 shows the basic packet flow between RADIUS client and server.

**Figure 4-1 RADIUS Access (Authentication) and Accounting Packet Flow**

---

### The 802.1x Protocol

The IEEE 802.1x standard defines a client/server access control and authentication protocol that restricts unauthorized devices from connecting to a LAN through publicly accessible ports. The 802.1x protocol controls network access by creating two distinct virtual access points at each port. One access point is an uncontrolled port; the other is a controlled port. All traffic through the single port is available to both access points.

802.1x authenticates each user device that is connected to a switch port. It can assign the port to a VLAN before making available any services that are offered by the switch or the LAN. Until a device is authenticated, 802.1x access controls restrict traffic through the port to which the device is connected; only Extensible Authentication Protocol over LAN (EAPOL) traffic is permitted. After authentication is completed, enterprise traffic is allowed through the port.

An 802.1x environment consists of devices operating in one of three roles:

- **Supplicant (host workstation)**—Requests access to the LAN and switch services and responds to requests from the switch. The workstation must be running 802.1x-compliant software.

- **Switch**—Controls the physical access to the network based on the authentication status of the host. Because the switch acts as an intermediary (proxy), the actual RADIUS authentication sequence is transparent to the supplicant. The switch performs the following operations:
  - Requests identity information (credentials) from the supplicant
  - Verifies credentials using the RADIUS server
  - Relays a response to the supplicant.
  - Encapsulates and decapsulates the EAP frames and interacts with the RADIUS server as a client.
• **Authentication server**—Performs the actual authentication of the supplicant. The authentication server is a RADIUS server that validates the identity of the supplicant and notifies the switch whether or not the supplicant is authorized to access the LAN and switch services. The authentication server can also send information to the switch about the VLAN that is assigned to the user.

Figure 4-2 depicts a typical 802.1x sequence.

**Figure 4-2  802.1x Authentication Sequence**

The switch begins RADIUS authentication by sending IP datagrams to UDP destination port 1812 on the RADIUS server. After successful authentication, the switch sends `session-start` and `session-stop` accounting records to the server using UDP destination port 1813.

Wireless users may encounter problems with 802.1x re-authentication while roaming between access points that specify different RADIUS servers. IOS SLB has some extensions to help address this problem, and there are other initiatives in development, including:

- WLAN aggregators that generate all RADIUS traffic on behalf of several access points
- Session-state servers that can be queried by any Cisco Secure ACS server for the state of any session
- An IEEE standard that allows passing a supplicant’s MAC address as a RADIUS attribute during RADIUS exchanges. This is of interest because future RADIUS load balancers may use a supplicant’s MAC address as the load-balancing determinant and sticky database index.

**IOS SLB**

IOS SLB allows an enterprise to represent a collection of similar servers as a single server instance, called a server farm. A server farm consists of multiple physical servers that are represented by a single virtual server. An IOS SLB virtual server is simply an IP address in the IOS SLB platform configuration.
IOS SLB fields the request and distributes the processing to the physical servers, which is how it balances the load. Clients typically learn the IP address of the virtual server through a DNS forward lookup, but the IP address may also be included in the client configuration. Virtual servers may be TCP-based, such as FTP or HTTP, or UDP-based, such as DNS or RADIUS.

IOS SLB provides the following benefits:

- High performance and availability by distributing client requests across a cluster of servers.
- Improved ease of administration because clients know only about virtual servers, and no reconfiguration is required when a physical server changes.
- Improved security for the physical server because its address is never announced to the external network. Users are only aware of the virtual IP address. Also, unwanted traffic can be filtered using both IP addresses and IP port numbers.
- Ease of maintenance with no downtime because physical servers can be transparently maintained.

**IOS SLB Basics**

Figure 4-3 illustrates a basic IOS SLB network connection to a generic server farm. The switch running IOS SLB interconnects a router and a number of servers, and balances the load among the servers.

An IP address for the virtual server is inserted as a host entry into the routing table of the switch. Propagation of this IP address is controlled by the enterprise routing protocol.

**Figure 4-3 Basic IOS SLB Schematic**

The configuration for the SLB implementation illustrated in Figure 4-3 looks like this:

```plaintext
ip slb serverfarm WEBFARM
nat server
! real 10.1.3.1
   inservice
! real 10.1.3.2
   inservice
! real 10.1.3.3
   inservice
```
IOS SLB Buddied Virtual Servers

This concept of teamed virtual servers is referred to as buddied virtual servers. To meet the requirement that access and accounting records from a specific client go to the same RADIUS server, configure both virtual servers in the same sticky group. A sample configuration is shown below.

```sh
ip slb serverfarm RADFARM
  nat server
  !
  real 10.1.3.11
  inservice
  !
  real 10.1.3.12
  inservice
  !
  real 10.1.3.13
  inservice
!
ip slb vserver RAD1812
  virtual 10.170.1.1 udp 1812
  serverfarm RADFARM
  sticky 86400 group 10
  inservice
!
ip slb vserver RAD1813
  virtual 10.170.1.1 udp 1813
  serverfarm RADFARM
  sticky 86400 group 10
  inservice
```

In this configuration, both virtual servers are configured with the same sticky group (group 10), and the same server farm (RAFDARM). When a client first accesses one of the virtual servers, that client’s IP address is added to the IOS SLB database for group 10, where the address is associated with the physical server chosen for the first RADIUS access request. Thereafter, requests from that client for either virtual server, will always go to the same physical server. This configuration causes the sticky database to store its entries for 86,400 seconds of inactivity. That value can be configured up to 1,000,000,000 seconds—essentially forever.

IOS SLB in the Data Center

IOS SLB can be deployed as illustrated in the basic scenario described in the section “IOS SLB Basics.” However, in larger data centers, IOS SLB is typically deployed in redundant fashion at the aggregation layer, as a front-end for Internet, intranet, or extranet server farms. Figure 4-4 illustrates a topology that provides redundancy for all single failures and some double failures.
While IOS SLB can load balance RADIUS servers and provide for the high availability of RADIUS services, an IOS SLB device can represent a single point of failure. Fortunately, mechanisms within IOS SLB provide for redundancy, including stateful backup, which is required for sticky load balancing.

Stateful backup enables IOS SLB to maintain the state between primary and standby switches by periodically communicating the load-balancing decision parameters. Stateful backup provides IOS SLB with a one-to-one, idle backup scheme, in which one instance of IOS SLB is handling client and server flows at a given time, and no more than one backup platform exists for each active IOS SLB switch.

The Hot Standby Router Protocol (HSRP) is a rich, robust protocol that can be configured to provide great resilience in complex and varied topologies. For example, in newer versions of IOS, an HSRP router can track interface statuses and IP routes and adjust its HSRP priorities accordingly. This is a powerful capability.

IOS SLB tracks and follows the active HSRP router. The IOS SLB standby switch is configured as a lower-priority HSRP router and maintains its virtual servers in a dormant state until its HSRP process detects a failure and assumes the active router role. The IOS SLB standby switch monitors the HSRP failover, follows suit, and assumes the IOS SLB active role by advertising virtual addresses and processing flows. HSRP timers control how quickly the failover is initiated. An example configuration for the scenario illustrated in Figure 4-4 is shown below,

```
interface Loopback1
  description "** For use by IOS SLB replication process **"
  ip address 192.168.1.1 255.255.255.0

ip slb vserver RAD1812
  virtual 10.170.1.1 udp 1812
  serverfarm RADFARM
  sticky 86400 group 10
  replicate casa 192.168.1.1 192.168.2.1 2003
  inservice standby CaminoReals

int Vlan11
  description "** Real Server side **"
  ip address 10.1.1.2 255.255.255.0
```
standby 1 ip 10.1.1.1
standby 1 timers 1 3
**standby 1 priority 51**
standby 1 preempt delay minimum 180 sync 300
standby 1 name CaminoReals

This configuration is for the primary HSRP/IOS SLB switch. The command line `standby 1 priority 51` sets a higher priority for this server than the configuration for the standby switch, which is not shown but has a priority of 50. This configuration causes the primary to preempt, meaning that, all other things being equal, it asserts itself as the active HSRP router. Configuring the primary to preempt provides a more deterministic deployment but is not a strict requirement.

During normal operation, the IOS SLB primary switch sends incremental updates to the sticky database on the standby switch, using the destination IP address 192.168.2.1 and port 2003. CASA is an acronym for Cisco Appliance Service Architecture, humorously called the Maytag protocol. If the primary switch fails, the standby takes over with complete statefulness.

The HSRP group is given a name, which, in this example is “CaminoReals.” IOS SLB uses the group name to track the group on behalf of its virtual server, which, in this example is “RAD1812.” When a transition occurs in relation to the active HSRP router, IOS SLB follows suit and assumes the same role as the switch in its HSRP group.

In this example, the primary switch is configured to preempt, which means that even when the standby switch is functioning and fully operational, the primary will wrest control away from it. The standby switch is never configured to preempt. Consider the following sequence of events. The primary switch fails or is intentionally removed from service. The standby immediately takes over the active role. HSRP transitions first and IOS SLB follows. When the primary is restored to service, its HSRP process preempts and takes control away from the standby, and IOS SLB follows again.

Two problems can occur if the primary preempts immediately upon restoration of service. First, the primary switch may not be completely initialized and second, the sticky database may not be fully populated. To prevent these problems, set the **minimum** and **sync** options using the **preempt delay** command. The minimum delay gives the primary switch time to fully initialize and the sync delay gives time to fully populate the sticky database.

In this example, the minimum delay is 180 seconds and the sync delay is 300 seconds. Given these values, after being restored to service, the primary switch waits 180 seconds before querying the standby switch for all the database entries for all its recovering virtual servers. After initiating this request, the primary waits another 300 seconds, as specified by the sync delay, before wresting control from the standby switch.

In a failover scenario, speed is of the essence. However, in a **failback** scenario, where the primary has been restored and is preempting, speed is typically **not** an issue. Allowing generous delays assures that the switch initialization and IOS SLB synchronization processes are completed before failback occurs. The 180 and 300-second delays used in this example are probably excessive, but longer delays are helpful if when trying to read debug output messages.
iOS SLB Test Results

Proof-of-concept testing was performed to verify all aspects of the configurations included in this application note. In addition, limited performance tests were run against IOS SLB. Figure 4-5 depicts the network topology for the proof-of-concept and performance tests. This is the ESE “Private Dancer” test bed (a.k.a. CHUB), with extensions.

The north side of the topology shows the clients in these tests. The default gateway for the clients is 10.3.1.1, which is an IP address on the VLAN 10 interface of the northernmost switch.

The center of the topology shows dual Catalyst 6500 units in a redundant topology. The test bed was customized to test a primary/standby deployment of HSRP/IOS SLB instances. The two Catalyst 6500 units with MSFC2 run Native IOS Version 12.1(13)e. An ESE standard test HTTP transaction, illustrated in Figure 4-6, is used to measure IOS SLB performance.

The south side of the topology depicts the servers. Their default gateway is 10.1.1.6 – the HSRP shared address in VLAN 11. VLAN 11 is a spanning tree domain.

Gigabit EtherChannel links are used throughout to ensure that IOS SLB is the bottleneck for this test.
ESE has conducted server load balancing tests against the Content Switching Module (CSM) and the Content Services Switch (CSS). Complete test results for these platforms are documented in EDCS-257300 and EDCS-300527, respectively. The CSM and CSS can load balance Layer 4 and Layer 5 servers; that is TCP/UDP port numbers and URL strings. With minor exceptions, IOS SLB is a Layer 4 load balancing protocol. Layer 4 connections per second (cps) results are 85,000 cps for the CSM, 34,000 cps for a fully populated CSS, and 14,000 cps for the tested IOS SLB platform. At 14,000 cps, the tested MSFC2 registered 87% CPU utilization.

Further tests revealed that the performance impact of stateful replication is approximately 4% CPU utilization. The probe impact on CPU utilization was negligible.

**IOS SLB Selected Debug Output**

During failover, runtime debug output messages were captured on the standby switch. They are shown below:

```
Jul 9 10:19:25: SB1: Vl11 Hello out 10.1.1.8 Standby pri 50 ip 10.1.1.6
Jul 9 10:19:52: %DUAL-5-NBRCHANGE: IP-EIGRP 100: Neighbor 10.1.1.9 (Vlan11) is down: holding time expired
Jul 9 10:19:53: %DUAL-5-NBRCHANGE: IP-EIGRP 100: Neighbor 10.2.1.1 (Vlan14) is down: holding time expired
Jul 9 10:20:01: SB1: Vl11 Standby: c/Active timer expired (10.1.1.9)
Jul 9 10:20:01: SB1: Vl11 Active router is local, was 10.1.1.9
Jul 9 10:20:01: SB1: Vl11 Standby router is unknown, was local
Jul 9 10:20:01: SB1: Vl11 Standby -> Active
Jul 9 10:20:01: %STANDBY-6-STATECHANGE: Vlan11 Group 1 state Standby -> Active
Jul 9 10:20:01: SLB_VS_DEBUG: RAD1812 event= STANDBY_UP, state= STANDBY -> OPERATIONAL
```

In the above output, the IOS SLB process on the standby switch becomes operational for virtual server RAD1812 after the switch becomes the active HSRP router. As a result, a routing table entry for the virtual server is inserted into the routing table. This is displayed below.

```
10.0.0.0/8 is variably subnetted, 5 subnets, 3 masks
 C 10.8.0.0/16 is directly connected, Vlan8
 D 10.3.0.0/16 [90/3072] via 10.8.1.2, 00:09:02, Vlan8
 C 10.1.0.0/16 is directly connected, Vlan11
 S 10.170.1.1/32 is directly connected, Null0
 C 192.168.2.0/24 is directly connected, Loopback1
```

1. IP addresses in the trace do not correlate with IP addresses in Figure5. Please disregard.
The following debug output shows a failback scenario from the perspective of the HSRP primary switch. The primary switch has just reloaded and is wresting control back from the standby. Many messages are omitted from this example for the sake of brevity. The bold typeface messages highlight the most important stages in the sequence of operations.

*Jul  9 14:16:21: %OIR-SP-6-INSCARD: Card inserted in slot 5, interfaces are now online
*Jul  9 14:16:30: %DUAL-5-NBRCHANGE: IP-EIGRP 100: Neighbor 10.1.1.8 (Vlan11) is up: new adjacency
*Jul  9 14:16:36: SB1: Vl11 Active router is 10.1.1.8
*Jul  9 14:16:36: SB1: Vl11 Speak: h/Hello rcvd from lower pri Active router (50/10.1.1.8)
*Jul  9 14:16:36: SB1: Vl11 Starting min preempt delay (180 secs)
*Jul  9 14:16:44: SB1: Vl11 Standby router is local
.Jul  9 14:17:19: SB1: Vl11 Hello out 10.1.1.9 Standby pri 51 ip 10.1.1.6
.Jul  9 14:17:19: SB1: Vl11 Hello in 10.1.1.8 Active pri 50 ip 10.1.1.6
Jul  9 14:19:45: SB1: Vl11 Min preempt delay expired
Jul  9 14:19:45: SB1: Vl11 Starting sync preempt delay (300 secs)
Jul  9 14:19:45: SLB Replicate: (send) req for conn table vs: RADFARM
Jul  9 14:19:45: SLB Replicate: (rcvd) update, vs: RADFARM
Jul  9 14:19:45: v_ip prot client real
state timeout
Jul  9 14:19:45:

<table>
<thead>
<tr>
<th>v_ip</th>
<th>prot client</th>
<th>real</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0.0.0:0</td>
<td>0</td>
<td>10.3.130.3:0</td>
</tr>
<tr>
<td>0.0.0.0:0</td>
<td>0</td>
<td>10.3.130.38:0</td>
</tr>
<tr>
<td>0.0.0.0:0</td>
<td>0</td>
<td>10.3.130.46:0</td>
</tr>
<tr>
<td>0.0.0.0:0</td>
<td>0</td>
<td>10.3.130.46:0</td>
</tr>
<tr>
<td>0.0.0.0:0</td>
<td>0</td>
<td>10.3.130.46:0</td>
</tr>
<tr>
<td>0.0.0.0:0</td>
<td>0</td>
<td>10.3.130.46:0</td>
</tr>
<tr>
<td>0.0.0.0:0</td>
<td>0</td>
<td>10.3.130.46:0</td>
</tr>
<tr>
<td>0.0.0.0:0</td>
<td>0</td>
<td>10.3.130.46:0</td>
</tr>
<tr>
<td>0.0.0.0:0</td>
<td>0</td>
<td>10.3.130.46:0</td>
</tr>
<tr>
<td>0.0.0.0:0</td>
<td>0</td>
<td>10.3.130.46:0</td>
</tr>
<tr>
<td>0.0.0.0:0</td>
<td>0</td>
<td>10.3.130.46:0</td>
</tr>
<tr>
<td>0.0.0.0:0</td>
<td>0</td>
<td>10.3.130.46:0</td>
</tr>
<tr>
<td>0.0.0.0:0</td>
<td>0</td>
<td>10.3.130.46:0</td>
</tr>
<tr>
<td>0.0.0.0:0</td>
<td>0</td>
<td>10.3.130.46:0</td>
</tr>
<tr>
<td>0.0.0.0:0</td>
<td>0</td>
<td>10.3.130.46:0</td>
</tr>
<tr>
<td>0.0.0.0:0</td>
<td>0</td>
<td>10.3.130.46:0</td>
</tr>
<tr>
<td>0.0.0.0:0</td>
<td>0</td>
<td>10.3.130.46:0</td>
</tr>
<tr>
<td>0.0.0.0:0</td>
<td>0</td>
<td>10.3.130.46:0</td>
</tr>
<tr>
<td>0.0.0.0:0</td>
<td>0</td>
<td>10.3.130.46:0</td>
</tr>
<tr>
<td>0.0.0.0:0</td>
<td>0</td>
<td>10.3.130.46:0</td>
</tr>
<tr>
<td>0.0.0.0:0</td>
<td>0</td>
<td>10.3.130.46:0</td>
</tr>
<tr>
<td>0.0.0.0:0</td>
<td>0</td>
<td>10.3.130.46:0</td>
</tr>
<tr>
<td>0.0.0.0:0</td>
<td>0</td>
<td>10.3.130.46:0</td>
</tr>
<tr>
<td>0.0.0.0:0</td>
<td>0</td>
<td>10.3.130.46:0</td>
</tr>
<tr>
<td>0.0.0.0:0</td>
<td>0</td>
<td>10.3.130.46:0</td>
</tr>
<tr>
<td>0.0.0.0:0</td>
<td>0</td>
<td>10.3.130.46:0</td>
</tr>
<tr>
<td>0.0.0.0:0</td>
<td>0</td>
<td>10.3.130.46:0</td>
</tr>
<tr>
<td>0.0.0.0:0</td>
<td>0</td>
<td>10.3.130.46:0</td>
</tr>
<tr>
<td>0.0.0.0:0</td>
<td>0</td>
<td>10.3.130.46:0</td>
</tr>
<tr>
<td>0.0.0.0:0</td>
<td>0</td>
<td>10.3.130.46:0</td>
</tr>
<tr>
<td>0.0.0.0:0</td>
<td>0</td>
<td>10.3.130.46:0</td>
</tr>
<tr>
<td>0.0.0.0:0</td>
<td>0</td>
<td>10.3.130.46:0</td>
</tr>
<tr>
<td>0.0.0.0:0</td>
<td>0</td>
<td>10.3.130.46:0</td>
</tr>
</tbody>
</table>
Below you see the same scenario from the perspective of the standby switch. Control is wrested from it. Many messages are omitted for brevity and clarity.

Jul 9 10:31:45: SB1: Vl11 Hello out 10.1.1.8 Active pri 50 ip 10.1.1.6
Jul 9 10:31:53: %DUAL-5-NBRCHANGE: IP-EIGRP 100: Neighbor 10.1.1.9 (Vlan11) is up: new adjacency
Jul 9 10:32:04: SB1: Vl11 Hello out 10.1.1.8 Active pri 50 ip 10.1.1.6
Jul 9 10:32:25: %DUAL-5-NBRCHANGE: IP-EIGRP 100: Neighbor 10.1.1.9 (Vlan11) is up: new adjacency
Jul 9 10:32:27: SB1: Vl11 Hello in 10.1.1.9 Speak pri 51 ip 10.1.1.6
Jul 9 10:32:34: SB1: Vl11 Hello out 10.1.1.8 Active pri 50 ip 10.1.1.6
Jul 9 10:32:36: SB1: Vl11 Hello in 10.1.1.9 Speak pri 51 ip 10.1.1.6
Jul 9 10:32:37: SB1: Vl11 Hello in 10.1.1.9 Standby pri 51 ip 10.1.1.6
Jul 9 10:32:37: SB1: Vl11 Standby router is 10.1.1.9
Jul 9 10:32:44: SB1: Vl11 Hello out 10.1.1.8 Active pri 50 ip 10.1.1.6
Jul 9 10:32:57: SB1: Vl11 Hello in 10.1.1.9 Standby pri 51 ip 10.1.1.6
Jul 9 10:35:30: SB1: Vl11 Hello in 10.1.1.9 Standby pri 51 ip 10.1.1.6
Jul 9 10:35:34: SB1: Vl11 Hello in 10.1.1.9 Standby pri 51 ip 10.1.1.6
Jul 9 10:35:37: SB1: Vl11 Hello in 10.1.1.9 Standby pri 51 ip 10.1.1.6
Jul 9 10:35:37: SB1: Vl11 Hello in 10.1.1.9 Standby pri 51 ip 10.1.1.6
Jul 9 10:35:37: SB1: Vl11 Hello in 10.1.1.9 Standby pri 51 ip 10.1.1.6
Jul 9 10:35:40: SB1: Vl11 Hello out 10.1.1.8 Active pri 50 ip 10.1.1.6
Jul 9 10:35:40: SB1: Vl11 Hello out 10.1.1.8 Active pri 50 ip 10.1.1.6
Jul 9 10:35:40: SB1: Vl11 Hello out 10.1.1.8 Active pri 50 ip 10.1.1.6
Jul 9 10:35:40: SB1: Vl11 Hello out 10.1.1.8 Active pri 50 ip 10.1.1.6
Jul 9 10:35:40: SB1: Vl11 Hello out 10.1.1.8 Active pri 50 ip 10.1.1.6
Jul 9 10:35:40: SB1: Vl11 Hello out 10.1.1.8 Active pri 50 ip 10.1.1.6
Jul 9 10:35:40: SB1: Vl11 Hello out 10.1.1.8 Active pri 50 ip 10.1.1.6
Jul 9 10:35:40: SB1: Vl11 Hello out 10.1.1.8 Active pri 50 ip 10.1.1.6
Jul 9 10:35:40: SB1: Vl11 Hello out 10.1.1.8 Active pri 50 ip 10.1.1.6
Jul 9 10:35:40: SB1: Vl11 Hello out 10.1.1.8 Active pri 50 ip 10.1.1.6
Jul 9 10:35:40: SB1: Vl11 Hello out 10.1.1.8 Active pri 50 ip 10.1.1.6
Jul 9 10:35:40: SB1: Vl11 Hello out 10.1.1.8 Active pri 50 ip 10.1.1.6
Jul 9 10:35:40: SB1: Vl11 Hello out 10.1.1.8 Active pri 50 ip 10.1.1.6
Conclusion

IOS SLB can load balance RADIUS servers and provide for the high availability of RADIUS services. IOS SLB can be configured in a redundant, stateful fashion. This supports a robust data center deployment of RADIUS services, and can also be used for other services that require high availability with stateful failover and failback.
<table>
<thead>
<tr>
<th><strong>A</strong></th>
<th><strong>B</strong></th>
<th><strong>C</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AAA</strong></td>
<td><strong>BGP</strong></td>
<td><strong>CA</strong></td>
</tr>
<tr>
<td><strong>ABR</strong></td>
<td><strong>BOOTP</strong></td>
<td><strong>Cisco Content Transformation Engine (CTE)</strong></td>
</tr>
<tr>
<td>Area border router.</td>
<td>Bootstrap Protocol—Lets diskless workstations boot over the network and is described in RFC 951 and RFC 1542.</td>
<td>Cisco appliance optimized to perform the task of converting back-end data to a format appropriate for devices with special display requirements. This includes markup language translation, image translation, content compression, as well as applying user-defined rules to format the data so that it is appropriate for a given target device (for example, mobile phone, personal digital assistant (PDA), or Cisco IP Phone).</td>
</tr>
<tr>
<td><strong>ACE</strong></td>
<td><strong>Client</strong></td>
<td><strong>Client Network Address Translation (NAT)</strong></td>
</tr>
<tr>
<td>Access control entry.</td>
<td>An application program that establishes connections for the purpose of sending requests [2] on behalf of an end user.</td>
<td>Client-side source IP network addresses are converted or mapped at to an internal private address space.</td>
</tr>
<tr>
<td><strong>ACL</strong></td>
<td><strong>Client Network Address Translation (NAT)</strong></td>
<td></td>
</tr>
<tr>
<td>Access control list.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>AH</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Authentication header.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>ARP</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Address Resolution Protocol—A low-level TCP/IP protocol that maps a node’s hardware address (called a “MAC” address) to its IP address. Defined in RFC 826.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>ASBR</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Autonomous system boundary router.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
**Concurrent Connections (CC)**
The total number of single connections established through a device at a given time.

**Connections per second (CPS)**
The number of single connection requests received during one second. Loading a single Web page could generate, and usually does, multiple connection requests.

**Connection Persistence**
The ability to pipeline multiple HTTP requests to the same TCP Connection. This method uses HTTP 1.1.

**Cookie**
A message given to a Web browser by a Web server or Cisco CSS 11000 CSS. The browser stores the message in a text file called cookie.txt. The message is then sent back to the server each time the browser makes a request to the URL associated with the cookie.

The main purpose of cookies is to identify users and possibly prepare customized Web pages for them. For example, instead of seeing just a generic welcome page, the user might see a welcome page with their name on it.

**Cookie Active**
The load balancer inserts cookies in client packets. The cookies identify the server that will handle subsequent client requests.

**Cookie Passive**
The server inserts cookies in the client packets. The load balancer reads the cookies in the responsive stream and creates a table entry containing the cookie and server ID.

**Cookie Sticky**
The ability for a load balancing device to “stick” or create a persistent connection between a client and server based on cookie information located in the browser.

**Content Optimization Engine (COE)**
Solution targeted at solving some of the inherent problems with transmitting content over wireless networks. COE provides content and link optimization for Service Providers, Mobile Operators, and large Enterprises.

Content Engine based solution on the CE590 that compresses GIFs, JPGs, and text to lessen the data that will flow of the network. Also included is optimization of the TCP stack. Hooks have been added to allow a provider to add XML style sheets to the Cisco Content Engine, as well, to provide transcoding (or content adaptation) for multiple device support. Mobile Operators will require this type of solution to lessen the bandwidth requirements needed today for transmitting data over their wireless infrastructure.

**Content Rule**
A hierarchal rule set containing individual rules that describe which content (for example, .html files) is accessible by visitors to the Web site, how the content is mirrored, on which server the content resides, and how the CSS processes the request for content. Each rule set must have an owner.

**Content Switching, CS**
A device that front-ends the servers and dynamically decides which back-end server services the client request. The CS makes the decision based on a multitude if Layer 3 through Layer 7 criteria that is “sniffed from the client request. See delayed binding.

**CPU**
Central processing unit.

**CRL**
Certificate revocation list.

**CTI**
Computer telephony integration.

**CSM**
Content Switching Module

**CSS**
Content Service Switch
### Glossary

#### D

**Data Center**
Consist of either one or more physical facilities whose responsibility is to house network infrastructure devices, which provide transport of hosted services. These services are often sourced from server farms residing inside the Data Center.

**Delayed Binding**
This process involves the server load balancing device intercepting the initial TCP connection, but delaying the actual connection to the destination (real) server. Instead, the server load balancing devices send the TCP acknowledgement to the client that triggers the browser to transmit the HTTP request header, which contains the URL and the cookie.

**DES**
Data Encryption Standard.

**DH**
Diffie-Hellman.

**DHCP**
Dynamic Host Configuration Protocol.

**Directed Mode**
Directed mode rewrites the IP address of the packet when the packet is designed for the virtual server. This allows more flexibility by not requiring the real server to be on the same subnet.

**Dispatch Mode**
Dispatch mode rewrites the MAC address of each packet when the packet is destined for the virtual server. This requires that the server be Layer 2 adjacent to the load balancer (i.e., same subnet). It also requires that the network topology physically restricts the traffic so that the traffic returning from the real servers must go through the load balancer. Each real server must have the IP address of the virtual server as a secondary IP address/loopback interface.

**DNS**
Domain name system—Operates over UDP unless zone file access over TCP is required.

**DoS**
Denial of service.

**Dynamic Feedback Protocol (DFP)**
Enables load-balancing devices (DFP Managers) to leverage valuable information that reside on servers (DFP Agents) and other network appliances. Used in local or global load-balancing environments, DFP gives servers the ability to dynamically provide statistical load and availability information back to the SLB device. DFP allows servers to communicate relative weights for the availability of application systems or the server itself. Weights are dynamically reported for each real server that is supported by a virtual address as represented by the SLB device.

#### E

**ECMP**
Equal cost multi-path.

**EEPROM**
Electrically erasable programmable read-only memory.

**EGP**
Exterior Gateway Protocol—While PIX Firewall does not support use of this protocol, you can set the routers on either side of the PIX Firewall to use RIP between them and then run EGP on the rest of the network before the routers.

**EIGRP**
Enhanced Interior Gateway Routing Protocol—While PIX Firewall does not support use of this protocol, you can set the routers on either side of the PIX Firewall to use RIP between them and then run EIGRP on the rest of the network before the routers.
### Glossary

| **ESP** | Encapsulating Security Payload. Refer to RFC 1827 for more information. |
| **Extensible Markup Language** | A language that allows developers to create customizable tags to aid in the definition, transmission, validation, and interpretation of data between applications. |

---

**F**

| **Firewall Load Balancing (FWLB)** | For scalable firewall security, Cisco intelligently directs traffic across multiple firewalls, eliminating performance bottlenecks and single points of failure. Firewall load balancing eliminates system downtime that results when a firewall fails or becomes overloaded—breaking Internet connections and disrupting e-commerce purchases or other mission-critical transactions. |
| **Flash Crowd** | Unpredictable, “event-driven” traffic that swamps servers and disrupts site services. |
| **FTP** | File Transfer Protocol. |

---

**G**

| **Global Server Load Balancing (GSLB)** | Load-balancing servers across multiple sites, allowing local servers to respond to not only incoming requests, but to remote servers as well. The Cisco CSS 11000 Content Services Switch supports GSLB through inter-switch exchanges or via a proximity database option. |

---

**H**

| **H.323** | A collection of protocols that allow the transmission of voice data over TCP/IP networks. |
| **Health Checks** | Used by the server load balancing devices to check server state and availability based on standard application and network protocols and (depending on the server load balancing product) sometimes customized health check information. |
| **Hosting Solutions Engine (HSE)** | Turnkey, hardware-based solution for e-business operations in Cisco-powered data centers. Provides: Fault and performance monitoring of Cisco hosting infrastructure and Layer 4-7 hosted services, Layer 4-7 service activation, such as taking Web servers in and out of service, Historical data reporting for all monitored devices/services, and Tiered user access model and customer view personalization. |
| **HTTP Redirection** | The process by which Hypertext Transfer Protocol (HTTP) requests made by the Cisco Content Distribution Manager are redirected to a client “local” content engine. The request is then served from the content engine. |
| **HSRP** | Hot-Standby Routing Protocol. |
| **HTTPS** | HTTP over SSL. |
| **Hypertext Transfer Protocol (HTTP 1.0)** | This version of HTTP requires a separate TCP connection for each HTTP request initiated. Because of this, a high amount of overhead is associated with the use of HTTP 1.0. |
| **Hypertext Transfer Protocol (HTTP 1.1)** | This version provides persistent connection capability and allows multiple HTTP requests to be pipelined through a single TCP connection. |
IANA
Internet Assigned Number Authority—Assigns all port and protocol numbers for use on the Internet.

ICMP
Internet Control Message Protocol—This protocol is commonly used with the ping command. You can view ICMP traces through the PIX Firewall with the debug trace on command. Refer to RFC 792 for more information.

Internet Data Center (IDC)
A large scale, often shared infrastructure, which provides managed hosting services to customers.

IFP
Internet Filtering Protocol.

IGMP
Internet Group Management Protocol.

IGRP
Interior Gateway Routing Protocol.

IKE
Internet Key Exchange.

IKMP
Internet Key Management Protocol.

IOSSLB
IOS Server Load Balancing

IP
Internet Protocol.

IPSec

ISAKMP
Internet Security Association and Key Management Protocol.

ITU
International Telecommunication Union.

Lightweight Directory Access Protocol (LDAP)
Protocol that provides access for management and browser applications that provide read/write interactive access to the X.500 Directory.

LSA
link-state advertisement.

MD5
Message Digest 5—An encryption standard for encrypting VPN packets.

MIB
Management information base—The database used with SNMP.

MTU
Maximum transmission unit—The maximum number of bytes in a packet that can flow efficiently across the network with best response time. For Ethernet, the default MTU is 1500 bytes, but each network can have different values, with serial connections having the smallest values. The MTU is described in RFC 1191.
N

**Network Address Translation (NAT)**

Provides the ability to map hidden “internal” network IP addresses to routable “external” issued IP addresses. The internal IP addresses are typically drawn from the private address spaces defined in RFC 1918.

**NAT Peering**

Cisco CSS11000 Series Switches use NAT Peering to direct requests to the best site with the requested content based on URL or file type, geographic proximity and server/network loads, avoiding the limitations of Domain Name System (DNS)-based site selection and the overhead of HTTP redirect. NAT peering acts as a “triangulation protocol” allowing the response to be directly delivered to the user over the shortest Internet path.

**NBMA**

Non broadcast multiaccess.

**NetBIOS**

Network Basic Input Output System—An application programming interface (API) that provides special functions for PCs in local-area networks (LANs).

**NIC**

Network Information Center.

**NNTP**


**NOS**

Network operating system.

**NSSA**

Not so stubby area.

**NTP**

Network Time Protocol—Set system clocks via the network.

**NVT**

Network virtual terminal.

O

**OSPF**

Open Shortest Path First protocol.

**Origin Web Server**

Core of Content Networking. Base from where web services are sourced.

P

**PAT**

Port address translation.

**PFS**

perfect forward secrecy.

**PIM**

Protocol Independent Multicast.

**PIM-SM**

PIM sparse mode.
PIX

Private Internet Exchange.

PKI

Public Key Infrastructure.

RADIUS
Remote authentication dial-in user service—User authentication server specified with the `aaa-server` command.

RAS
The registration, admission, and status protocol. Provided with H.323 support.

RC4
RC4 is stream cipher designed by Rivest for RSA Data Security, Inc. It is a variable key-size stream cipher with byte-oriented operations. The algorithm is based on the use of a random permutation.

RFC
Request for comment—RFCs are the de facto standards of networking protocols.

RHI
Route health injection.

RIP
Routing Information Protocol.

RPC
Remote procedure call.

RSA
Rivest, Shamir, and Adelman. RSA is the trade name for RSA Data Security, Inc.

RTP
Real-Time Transport Protocol.

RTCP
RTP Control Protocol.

RTSP
Real Time Streaming Protocol.

SA
Security association.

SCCP
Simple (Skinny) Client Control Protocol.

Secure Content Accelerator (SCA)
The Cisco 11000 series Secure Content Accelerator (SCA 11000) is an appliance-based solution that increases the number of secure connections supported by a Web site by offloading the processor-intensive tasks related to securing traffic with SSL. Moving the SSL security processing to the SCA simplifies security management and allows Web servers to process more requests for content and handle more e-transactions.

Secure Sockets Layer (SSL)
A security protocol that provides communications privacy over the Internet. The protocol allows client/server applications to communicate in a way that is designed to prevent eavesdropping, tampering, or message forgery.

Server NAT
The IP address of the real server on the internal network is converted or mapped to a client side network IP address. Therefore, the real server’s IP address is never advertised to the client-side network.
<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Session</strong></td>
<td>Loosely defined as series of HTTP and TCP connections to a site, which constitute a single client’s visit.</td>
</tr>
<tr>
<td><strong>Session Persistence</strong></td>
<td>The ability for server load balancing device to “stick” a requester (client) to a particular server using various methods including: Source IP sticky, Cookie Sticky, SSL Sticky, HTTP Redirection Sticky.</td>
</tr>
<tr>
<td><strong>Session State</strong></td>
<td>Multiple connections that belong to the same session for which session state is kept.</td>
</tr>
<tr>
<td><strong>SLB Data Path</strong></td>
<td>Catalyst 6000 switching bus path followed by control SLB packets.</td>
</tr>
<tr>
<td><strong>SLB Engine</strong></td>
<td>The CISCO IOS®SLB Software process running at the SLB device. It controls all of Load Balancing capabilities and it may use ASIC to assist certain specific tasks.</td>
</tr>
<tr>
<td><strong>SLB Device</strong></td>
<td>Switch or router running Cisco IOS SLB. Currently Catalyst 6000, 7200 router, and Catalyst 4840.</td>
</tr>
<tr>
<td><strong>SMR</strong></td>
<td>Stub multicast routing.</td>
</tr>
<tr>
<td><strong>SMTP</strong></td>
<td>Simple Mail Transfer Protocol—Mail service. The <em>fixup protocol smtp</em> command enables the Mail Guard feature. The PIX Firewall Mail Guard feature is compliant with both the RFC 1651 EHLO and RFC 821 section 4.5.1 commands.</td>
</tr>
<tr>
<td><strong>SNMP</strong></td>
<td>Simple Network Management Protocol—Set attributes with the <em>snmp-server</em> command.</td>
</tr>
<tr>
<td><strong>SPI</strong></td>
<td>Security Parameter Index—A number which, together with a destination IP address and security protocol, uniquely identifies a particular security association.</td>
</tr>
<tr>
<td><strong>Source IP Sticky</strong></td>
<td>The ability for a load balancing device to “stick” or create a persistent connection between a client and server based on the client source IP address.</td>
</tr>
<tr>
<td><strong>Source of Authority (SOA)</strong></td>
<td>The primary DNS server for a particular domain.</td>
</tr>
<tr>
<td><strong>SSH</strong></td>
<td>Secure Shell.</td>
</tr>
<tr>
<td><strong>SSL Optimization</strong></td>
<td>Content-switching service wherein SSL sessions are terminated prior to the server in order to enable application of content rules to encrypted traffic. Utilization of this service enhances performance without sacrificing security.</td>
</tr>
<tr>
<td><strong>SSL Sticky</strong></td>
<td>The ability for a load balancing device to “stick” or create a persistent connection between a client and server based on the SSL session id of the client.</td>
</tr>
<tr>
<td><strong>Stateful Failover</strong></td>
<td>Ensures that connection “state” information is maintained upon failover from one device to another. Session transaction information is also maintained and copied between devices to alleviate any downtime from occurring with websites and services.</td>
</tr>
<tr>
<td><strong>Stateless Failover</strong></td>
<td>Maintains both device and link failure status and provides failover notifications if one of these fails. However, unlike stateful failover, stateless failover does not copy session state information from one device to another upon failure. Therefore, any “state” information between the client and server must be retransmitted.</td>
</tr>
<tr>
<td><strong>Storage Array (SA)</strong></td>
<td>Cisco storage arrays provide storage expansion to Cisco’s Content Delivery Network products. Two models are offered: Cisco Storage Array 6 (108 GB) and Cisco Storage Array 12 (216 GB).</td>
</tr>
<tr>
<td><strong>SYN</strong></td>
<td>Synchronize sequence numbers flag in the TCP header.</td>
</tr>
</tbody>
</table>
**T**

**TACACS+**
Terminal access controller access control system plus.

**Time to Live (TTL)**
The time to live a packet has to traverse the network. Each hop that a packet takes through the network, decrements the TTL value until it is eventually dropped. Keeps the packet from bouncing around the network. For multicast, the TTL should never be greater than 7, for routing the TTL should never greater than 15.

**TCP**
Transmission Control Protocol. Refer to RFC 793 for more information.

**TCP Control Packets**
TCP packets with certain flags turned on that indicate a particular action is to take place.
- **SYN:** Connection request indication
- **ACK:** Acknowledgement
- **SYN/ACK:** Connection request acknowledgement indication
- **FIN:** Connection Teardown indication
- **FIN/ACK:** Connection Teardown Acknowledgement indication
- **RST:** Connection reset indication

**TFTP**
Trivial File Transfer Protocol.

**Triple DES**
Triple Data Encryption Standard. Also known as 3DES.

**Transport Layer Security**
A protocol providing communications privacy over the Internet. The protocol allows client/server applications to communicate in a way that prevents eavesdropping, tampering, or message forgery. TLS is the successor to Secure Sockets Layer (SSL).

**U**

**uauth**
User authentication.

**UDP**
User Datagram Protocol.

**Universal Resource Locator (URL)**
Standardized addressing scheme for accessing hypertext documents and other services using a browser. URLs are contained within the User Data field and point to specific Web pages and content.

**URL Hashing**
This feature is an additional predictor for Layer 7 connections in which the real server is chosen using a hash value based on the URL. This hash value is computed on the entire URL or on a portion of it.

**V**

**Virtual Server**
Logical server in a content switch used to a service offered by multiple Real Servers to a single IP address, protocol and port number used by clients to access the specific service.

**VLAN**
virtual LAN.

**VoIP**
Voice over IP.
WWW  World wide web.
INDEX

Numerics

2.1(11b)E2  3-16
7600  2-3
802.1q  2-6
802.1q tag  2-6, 2-8
802.1x  4-2

A

accounting, RADIUS  4-5
ACS server  4-3
active-standby configuration  2-6
alias command  2-12
application health checks  2-2
application optimization  1-1
authentication sequence, RADIUS  4-3
AXFR  3-7, 3-11, 3-22

B

bridge mode  2-21, 3-5
buddied virtual servers  4-5
business continuance  1-9

C

cache  3-5
caching, application optimization  1-1
Catalyst 6500  2-3, 2-5, 2-6, 2-12, 2-13, 3-3
CatOS  2-3
CC  2-5, 2-6
Cisco Appliance Service Architecture (CASA)  4-7
Cisco MDS 9000 Family  1-8
client NAT  2-14
concurrent connections  2-5
configuration file transfer  3-2
connection “state” information  3-6
connection setup rate  3-12
connections per second  2-5
content switch  2-2, 3-3, 3-5, 3-6
content switching
  application optimization  1-1
  benefits  2-2
Content Switching Module
  see CSM
content transformation  2-10
CPS  2-3, 2-5, 2-6
CSCdx61788  3-22
CSCdx62349  3-16, 3-22
CSM  1-4
  configuration scalability  2-24
  deployment  2-7
  deployment modes  2-11
  hardware requirements  2-3
  high availability  2-6, 2-23
  interoperability  2-5
  modes  2-7
  modes of operation  2-7
  performance  2-6, 3-11
  performance summary  2-25
  scalability  2-5
  throughput  2-25
D

data center security 1-8
debug, RADIUS 4-9
demilitarized zone 3-2
design details, network infrastructure 2-4
dig 3-21
direct-attached storage 1-8
disaster recovery 1-9
DMZ 3-2, 3-7
DNS
  probes 3-6
  proxies 3-1, 3-10
  queries 3-1
  query type A 3-9
  query type NS 3-10
  record configuration 3-9
  requests 3-20
  responses 3-1
  server design 3-7
  traffic 3-1
  zones 3-2

H

high availability 2-5, 2-6, 3-4
Hot Standby Router Protocol
  see HSRP
HSRP 2-12, 2-13, 2-18, 2-22
HSRP groups 2-5

I

ICMP probes 3-15
idle connection 3-5
idle timeout 3-5, 3-12
infrastructure 1-8
inservice standby SERVER_VLAN 3-17
interoperability 2-5
IOS-SLB 2-7, 3-3, 3-4, 3-12
  see also SLB
ip default next-hop 3-5

K

keepalives 2-5

L

Layer 2
  EtherChannel 2-23
  leveraging 2-4
  links 2-10
  lookup 2-11
  MAC address 2-12
  trunk 2-11
Layer 3 2-4
Layer 4 2-1
Layer 4-7 2-3, 2-4, 2-10
Layer 5 2-1, 2-3
  level of authority 3-2
**load balancing algorithms**  2-2  
**low convergence times**  2-4

---

**M**

**master to slave communication**  3-2  
**maximum number of concurrent connections**  3-11  
**maximum teardown rate**  3-12  
**metrics**  2-5  
**MHSRP**  2-19  
**MSFC**  2-3, 2-6, 2-9, 2-11, 2-12, 2-13  
**multi-group HSRP**  2-19

---

**N**

**name servers**  3-2  
**NAT**

  - **client**  2-14, 2-20, 2-22  
  - **per-packet**  3-15  
  - **server**  2-14, 2-20, 2-23  
  - **servers**  3-15  
  - **source**  3-14  
  - **static**  3-6

**native IOS**  2-3  
**network address translation**

  - see **NAT**

**network-attached storage (NAS)**  1-8  
**network infrastructure**  1-8  
**non-recursive queries**  3-9  
**non-responsive server**  2-2  
**nslookup**  3-21  
**N-tier migration**  1-3, 1-4

---

**O**

**one-arm CSM mode**  2-5, 2-13

---

**P**

**packet flow**

  - **RADIUS**  4-2  
  - **packets per second**  2-5  
  - **PAT**  3-6, 3-14  
  - **performance**  2-3  
  - **per-packet**  3-6, 3-15  
  - **ping probes**  3-14  
  - **policies**  2-5  
  - **port address translation**

    - see **PAT**

**PPS**  2-5, 2-6  
**private VLANs**  2-13  
**public IP addresses**  3-6

---

**Q**

**query type**

  - A  3-1

---

**R**

**RADIUS**

  - **accounting**  4-5  
  - **attributes**  4-3  
  - **debug**  4-9  
  - **load balancing**  4-1 to 4-12  
  - **packet flow**  4-2  
  - **protocol**  4-1  
  - **UDP ports**  4-1  
  - **real servers**  2-5  
  - **recovery point objectives (RPOs)**  1-9  
  - **recovery time objectives (RTOs)**  1-9  
  - **recursive queries**  3-9, 3-10  
  - **redundancy**  2-12  
  - **resiliency**  2-1  
  - **reverse proxy caching**  1-2  
  - **RFC**

    - 191  2-14

---
SAN extension 1-9
scalability 2-1, 2-5
scaling DNS services 1-5, 3-1
  benefits 3-4
caveats 3-22
CSM configuration 3-13
design description 3-6
hardware requirements 3-5
IOS-SLB configuration 3-14
redundancy 3-17
sample configurations 3-17
server farms 3-15
software requirements 3-5
traffic paths 3-9
scaling RADIUS services 1-6, 4-1 to 4-12
  see also RADIUS
scaling server farms
  fundamental challenges 2-1
secure router mode 2-5, 2-12, 2-21, 2-22
  implementation 2-21
Secure Sockets Layer (SSL) offloading 1-1
security 1-8
server farms
  with SLB 4-4
server-initiated connections 2-19, 3-15
Server Load Balancing
  see SLB
server load distribution 2-2
server NAT 2-14
server-originated queries 3-10
server-to-server communication 2-13
service switches 2-9, 2-10
session persistence 2-6, 3-11
session-state servers 4-3
setup rate 3-11
show module csm connections 3-11
show module csm vserver 3-11
single subnet bridge mode 2-5, 2-12, 2-17, 2-18
  implementation 2-15
site selection 1-9
SLAs 1-3
SLB
  benefits 4-4
  extensions 4-3
    RADIUS 4-1 to 4-12
SOA 3-6, 3-7, 3-10
SOA queries 3-10, 3-11, 3-14
source of authority
  see SOA
stateful backup 4-6
stateful failover 2-6
stateless vserver 3-15
static NAT 2-23, 3-6
stickiness 3-11
sticky groups 3-13, 3-14
sticky load balancing 4-6
storage-area networks (SAN) 1-8
storage networking 1-7
supervisor card 2-3
synchronization processes 4-7

TCP 3-2
TCP probe 3-13
teamed virtual servers 4-5
teardown rate 3-11
traffic flow statistics 2-5
U

UDP  3-2, 3-6, 3-10
   ports for RADIUS services  4-1

V

VIP  2-11, 2-14, 2-17, 3-11
   virtual primary name server  3-7
   virtual secondary name server  3-7, 3-8
   virtual servers
      see vservers
VLAN  2-6, 4-2
   vservers
      configuration  3-13
      IP addresses  2-14

Z

zone file revision numbers  3-2
zone files  3-7, 3-8
zone transfers  3-2, 3-7, 3-10, 3-13, 3-14, 3-15, 3-16, 3-22