Data Center—Site Selection for Business Continuance

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Preface

For small, medium, and large businesses, it is critical to provide high availability of data for both customers and employees. The objective behind disaster recovery and business continuance plans is accessibility to data anywhere and at any time. Meeting these objectives is all but impossible with a single data center. The single data center is a single point of failure if a catastrophic event occurs. The business comes to a standstill until the data center is rebuilt and the applications and data are restored.

As mission-critical applications have been Web-enabled, the IT professional must understand how the application will withstand an array of disruptions ranging from catastrophic natural disasters, to acts of terrorism, to technical glitches. To effectively react to a business continuance situation, all business organizations must have a comprehensive disaster recovery plan involving several elements, including:

- Compliance with federal regulations
- Human health and safety
- Reoccupation of an effected site
- Recovery of vital records

- Recovery information systems (including LAN/WAN recovery), electronics, and telecommunications recovery

Enterprises can realize application scalability and high availability and increased redundancy by deploying multiple data centers, also known as distributed data centers (DDC). This Solutions Reference Network Design (SRND) guide discusses the benefits, technologies, and platforms related to designing distributed data centers. More importantly, this SRND discusses disaster recovery and business continuance, which are two key problems addressed by deploying a DDC.

Intended Audience

This document is for intended for network design architects and support engineers who are responsible for planning, designing, implementing, and operating networks.

Chapter 1—Site Selection Overview

This chapter describes how application recovery, disaster recovery, and business continuance are achieved through site selection, site-to-site recovery, and load balancing. It includes the following sections:

The Need for Site Selection

Centralized data centers have helped many Enterprises achieve substantial productivity gains and cost savings. These data centers house mission-critical applications, which must be highly available. The demand on data centers is therefore higher than ever before. Data center design must focus on scaling methodology and achieving high availability. A disaster in a single data center that houses Enterprise applications and data has a crippling affect on the ability of an Enterprise to conduct business. Enterprises must be able to survive any natural or man-made disaster that may affect the data center.

Enterprises can achieve application scalability, high availability, and redundancy by deploying distributed data centers. This document discusses the benefits, technologies, and platforms related to designing distributed data centers, disaster recovery, and business continuance.

For small, medium and large businesses, it is critical to provide high availability of data for both customers and employees. 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, the business comes to a standstill until the single data center is rebuilt and the applications and data are restored.

Business Goals and Requirements

Before going into the details, it is important to keep in mind why organizations use data centers and require business continuance strategies. Technology allows businesses to be productive and to quickly react to business environment changes. Data centers are one of the most important business assets and data is the key element. Data must be protected, preserved, and highly available.

For a business to access data from anywhere and at any time, the data center must be operational around the clock, under any circumstances. In addition to high availability, as the business grows, businesses should be able to scale the data center, while protecting existing capital investments. In summary, data is an important aspect of business and from this perspective; the business goal is to achieve redundancy, high availability, and scalability. Securing the data must be the highest priority.

The Problem

In today's electronic economy, any application downtime quickly threatens a business's livelihood. Enterprises lose thousands of dollars in productivity and revenue for every minute of IT downtime. A recent study by Price Waterhouse Coopers revealed that globally network downtime costs business $1.6 Trillion in the last year. This equated to 4.4 Billion per day, $182 million per hour, or $51,000 per second. In the U.S. with companies with more than 1000 employees, it is a loss of $266 Billion in the last year. A similar Forrester Research survey of 250 Fortune 1000 companies revealed that these businesses lose a staggering US$13,000 for each minute that an Enterprise resource planning (ERP) application is inaccessible. The cost of supply-chain management application downtime runs a close second at US$11,000 per minute, followed by e-commerce (US$10,000).

To avoid costly disruptions, Enterprises are turning to intelligent networking capabilities to distribute and load balance their corporate data centers—where many of their core business applications reside. The intelligence now available in IP networking devices can determine many variables about the content of an IP packet. Based on this information, the network can direct traffic to the best available and least loaded sites and servers that will provide the fastest-and best-response.

Business continuance and disaster recovery are important goals for businesses. According to the Yankee Group, business continuity is a strategy that outlines plans and procedures to keep business operations, such as sales, manufacturing and inventory applications, 100% available.

Companies embracing e-business applications must adopt strategies that keep application services up and running 24 x 7 and ensure that business critical information is secure and protected from corruption or loss. In addition to high availability, the ability to scale as the business grows is also important.

The Solution

Resilient networks provide business resilience. A business continuance strategy for application data that provides this resilience involves two steps.

- Replicating data, either synchronously or asynchronously
- Directing users to the recovered data

Data needs to be replicated synchronously or at regular intervals (asynchronously). It must then be retrieved and restored when needed. The intervals at which data is backed up is the critical component of a business continuance strategy. The requirements of the business and its applications dictate the interval at which the data is replicated. In the event of a failure, the backed up data must be restored, and applications must be enabled with the restored data.

The second part of the solution is to provide access and direct users to the recovered data. The main goal of business continuance is to minimize business losses by reducing the time between the loss of data and its full recovery and availability for use. For example, if data from a sales order is lost, it represents a loss for the business unless the information is recovered and processed in time to satisfy the customer.

Single Site Architecture

When you consider business continuance requirements, it is clear that building a single data center can be very risky. Although good design protects access to critical information if hardware or software breaks down at the data center, that doesn't help if the entire data center becomes inaccessible. To deal with the catastrophic failure of an entire site, applications and information must be replicated at a different location, which requires building more than one data center.

Multi-Site Architecture

When application data is duplicated at multiple data centers, clients go to the available data center in the event of catastrophic failure at one site. Data centers can also be used concurrently to improve performance and scalability. Building multiple data centers is analogous to building a global server farm, which increases the number of requests and number of clients that can be handled.

Application information, often referred to as content, includes critical application information, static data (such as web pages), and dynamically generated data.

After content is distributed to multiple data centers, you need to manage the requests for the distributed content. You need to manage the load by routing user requests for content to the appropriate data center. The selection of the appropriate data center can be based on server availability, content availability, network distance from the client to the data center, and other parameters.

Application Overview

The following sections provide an overview of the applications at the heart of the data center, which can be broadly classified into two categories:

- Legacy Applications
- Non-Legacy Applications

Legacy Applications

Legacy applications are based on programming languages, hardware platforms, operating systems, and other technology that were once state-of-the art, but are now outmoded. Many large Enterprises have legacy applications and databases that serve critical business needs. Organizations are often challenged to keep legacy application running during the conversion to more efficient code that makes use of newer technology and software programming techniques. Integrating legacy applications with more modern applications and subsystems is also a common challenge.

In the past, applications were tailored for a specific operating system or hardware platform. It is common today for organizations to migrate legacy applications to newer platforms and systems that follow open, standard programming interfaces. This makes it easier to upgrade software applications in the future without having to completely rewrite them. During this process of migration, organizations also have a good opportunity to consolidate and redesign their server infrastructure.

In addition to moving to newer applications, operating systems, platforms, and languages, Enterprises are redistributing their applications and data to different locations. In general, legacy applications must continue to run on the platforms for which they were developed. Typically, new development environments provide ways to support legacy applications and data. With many tools, newer programs can continue to access legacy databases.

In an IP environment, the legacy applications typically have hard-coded IP addresses for communicating with servers without relying on DNS.

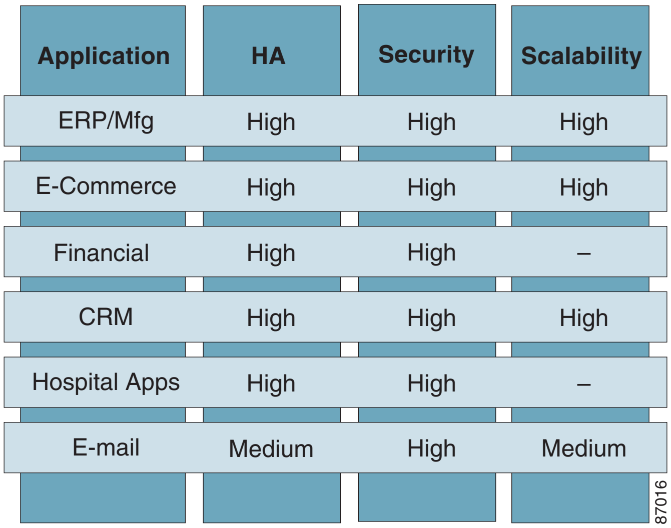
Non-Legacy Applications

The current trend is to provide user-friendly front-ends to applications, especially through the proliferation of HTTP clients running web-based applications. Newer applications tend to follow open standards so that it becomes possible to interoperate with other applications and data from other vendors. Migrating or upgrading applications becomes easier due to the deployment of standards-based applications. It is also common for Enterprises to build three-tier server farm architectures that support these modern applications. In addition to using DNS for domain name resolution, newer applications often use HTTP and other Internet protocols and depend on various methods of distribution and redirection.

Application Requirements

Applications store, retrieve and modify data based on client input. Typically, application requirements mirror business requirements for high availability, security, and scalability. Applications must be capable of supporting a large number of users and be able to provide redundancy within the data center to protect against hardware and software failures. Deploying applications at multiple data centers can help scale the number of users. As mentioned earlier, distributed data centers also eliminate a single point of failure and allow applications to provide high availability. Figure 1 provides an idea of application requirements.

Figure 1 Application Requirements



Application	HA	Security	Scalability
ERP/Mfg	High	High	High
E-Commerce	High	High	High
Financial	High	High	–
CRM	High	High	High
Hospital Apps	High	High	–
E-mail	Medium	High	Medium

Most modern applications have high requirements for availability, security, and scalability.

Benefits of Distributed Data Centers

The goal of deploying multiple data centers is to provide redundancy, scalability and high availability. Redundancy is the first line of defense against any failure. Redundancy within a data center protects against link failure, equipment failure and application failure and protects businesses from both direct and indirect losses. A business continuance strategy for application data backup that addresses these issues includes data backup, restoration, and disaster recovery. Data backup and restoration are critical components of a business continuance strategy, which include the following:

- Archiving data for protection against data loss and corruption, or to meet regulatory requirements
- Performing remote replication of data for distribution of content, application testing, disaster protection, and data center migration
- Providing non-intrusive replication technologies that do not impact production systems and still meet shrinking backup window requirements
- Protecting critical e-business applications that require a robust disaster recovery infrastructure. Providing real-time disaster recovery solutions, such as synchronous mirroring, allow companies to safeguard their data operations by:
  - Ensuring uninterrupted mission-critical services to employees, customers, and partners
  - Guaranteeing that mission-critical data is securely and remotely mirrored to avoid any data loss in the event of a disaster

Another benefit of deploying distributed data centers is in the Wide-Area Bandwidth savings. As companies extend applications throughout their global or dispersed organization, they can be hindered by limited Wide Area Network (WAN) bandwidth. For instance, an international bank has 500 remote offices world-wide that are supported by six distributed data centers. This bank wants to roll-out sophisticated, content-rich applications to all their offices without upgrading the entire WAN infrastructure. An intelligent site selection solution that can point the client to a local data center for content requests instead of one located remotely will save costly bandwidth and upgrade expenses.

The following sections describe how these aspects of a business continuance strategy are supported through deploying distributed data centers.

Site-to-Site Recovery

Deploying more than one data center provides redundancy through site-to-site recovery mechanisms. Site-to-site recovery is the ability to recover from a site failure by ensuring failover to a secondary or backup site. As companies realize the productivity gains the network brings to their businesses, more and more companies are moving towards a distributed data center infrastructure, which achieves application redundancy and the other goals of a business continuance strategy.

Multi-Site Load Distribution

Distributing applications among multiple sites provides a more efficient, cost-effective use of global resources, ensures scalable content, and gives end users better response time. Routing clients to a site based on load conditions and the health of the site results in scalability for high demand and ensures high availability.

You can load balance many of the applications that use standard HTTP, TCP or UDP, including mail, news, chat, and lightweight directory access protocol (LDAP). Multi-site load distribution provides enhanced scalability for a variety of mission-critical e-Business applications. However, these benefits come with some hurdles. Some of the challenges include mirroring database state information and mirroring data and session information across multiple data centers. Many application vendors are wrestling with these issues. Providing the underlying infrastructure required to facilitate mirroring helps simplify the problem by providing high bandwidth and a high-speed connection between the data centers.

As mentioned earlier, you can improve data center availability and balance the load between sites by routing end users to the appropriate data centers. You can use different criteria to route end users to different data centers. In most cases, routing users to a data center that is geographically closer improves the response time. This is referred to as proximity-based site selection. In addition to this, you can route users to different data centers based on the load at the data center and on the availability of a specific application.

You can distribute applications like video on demand (VoD) or media on demand (MoD) across different data centers. Load distribution based on proximity plays an important role when delivering multimedia to end-users. In this instance, clients are redirected to the closest data center. This improves the end users experience and helps reduce congestion on the network.

Access to applications is limited by a number of factors related to hardware, software, and the network architecture. To accommodate anticipated demand, you should estimate peak traffic loads on the system to determine the number of nodes required.

Distributed data centers let you deploy the same application across multiple sites, increasing scalability and providing redundancy—both of which are key goals when supporting mission-critical applications.

Solution Topologies

This section describes some general topologies for using distributed data centers to implement a business continuance strategy. It includes the following topics:

- Site-to-site recovery
- User-to-application recovery
- Database-to-database recovery
- Storage-to-storage recovery

Site-to-Site Recovery

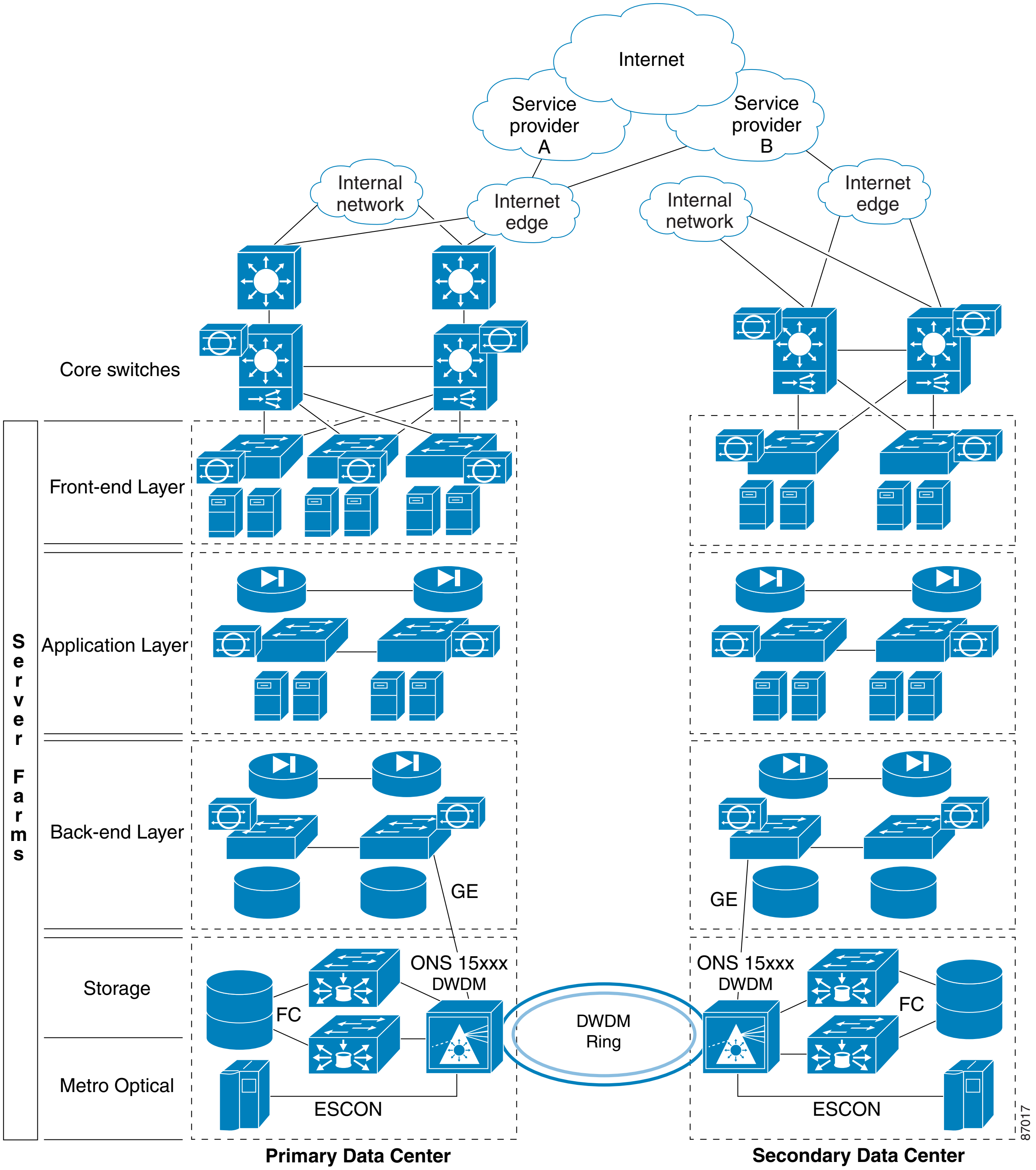
Typically, in a data center, the web servers, application servers, databases, and storage devices are organized in a multi-tier architecture, referred to as an instance of the multi-tier architecture or N-Tier architecture. This document describes the most common N-Tier model, which is the three-tier model. A three-tier architecture has the following components:

- Front-end layer
- Application layer
- Back-end layer

The front-end layer or presentation tier provides the client interface and serves information in response to client requests. The servers in this tier assemble the information and present it to the client. This layer includes DNS, FTP, SMTP and other servers with a generic purpose. The application tier, also known as middleware or business logic, contains the applications that process the requests for information and provide the logic that generates or fulfills dynamic content. This tier runs the processes needed to assemble the dynamic content and plays the key role of interconnecting the front-end and back-end tiers. Various types of databases form the back end tier.

Typically, a disaster recovery or a business continuance solution involves two data centers, as depicted in Figure 2.

Figure 2 ***Distributed Data Center Model***



There are two main topologies from a solutions perspective:

- Hot standby
- Warm standby

In a hot standby solution, the secondary data center has some applications running actively and has some traffic processing responsibilities. Resources are not kept idle in the secondary data center, and this improves overall application scalability and equipment utilization.

In a warm standby solution, the applications at the secondary data center are active at all times but the traffic is only processed by the secondary data center when the primary data center goes out of service. Note that in Figure 2, the multi-tier architecture is replicated at both the primary and secondary data centers.

User to Application Recovery

When a catastrophic failure occurs at a data center and connectivity with the application is lost, the client application might try to reconnect to the cached IP address of the server. Ultimately, you have to restart the application on the desktop because the primary data center is not available. When the client application connects to the remote server, it resolves the domain to an IP address. In a recovery scenario, the new IP address belongs to the secondary data center. The application is unaware that the secondary data center is active and that the request has been rerouted. The site selection devices monitor the applications at both data centers and route requests to the appropriate IP address based on application availability.

The alternative is to use the same IP address in both data centers and if the application in one data center becomes available the user is routed to the application in the standby data center. If the applications are stateful, the user can still connect to the application in the standby data center. However, a new connection to the standby data center is used because application state information is not exchanged between the data centers.

Database-to-Database Recovery

Databases maintain keep-alive traffic and session state information between the primary and secondary data centers. Like the application tier, the database tier has to update the state information to the secondary data center. Database state information updates tend to be chattier than application state information updates. Database updates consume more bandwidth and have a drastic impact on the corporate network if they happen frequently during regular business hours. Database synchronization benefits from the backend network infrastructure introduced to support the application tier. During a catastrophic failure at the primary data center, the secondary data center becomes active and the database rolls back to the previous update.

Storage-to-Storage Recovery

The destination for all application transactions is the storage media, like disk arrays, which are part of the data center. These disks are backed up locally using tapes and can be backed up either synchronously or asynchronously to the remote data center. If the data is backed up using disk arrays, after a catastrophic failure, the data is recovered from the tapes at an alternate data center, which requires a great deal of time and effort.

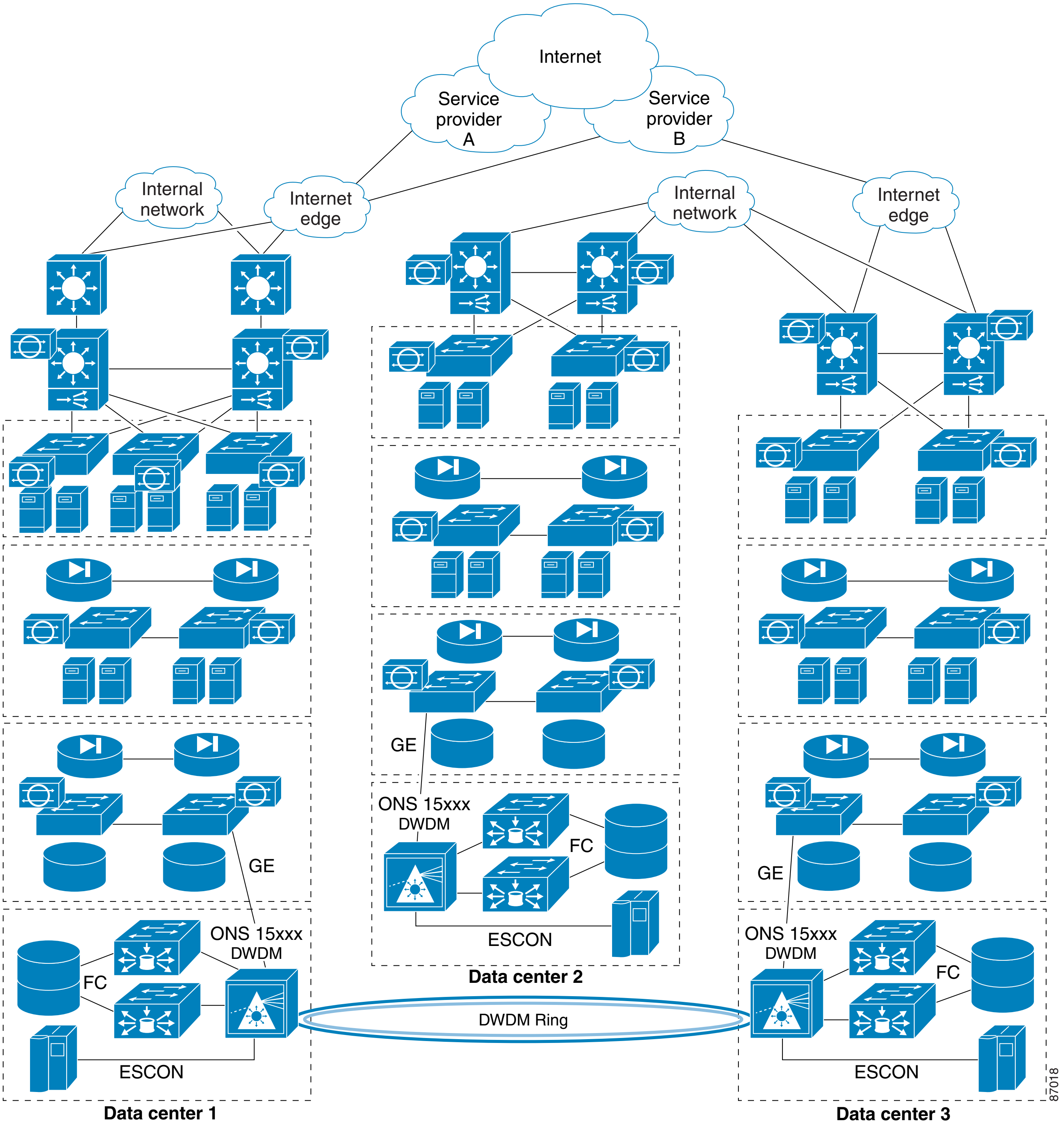
In asynchronous backup, data is written to the secondary data center at regular intervals. All the data saved on the local disk arrays for a specific window of operation is transferred to the secondary data center. When a disaster occurs, the data is retrieved from the previous update and operation resumes, starting at the last update. With this mechanism, data is rolled back to the previous update. This method has less recovery overhead when compared to tape backup mechanism and recovery is quick. Although some data loss is still likely, nearly all of the essential data is recovered immediately after a catastrophic failure.

Organizations with a low tolerance for downtime and lost data use synchronous data backup. With synchronous backup, data is written to the remote or secondary data center every time the data is written at the primary data center. If there is a catastrophic failure, the secondary data center takes over with almost no loss of data. The end user, after completing the user to application recovery process can access the secondary data center with almost no loss of data. Close to 100% of all data is recovered and there is virtually no business impact.

Multi-Site Topology

It is difficult to provide a specific multi-site topology. Multi-site topology might mean multiple sites connected together using different network technologies. The number of sites and the location of these sites depends on the business. Various factors like the number of users, the user location, and business continuance plans, dictate where the sites are located and how they are interconnected. Figure 3 provides one example of a multi-site topology.

Figure 3 Multi-Site Architecture



In a local server load-balancing environment, scalability is achieved by deploying a server farm and front-ending that server farm with a content switch. Multiple data centers can be thought of as islands of server farms with site selection technology front-ending these servers and directing end users to different data centers. The applications are distributed across different data centers. The clients requesting connection to these applications get directed to different data centers based on various criteria. This is referred to as a site selection method. Different site selection methods include least loaded, round robin, preferred sites and source IP hash.

Conclusion

Data is such a valuable corporate asset in the information age that accessibility to this data around the clock is essential to allow organizations to compete effectively. Building redundancy into the application environment helps keep information available around the clock. Because the time spent recovering from disaster has a significant impact on operations; business continuance has become an extremely critical network design goal. Statistical evidence shows a direct relationship between a successful business continuance plan and the general health of a business in the face of disaster. The Return on Investment (ROI) is justified by the costs of the direct and indirect losses incurred by a critical application outage. For these and the other compelling reasons described in this paper, all large Enterprises must seriously consider implementing business continuance strategies that include distributed data centers.

Chapter 2 —Site Selection Technologies

Several technologies make up a complete site-to-site recovery and multi-site load distribution solution. In a client to server communication, the client looks for the IP address of the server before communicating with the server. When the server is found, the client communicates with the server and completes a transaction. This transaction data is stored in the data center. The technology that deals with routing the client to the appropriate server is at the front end of data centers. In a distributed data center environment, the end users have to be routed to the data center where the applications are active. The technology that is at the front end of distributed data centers is called Request Routing.

Site Selection

Most applications use some form of address resolution to get the IP address of the servers with which they communicate. Some examples of the applications that use address resolution mechanisms to communicate with the servers or hosts are Web browsers, telnet, and thin clients on users desktop. Once an IP address is obtained, these applications connect to the servers in a secure or non-secure way, based on the application requirements, to carry out the transaction.

Address resolution can be further extended to include server health tracking. Tracking sever health allows the address resolution mechanism to select the best server to handle client requests and adds high availability to the solution. In a distributed data center environment, where redundant servers which serve the same purpose at geographically distant data centers are deployed, the clients can be directed to the appropriate data center during the address resolution process. This method of directing the clients to the appropriate server by keeping track of server health is called Request Routing.

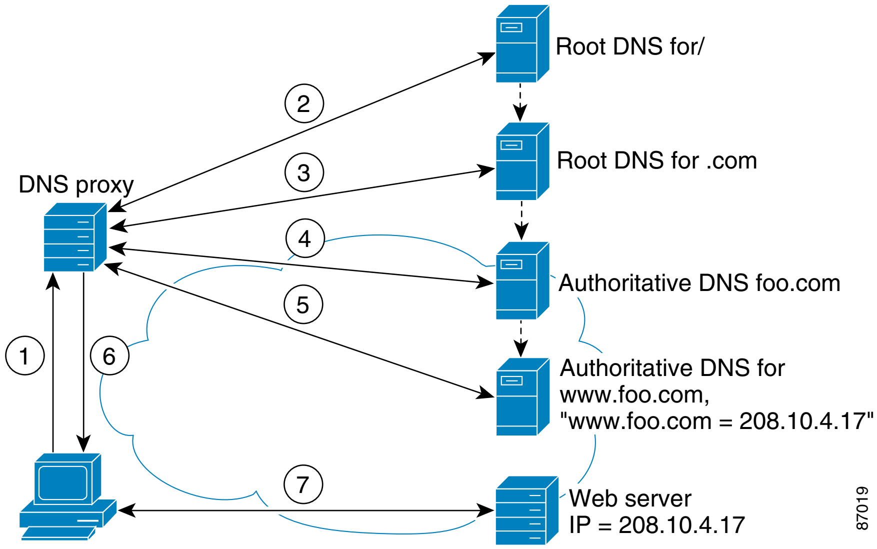
There are three methods of site selection mechanisms to connect clients to the appropriate data center:

- DNS-based request routing
- HTTP redirection
- Route Health Injection (RHI) with BGP/IGP

DNS-Based Site Selection

The first solution, depicted in Figure 4, is based on DNS. Normally, the first step when connecting to a server is resolving the domain name to an IP address. The client's resolution process becomes a DNS query to the local DNS server, which then actively iterates over the DNS server hierarchy on the Internet/Intranet until it reaches the target DNS server. The target DNS server finally issues the IP address.

Figure 4 Basic DNS Operation



1. The client requests to resolve www.foo.com.
2. The DNS proxy sends a request to the root DNS. The root DNS responds with an address of the root DNS for foo.com.
3. The DNS proxy requests the root DNS for foo.com. The response comes back with the IP address of the authoritative DNS server for foo.com.
4. The DNS proxy requests the authoritative DNS server for foo.com. The response comes back with an IP address for www.foo.com.
5. The DNS proxy requests the authoritative DNS server for www.foo.com. The response comes back with an IP address of the web server.
6. The DNS proxy responds to the client with the IP address of the web server.
7. The client establishes a connection with the web server.

At its most basic level, the DNS provides a distributed database of name-to-address mappings spread across a hierarchy of domains and sub domains with each domain administered independently by an authoritative name server. Name servers store the mapping of names to addresses in resource records. Each record keeps an associated time to live (TTL) field that determines how long the entry is cached by other name servers.

Name servers implement iterative or recursive queries:

- Iterative queries return either an answer to the query from its local database (A-record), or a referral to another name server that is able to answer the query (NS-record).
- Recursive queries return a final answer (A-record), querying all other name servers necessary to resolve the name.

Most name servers within the hierarchy send and accept only iterative queries. Local name servers, however, typically accept recursive queries from clients. Recursive queries place most of the burden of resolution on a single name server.

In recursion, a client resolver sends a recursive query to a name server for information about a particular domain name. The queried name server is then obliged to respond with the requested data, or with an error indicating that the data of the requested type or the domain name does not exist. Because the query was recursive, the name server cannot refer the querier to a different name server. If the queried name server is not authoritative for the data requested, it must query other name servers for the answer. It could send recursive queries to those name servers, thereby obliging them to find the answer and return it (and passing the buck). Alternately, the DNS proxy could send iterative queries and be referred to other name servers for the name it is trying to locate. Current implementations tend to be polite and do the latter, following the referrals until an answer is found.

Iterative resolution, on the other hand, does not require nearly as much on the part of the queried name server. In iterative resolution, a name server simply gives the best answer it already knows back to the querier. There is no additional querying required.

The queried name server consults its local data, including its cache, looking for the requested data. If it does not find the data, it makes the best attempt to give the querier data that helps it continue the resolution process. Usually these are names and addresses of other name servers.

In iterative resolution, a client’s resolver queries a local name server, which then queries a number of other name servers in pursuit of an answer for the resolver. Each name server it queries refers it to another name server further down the DNS name space and closer to the data sought. Finally, the local name server queries the name server authoritative for the data requested, which returns an answer.

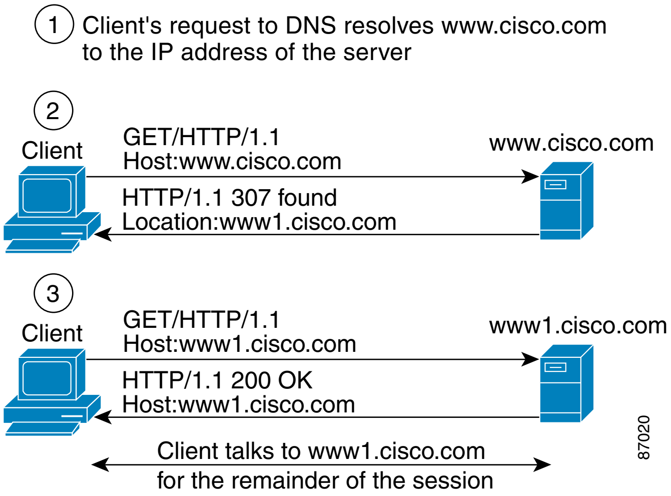
HTTP Redirection

Many applications currently available today have a browser front end. The browsers have built in http redirection built so that they can communicate with the secondary server if the primary servers are out of service. In HTTP redirection, the client goes through the address resolution process once. In the event that the primary server is not accessible, the client gets redirected to a secondary server with out having to repeat the address resolution process.

Typically, HTTP redirection works like this. HTTP has a mechanism for redirecting a user to a new location. This is referred to as HTTP-Redirection or HTTP-307 (the HTTP return code for redirection). The client, after resolving the IP address of the server, establishes a TCP session with the server. The server parses the first HTTP get request. The server now has visibility of the actual content being requested and the client’s IP address. If redirection is required, the server issues an HTTP Redirect (307) to the client and sends the client to the site that has the exact content requested. The client then establishes a TCP session with the new host and requests the actual content.

The HTTP redirection mechanism is depicted in Figure 5.

Figure 5 Basic Operation of HTTP Redirect



The advantages of HTTP redirection are:

- Visibility into the content being requested.
- Visibility of the client's IP address helps in choosing the best site for the client in multi-site load distribution.

The disadvantages of HTTP redirection are:

- In order for redirection to work, the client has to always go to the main site first and then get redirected to an alternate site.
- Book marking issues arise because you can bookmark your browser to a particular site and not the global http://www.foo.com site, thus bypassing the request routing system.
- HTTP redirects only work for HTTP traffic. Some applications, which do not have browser front ends, do not support HTTP redirection.

Route Health Injection

Route Health Injection (RHI) is a mechanism that allows the same IP address to be used at two different data centers. This means that the same IP address (host route) can be advertised with different metrics. The upstream routers see both routes and insert the route with the better metric into its routing table. When RHI is enabled on the device, it injects a static route in the device's routing table when VIPs become available. This static route is withdrawn when the VIP is no longer active. In case of a failure of the device, the alternate route is used by the upstream routers to reach the servers thereby providing high availability. It is important to note that the host routes are advertised by the device only if the server is healthy.

Note Most routers do not propagate host-route information to the Internet. Therefore, RHI, since it advertises host routes, is normally restricted to intranets.

The same IP address can also be advertised from a different location, calling it the secondary location, but with a different metric. The mechanism is exactly the same as in the previous case, with the only difference being the route is advertised with a different metric.

For applications that serve Internet users, you can summarize the host routes at the Internet edge and redistribute them into BGP. You can advertise these routes from the secondary site by using the conditional advertisement feature of Cisco BGP,. This works as long as the IP address is active at the primary site or as long as the links to the multiple service providers are active and do not advertise the IP address from the secondary site.

The advantages of RHI are:

- Quick convergence (IGP convergence)
- Self regulated, no dependency on external content routing devices
- Ideal for business continuance and disaster recovery solutions
- Single IP address

The disadvantages of RHI are:

- Cannot be used for site-to-site load balancing because the routing table has only one entry. Typically it is used only for active/standby configurations.

Supporting Platforms

Cisco has various products that support request routing for distributed data centers. Each product has different capabilities. All the supporting products are described below.

- ACE Global Site Selector (GSS 4492R)
- Application Control Engine (ACE) Module for Cat6K platforms

Global Site Selector

The Cisco GSS 4492R load balances distributed data centers. GSS interoperates with server load balancing products like the Cisco CSS 11000 and CSS 11500 Content Services Switch and the Application Control Engine (ACE) for the Cisco Catalyst® 6500 Series switches.

The Cisco GSS 4492R product delivers the following key capabilities:

- Provides a scalable, dedicated hardware platform for Cisco's content switches to ensure applications are always available, by detecting site outages or site congestion
- Improves global data center or site selection process by using different site selection algorithms
- Complements existing DNS infrastructure by providing centralized sub-domain management

The Cisco GSS 4492R allows businesses to deploy internet and intranet applications by directing clients to a standby data center if a primary data-center outage occurs. The Cisco GSS 4492R continuously monitors the load and health of the server load balancing devices at multiple data centers and can redirect clients to a data center with least load. The load conditions are user defined at each data center.

The following are key features and benefits of GSS:

- Offers site persistence for e-commerce applications
- Provides architecture critical for disaster recovery and multi-site deployments
- Provides centralized command and control of DNS resolution process
- Provides dedicated processing of DNS requests for greater performance and scalability
- Offers DNS race feature. The Cisco GSS 4492R can direct clients in real time to the closest data center based on round trip time (RTT) between the local DNS and the multiple sites.

- Supports a web-based graphical user interface (GUI) and wizard to simplify the configuration

Figure 6 Basic Operation of GSS

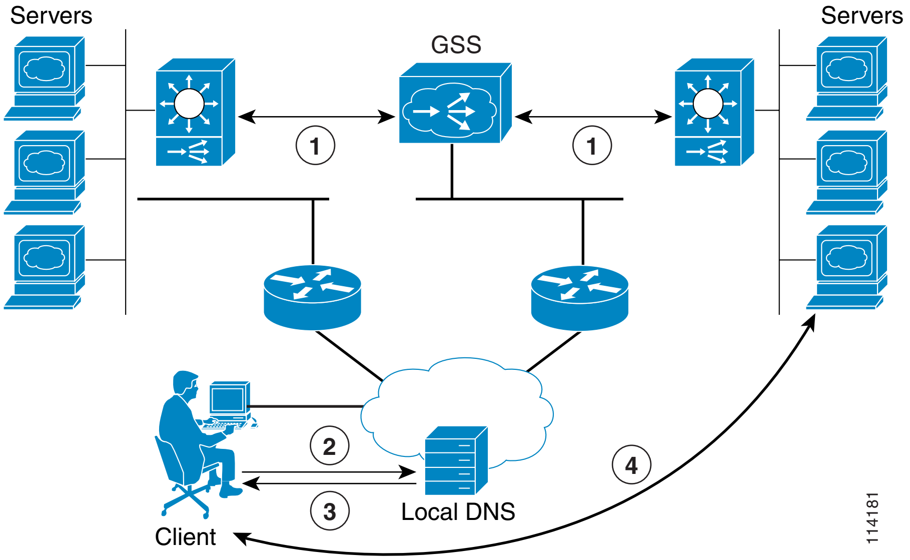


Figure 6 illustrates the basic operation of GSS, as summarized below:

1. The GSS probes for the server health and is aware of the server health and load.
2. The client requests to resolve the URL in the HTTP request.
3. The local DNS server performs the DNS query. The GSS responds with the IP address based on the configured algorithm.
4. The client connects to the server.

WebNS and Global Server Load Balancing

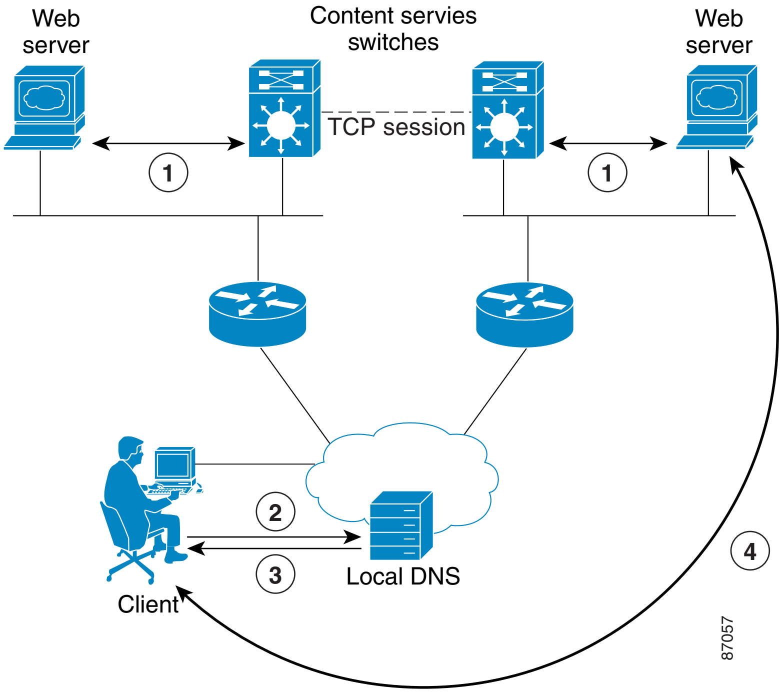
The Cisco 11000 series Content Services Switch (CSS) provide both global server load balancing (GSLB) and network proximity methods for content request distribution across multiple sites.

The Cisco 11000 series CSS is capable of GSLB of content requests across multiple sites, using content intelligence to distribute the requests according to what is being requested, and where the content is available. Network proximity is an enhanced version of GSLB that selects the closest or most proximate web site based on measurements of round-trip time to the content consumer's location. Network proximity naturally provides a high degree of global persistence, because the proximity calculation is typically identical for all requests from a given location (Local DNS) as long as the network topology remains constant.

WebNS also provides a scalable solution that provides sticky site selection without sacrificing proximity or GSLB. In this enhanced version, the sticky database allows the network administrator to configure how long a D-proxy remains sticky. The TTL value ranges from minutes to days.

Figure 7 explains the basic operation of GSLB using content services switch.

Figure 7 Basic Operation of GSLB Using Content Services Switch



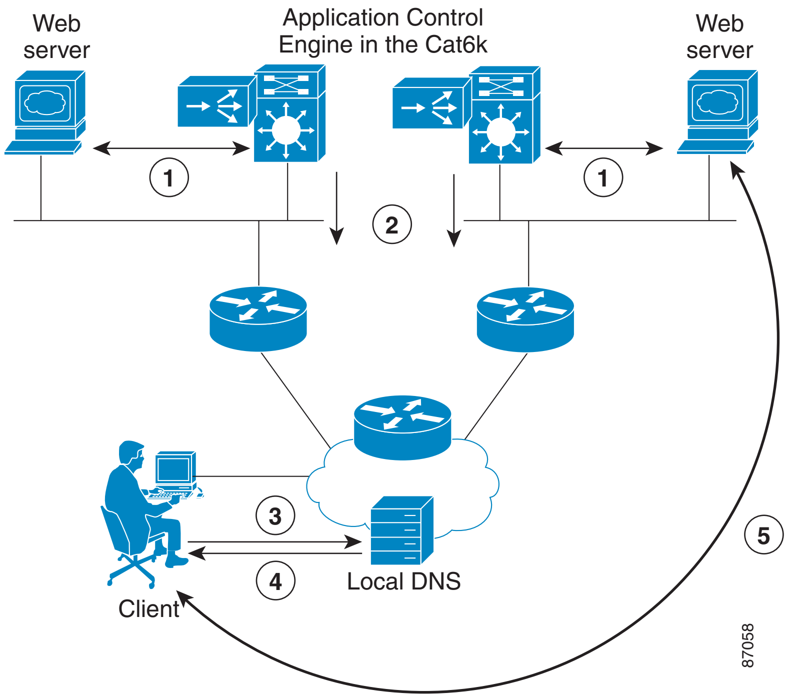
1. Each CSS probes for the server health and is aware of state of the servers and exchange the server availability information using the TCP session.
2. The client requests to resolve www.foo.com.
3. The local DNS server performs the iterative DNS query and the CSS responds with the IP address based on configuration.
4. The client connects to the server to complete the transaction.

Application Control Engine (ACE) for Catalyst 6500

The Cisco Application Control Engine (ACE) integrates advanced Layer 4-7 content switching into the Cisco Catalyst 6500 Series or Cisco 7600 Series Internet Router. The ACE provides high-performance, high-availability load balancing, while taking advantage of the complete set of Layer 2, Layer 3, and QoS features inherent to the platform. The ACE can communicate directly with the Global Site Selctor (GSS), for use in GSLB, and also supports the RHI feature.

Figure 8 provides an overview of how the route health injection works using ACE. When RHI is enabled on ACE, the ACE injects a static route into the MSFC’s routing table. This, in turn, is redistributed by the MSFC.

Figure 8 RHI with the ACE



1. Each ACE probes for the server health and if servers are available, puts in a static route into the MSFC routing table which gets advertised with different metrics from the two Catalyst 6500s (the same IP address gets advertised with different metrics from two locations).
2. The host routes are propagated to the upstream routers and the route with the best metric is used by the upstream routers.
3. The client requests to resolve www.foo.com.
4. The local DNS server performs the iterative DNS query and responds with an IP address.
5. The client connects to the web server on the right because the route is advertised with a better metric.

Conclusion

Site selection ensures that the best data center handles client requests. Each mechanism comes with advantages and disadvantages. There is no generic solution for all site-to-site recovery deployments. Regardless of the site selection mechanism you choose, the Cisco product portfolio supports all three site selection mechanisms.

When deploying the solution, you should consider the following:

- Is it Web based application?
- Is DNS caching an issue?
- Is it an Active-Active site or Active-Standby site?
- All the solutions except for HTTP Redirection redirect traffic to an alternate site based on the reachability/availability of the applications.
- HTTP redirection relies on the HTTP Redirection error code to be received before the client is redirected to an alternate site. In disaster situations this might not be an appropriate solution.

Chapter 3—Site-to-Site Recovery Using DNS

This chapter focuses on the design and deployment of distributed data centers for disaster recovery and business continuance. It explores interoperability between the GSS and the ACE and also provides details of relevant algorithms used in multi-site load distribution. These designs are based on request routing (formerly content routing) products and the ACE server load balancing product.

You can achieve redundancy and high availability by deploying multiple data centers and distributing applications across those data centers. This chapter focuses on the design and deployment of distributed data centers using the Global Site Selector (GSS) and the Application Control Engine (ACE).

Overview

The challenge of site selection to recover from site failures is to ensure that transaction requests from clients are directed to the most appropriate server load balancing device at the geographically distant data center. Geographic Site Selection requires control points for all transaction requests destined to any data center. The point of control for a geographic load-distribution function resides within DNS. Most clients must contact a DNS server to get an IP address to request service from a server. Because, geographically replicated content and applications reside on servers with unique IP addresses, unique DNS responses can be provided to queries for the same URLs or applications based on site or application availability.

Benefits

Site-to-site recovery enables businesses to provide redundancy in case of disasters at the primary data centers. Redundancy and high availability of business critical applications are the key benefits of site-to-site recovery.

Hardware and Software Requirements

The table below lists different hardware and software required to support site-to-site recovery and multi-site load distribution. The GSS interoperates with the ACE and CSS. It also works with other server load balancing products, but some of the features, like the least loaded connections and shared keepalive features, cannot be used with other server load balancers. In subsequent sections of this document, interoperability of the GSS and the ACE is described.

Product	**Release**	**Platforms**
Global Site Selector (GSS)	2.0.2.0.0	GSS-4492
Application Control Engine (ACE)	1.6.1	SLB complex for Catalyst 6K platforms
Cisco Network Registrar (CNR)	6.2.3.2 (this software version was used for testing)	

Design Details

Design Goals

The basic design goal is to be able to direct clients to appropriate data center based on the configured rules and the availability of the servers or services at the data center. The major design issues are:

- Redundancy
- High availability
- Scalability
- Security
- Other requirements as necessary

Redundancy

Within a data center, redundancy is achieved at different layers. It could be link redundancy or device redundancy. Redundancy provides a way of maintaining connectivity if there is a failure in the primary path. This is achieved by deploying devices that support stateful failover.

There are times when the entire data center might go out of service due to an unforeseen reason. In such cases, clients can be directed to a redundant data center. In the event of a data center fail over, it is difficult to provide stateful failover. Although, there is some impact due to the failure of primary data center, the impact is very small compared to not having a redundant design.

High Availability

High availability, from a global server load balancing point of view, is the ability to distribute the load among available data centers. If, for any reason such as a hardware or software failure, over loaded data center etc., it is determined that the new service requests cannot be handled by the data center, the request is directed to a data center that can really handle the service request. It is the constant monitoring of application availability and the load at a data center that helps in achieving high availability.

High availability is also prevalent at each layer of the network including Layer 2 and Layer 3. For a more detailed description of how high availability is achieved within a data center, refer to the *Data Center Networking: Infrastructure SRND:* http://www.cisco.com/en/US/docs/solutions/Enterprise/Data_Center/DC_Infra2_5/DCI_SRND_2_5_book.html.

Scalability

Scalability is an inherent element of distributed data center environment. Applications can be hosted at multiple data centers to distribute the load across multiple data centers and make the applications scalable and highly available. The design should be able to support the growth both in number of sites and the number of DNS records without performance degradation and without over-hauling the design. There are scaling limitations. The details are covered in Implementation Details, page 28.

Security

Security is deployed as an end to end service in the data center. Deploying request routing devices should not compromise security in the data center. For instance, the placement of authoritative DNS should ensure that security requirements are met because of the amount and type of overhead traffic needed to ensure application availability at each data center. Monitoring application availability might include determining both the health and load on the applications.

Other Requirements

Other design requirements include meeting client and application requirements. For a business to business client, the client has to stay with the same site as long as it is available for the length of the transaction period (site persistence). In the case of a client accessing a streaming application, the client should be directed to a topologically closest data center (proximity). Some other client and application requirements include directing clients to the data center based on round trip time, IGP, and BGP metrics. Ideally the design should be able to meet all these requirements.

Design Topologies

Cisco offers several products that are available for multi-site load distribution solutions. There are different topologies based on the products. All the topologies adhere to Cisco's design recommendations and contain a layered approach. The layer which connects to dual service providers is known as the Internet edge[1.]. At each layer, redundant devices are deployed for high availability. The core layer provides connectivity to branch offices, remote users and campus users. This document is focused on the GSS interoperating with the ACE.

Specific topologies covered are site-to-site recovery and multi-site load distribution topologies. Both the topologies look similar except for some minor differences. Cisco recommends the GSS for both site-to-site load distribution and multi-site load distribution. Although the GSS has its limitations, it provides most required features in a single product.

Note The GSS does not support MX records, IGP and BGP metrics. MX records can be supported by forwarding requests to devices that handle MX records such as Cisco Network Registrar (CNR).

Site-to-Site Recovery

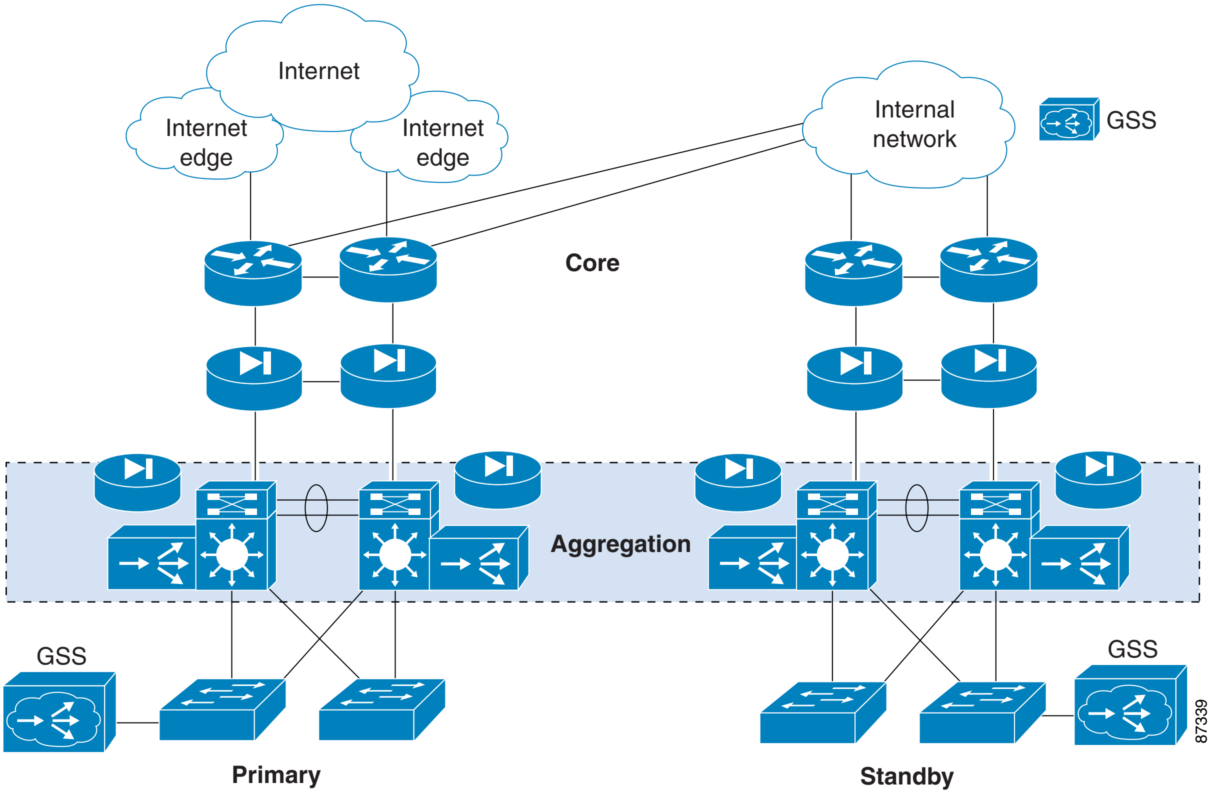
Site-to-site recovery provides a way of recovering data center applications and data in case of an unexpected outage. Sometimes, building redundancy into each layer of networking is not enough. This leads to building standby data centers. Standby data centers host similar applications and databases. You can replicate your data to the standby data center to minimize downtime in the event of an unexpected failure at the primary data center.

Figure 9 depicts the site-to-site recovery topology using the GSS as the request routing device. Typically, the request router is connected to the aggregate switch and the GSS is the authoritative DNS for the domains in the data center. In this example, the GSS is connected to the access switch instead. This is due to the link redundancy limitation. Connecting the request router to the access switch provides some level of redundancy. If the aggregate switch fails, the request router is still reachable. More details are provided in Implementation Details, page 28. Two other common places to deploy the GSS devices are either in the Data Center Security Internet DMZ or an ISP collocation facility. This places the DNS

[1] Multi-homing to dual ISPs is covered in more detail in the *Data Center Networking: Internet Edge Design SRND*.

resolution at the closest edge point so that traffic is not carried all the way to the aggregation or access layers of the data center before being redirected to a secondary site. Typically the primary data center connects to two ISPs through the Internet edge to achieve redundancy and the primary and secondary data center are connected either by WAN or metro optical links to replicate data for recovery in case of disasters at the primary data center. If disaster hits the primary data center, the end users or clients are directed to the secondary data center where the same applications and data is available.

Figure 9 Site-to-Site Recovery



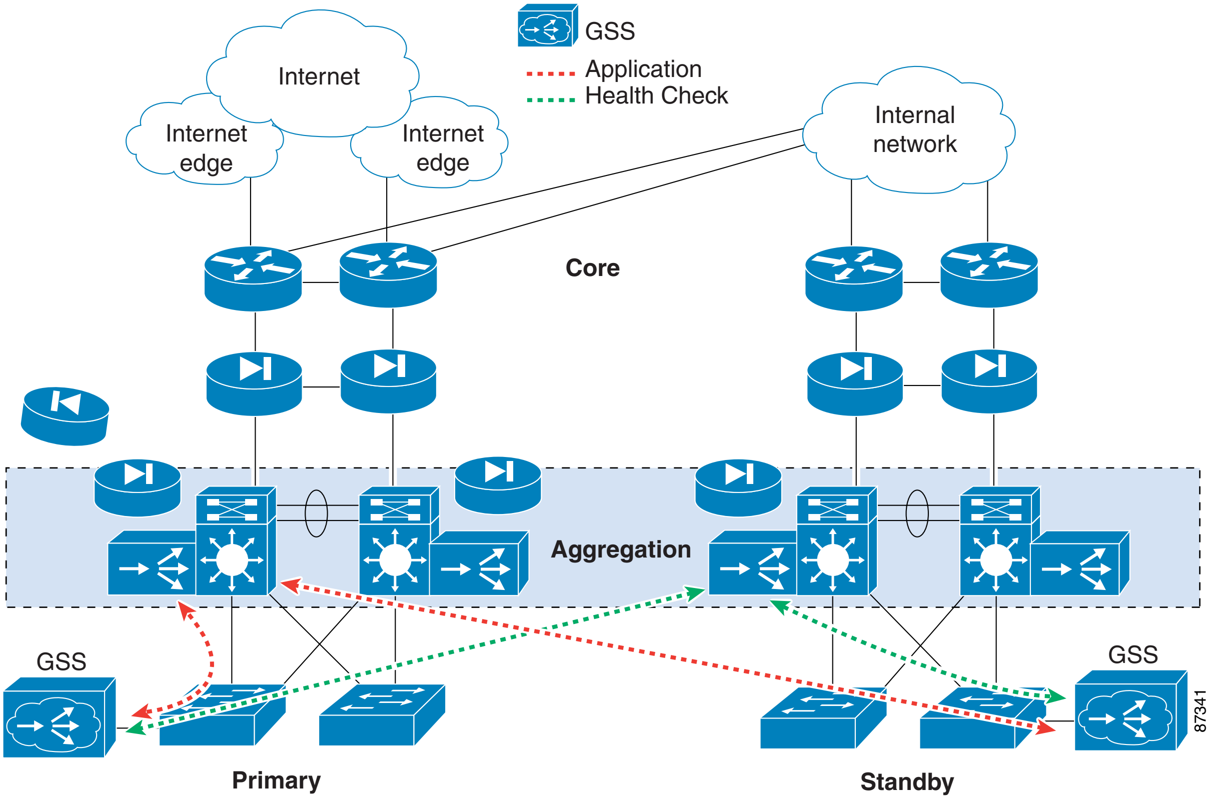
Implementation Details

Before getting into the details, this is an appropriate point to discuss the features common to both site-to-site recovery and multi-site load distribution. The first one to consider are the GSS health probes. It is important to note that the GSS has to send health probes to all sites or data centers or server load balancing devices to learn about application availability. The GSS also offers a shared keepalive. With the shared keepalive, the GSS sends out one request for all the VIPs in a specific data center and gets the response from all the VIPs in the same response. This is depicted in Figure 10.

Note These health probes have to traverse across the firewalls. The GSS configuration guide provides more information for deploying GSS behind firewalls.

Moreover, in case of multi-site load distribution, as the number of sites increase, each GSS has to send health probes to N data centers, N being the number of data centers. The amount of traffic is somewhat alleviated by the use of KAL-AP health probes.

Figure 10 Health Probes from GSS



Primary Standby

Keep in mind that the key design goals, when deploying both site-to-site recovery and multi-site load distribution, are the following:

- Redundancy
- High availability
- Scalability

Redundancy

Typically, redundancy in a single site is provided by deploying a redundant device in active/standby mode. A request router, such as a GSS, ACE, or a CSS, is typically connected to the aggregate switch; these devices can support both link and device redundancy.

To configure the IP address on the GSS interface, use the following command:

```
gss1.ese-cdn.com# conf t
gss1.ese-cdn.com(config)# interface ethernet 0
gss1.ese-cdn.com(config-eth0)# ip address 172.25.99.100 255.255.255.0
```

Note While configuring the interface IP addresses, the global site selector services should be stopped using the **gss stop** command at the enable mode.

Now the default gateway on the GSS has to be configured. The default gateway points to the active HSRP address on the aggregate switches.

```
gss1.ese-cdn.com# conf t
gss1.ese-cdn.com(config)# ip default-gateway 172.25.99.1
```

Figure 11 Link and Device Redundancy

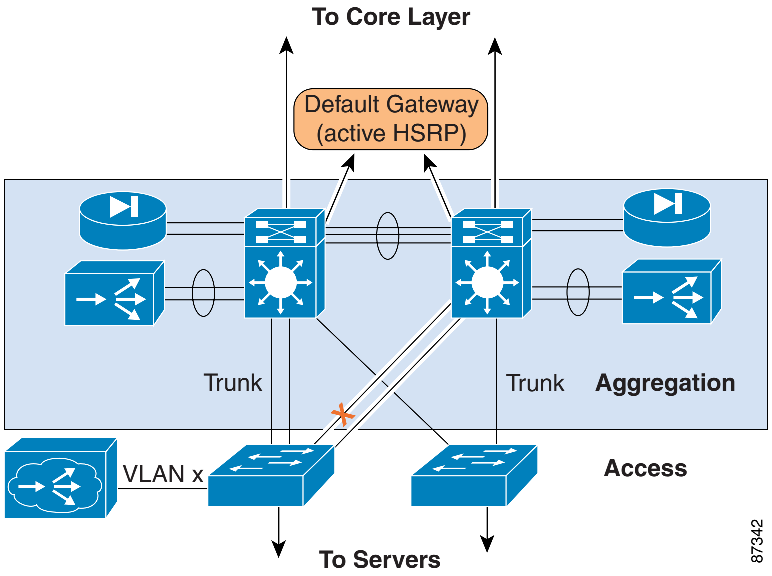


Figure 11 depicts the implementation details. The ACEs are deployed in active/standby configuration with the fault tolerant VLAN carried across the port channel between the two aggregate switches. The PIX firewalls are also deployed in active/standby configuration. With the aggregate switches and the access switches running spanning tree, one of the paths is blocked, as shown in Figure 11. Typically, the aggregate switch on the left is configured as the root for the spanning tree and the aggregate switch on the right is the secondary root for the spanning tree. With this topology, the GSS is deployed at the access switch.

Now, a Layer 3 interface is created on both the aggregate switches for the VLAN and are configured as part of the HSRP group. The aggregate switch on the right hand side is in standby mode. The default gateway on the GSS points to the active HSRP address. This topology minimizes the impact of aggregate switch failure. However, if the access switch fails, even though the spanning tree converges to provide redundancy to the server path, the GSS gets taken out of the picture.

Note There might be more than one client VLAN on the ACE. It is a good idea to put the GSS on a different VLAN.

Alternatively, the GSS can also be connected directly to the aggregate switch but in this case, if the link to the GSS fails, the GSS is out of the picture. With the GSS at the access layer, the GSS is protected from failures to the aggregate switch and failures to the links between the aggregate and the access switches.

High Availability

The secondary GSS deployed at the secondary data center also answers DNS queries. Typically, the upstream DNS round robins are between the primary and secondary GSSs. As long as the primary is active, it responds to DNS queries and directs the end users to the appropriate data center. If the primary GSS goes down for any reason, the secondary GSS continues to answer DNS queries.

Scalability

The GSS can scale up to 2000 authoritative domains and up to 8 GSSs can work together in a network. If there are more than 2 GSSs in the network, one of them is primary, the second one is standby and the remaining GSSs are configured as GSS.

Basic Configuration

Before getting into implementation details, there are a few basic setup steps that must be done on the GSSes. These help in enabling the GUI on the GSS. Only the basic steps that are helpful are described in this section. More details about this are found in the configuration document.

All the content routing information is stored in a SQL database. The database files must be created on the GSSMs before the GUI can be accessed. These are the initial steps to configure a GSS.

Using the SETUP command script you can enter in all of the following information. For those wishing to do a step-by-step command line implementation, the steps are detailed below.

Step 1 Initial configuration like the IP addresses for the interfaces, the default gateway, the host name and the name server is configured on the GSS. The name server has to configured on the GSS for the GSS to work.[1]

Step 2 Create the data base with the **gssm database create** command to enable the graphical user interface on the GSS. This command is executed in the enable mode. Also note that the database is enabled only on the primary and the standby GSS.

Step 3 Configure the node type on the GSS. The node type must be chosen for every GSS in the network. The different node types are primary, standby, or gss.

Step 4 Enable gss with the **gss enable gssm-primary** command. Again, this is done from the enable mode. To follow the activity on the gss, use the **show log follow** and **gss status** commands.

Step 5 Follow steps 1-4 for the standby GSS. In step 4, instead of gssm-primary, use the **gssm-standby** command to enable the GSS and specify the IP address of the primary GSS.

Step 6 Open a browser window and type https://*ip-address-of-gssm-primary* as the URL to access the GUI.

Step 7 The default username is admin and the password is default.

The next step is to configure the health probes, answers, answer group, domain lists and balance methods. This is explained in more detail in both site-to-site recovery and multi-site load distribution sections.

The information below assists you in understanding the relationship between different configuration rules, such as DNS rules, Domain lists etc. The DNS rules consist of the following objects. You can get to each object by clicking on DNS rules and then using the drop down menu on the left hand side.

[1] Refer to the GSS configuration guide for more information about the name server.

- Source Address List—A list of addresses of local DNS. For site-to-site recovery, this can be set to accept all IP addresses. This represents the source that is requesting the IP address of the domain.
- Domain List—A list of domains. This represents the list of domains, one of which matches the domain name requested.
- Answer Group—A group of resources from which the answers are provided.
- Balance Method—The global load balancing algorithm that is used to balance responses among the answer groups.
- Answers—Configure different VIPs here along with the type of keep alive method used.
- Shared Keepalives—Specifies the IP address on the load balancer to which the KAL-AP health probes are sent.

Both site-to-site recovery and multi-site load distribution use health probes. The different types of health probes used are shared keepalive and ICMP. Shared keepalive is also called as KAL-AP. There are two types of KAL-Aps: KAL-AP by VIP and KAL-AP by tag. Shared keepalives can be set up either using KAL-AP by VIP or KAL-AP by tag. KAL-AP by VIP uses the VIP and KAL-AP by tag uses a domain string. For KAL-AP by tag, some additional configuration is required on the ACE, which is the load balancer. The tag specifies the sub-domain and the length of the tag has to be less than 64 characters. This is because the KAL-AP query limits the length of the tag to 64 characters. The idea behind using the domain as a tag is to probe the health of the VIP by domain name instead of the IP address (KAL-AP by VIP). This comes in handy if the addresses are being translated between the GSS and the load balancer.

The required configuration on the ACE is as follows:

```
Probe icmp REAL_Servers
 ip address 10.10.100.1
interval 2
faildetect 1
passdetect interval 2
passdetect count 1

rserver host test
ip address 10.10.100.1
probe REAL_Servers
inservice

serverfarm host REAL_Servers
rserver test
inservice

Class-map VIP_200
2 match virtual-address 20.17.30.201 any

Policy-map type  loadbalance  http first-match  real.pol
Class class-default
Serverfarm REAL_Servers

Policy-map multi-match test
Class  VIP_200
Loadbalance vip inservice
Loadbalance policy real.pol
```

Note When setting up shared keepalives for ACE, the Primary IP address used can be either the IP address of the client VLAN or the alias IP address of the client VLAN. Also note that if the Content Services Switch (CSS) is used instead of ACE, use the circuit IP addresses on the CSS in the primary and secondary boxes of shared keep alive configuration.

Site-to-Site Recovery

In a site-to-site recovery solution, typically, the primary site is the active site and all the end users are directed to the primary site as long as the applications are alive and well. The secondary site, or the recovery site, also hosts the applications but these are in a standby mode. The data is replicated to the standby data center to be used in the event of unexpected downtime at the primary data center.

The site-to-site recovery topology was introduced in Figure 3-1. In this section, the implementation details for a single site are provided. This design also applies to the secondary site. The only difference is the configurations on the GSS itself: one is configured as the primary and the second GSS is configured as the standby GSS.

Note GSS is deployed as authoritative DNS for the sub-domains for critical applications. This implies that the IP addresses of the authoritative DNS have to be configured in the upstream DNS as name servers. Typically there is more than one name server in the upstream DNS. During the DNS resolution process, the upstream DNS uses the round trip time as a measure to query one of the name servers. Site-to-site recovery is always based on active-standby configurations and regardless of which GSSes are queried, the result should be the same.

Site Selection Method

The site selection method or balance method, also known as predictor, is the algorithm that is followed while answering DNS queries from the clients. For site-to-site recovery, where the data centers are in active/standby mode, a balance method called the ordered list is used.

Using the ordered list balance method, each resource within an answer group (for example, an SLB VIP or a name server) is assigned a number that corresponds to the rank of that answer within the group. Devices with lower numbers rank above those with higher numbers.

Using the rankings, the GSS tries each resource in the order that has been prescribed, selecting the first available (“live”) answer to serve a user request. List members are given precedence and tried in order. A member will not be used unless all previous members fail to provide a suitable result.

Configuration

GSSes are configured using GUI. It is difficult to show all the screens in this section to discuss the configurations. Refer to www.cisco.com for detailed descriptions of the GUI configuration.

Note The TTL (time to live) values configured on the GSS determines for how long the A records are cached. For site-to-site recovery, setting the TTL to a low value will ensure that a new request is made after the TTL expiration. It should also be noted that the lowest value of health probe interval that can be set on the GSS is 45 seconds. The recommended value for TTL on the GSS has to be between 5 and 45 seconds.

This section provides a configuration outline.

Step 1 Perform initial configurations as described above.

Step 2 On the web browser, use HTTPS to enter the IP address of your primary GSS and login.

Step 3 Once you are on the GSS, the domain list, answer group, balance methods, answers, and shared keepalives have to be configured either by selecting each of the items individually or by using the wizard.

Step 4 Configure the VIPs by selecting the "answers" option in the drop down menu for which health probes have to be sent.

The type of health probes used is also configured here. Different options available are 1. No health probe, 2. ICMP, 3. KAL-AP by VIP and Tag, 4. HTTP-HEAD. For more information on health probes refer Basic Configuration, page 31.

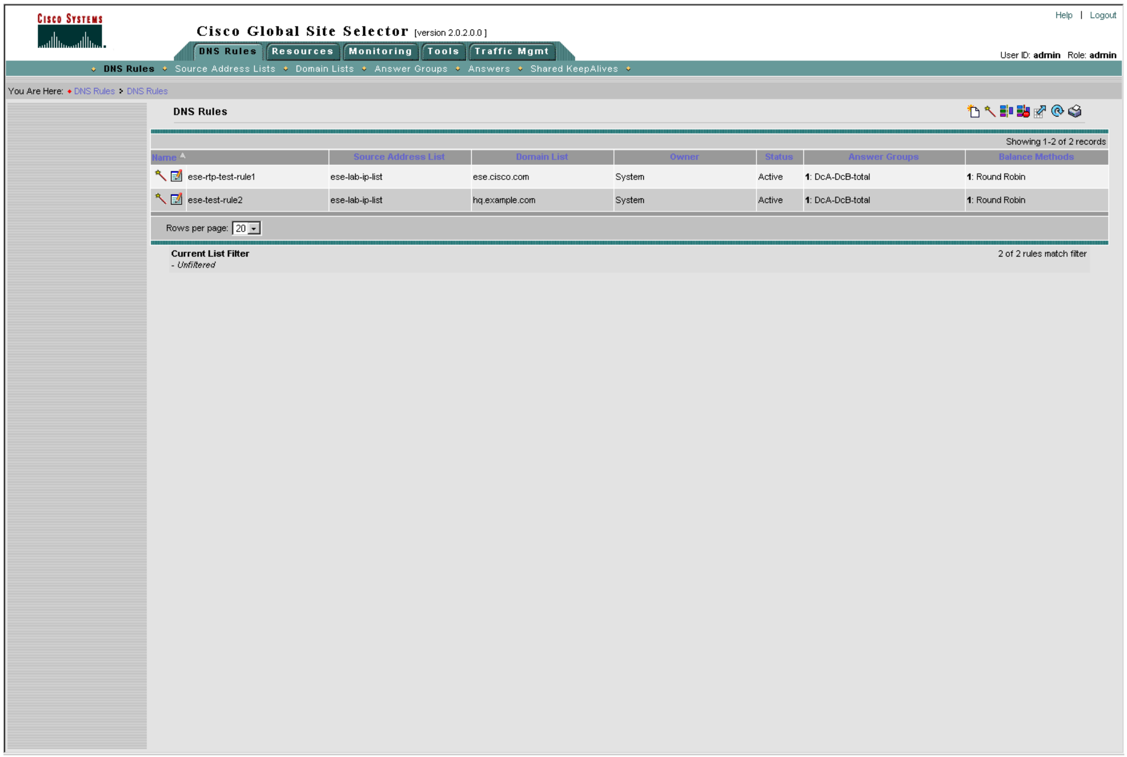
Step 5 Select the answer groups and configure the members of the answer group. The members of the answer group are the VIPs.

Step 6 Select the DNS rules from the drop down menu and tie all the information together.

The way the DNS rules read when translated to simple english is "For all the clients which belong to this source list, looking for the sub-domain in the domain list, if the status is active, select one from this answer group based on the this balance method." 1. For site-to-site recovery, always select ordered list for balance method. As long as the VIP is alive at the primary site, all clients are directed towards the primary site.2. The first active address is the primary site's VIP.3. The second active address is the secondary or standby data center's VIP.

Step 7 Once the configuration is complete, click on Monitoring to view the health information of all the different VIPs. The following sections describe the configurations for different balance methods.

Figure 12 *GSS GUI*



DNS Rules

Showing 1-2 of 2 records

Name	Source Address List	Domain List	Owner	Status	Answer Groups	Balance Methods
ese-rtp-test-rule1	ese-lab-ip-list	ese.cisco.com	System	Active	1: DcA-DcB-total	1: Round Robin
ese-test-rule2	ese-lab-ip-list	hq.example.com	System	Active	1: DcA-DcB-total	1: Round Robin

Rows per page: 20

Current List Filter
- *Unfiltered*

2 of 2 rules match filter

Figure 12 depicts the GSS screen. The drop down menu is on the left hand side of the screen. In the illustration, you can also see the rule wizard, monitoring, and so forth.

Conclusion

Unlike other global server load balancing devices, the GSS provides all the features in the same chassis. GSS also provides the decoupling between server load balancer and the global load balancing devices. GSS interoperates well with the ACE. GSS also provides most of the features and is easier to configure. GSS does support application high availability, load distribution, and business resilience. Except for a few features, GSS does meet most of the requirements today. GSS 2.0 and above also offer the ability to integrate Cisco Network Registrar (CNR) to provide for authoritative DNS services on one platform.

Chapter 4—Multi-Site Load Distribution Using DNS

You can achieve redundancy and high availability by deploying multiple data centers and distributing applications across those data centers. This design document focuses on the design and deployment of distributed data centers using the Global Site Selector (GSS) and the Application Control Engine (ACE).

This chapter explores interoperability between the GSS and the ACE and also provides details of relevant algorithms used in multi-site load distribution. These designs are based on request routing products and the ACE server load balancing product.

Overview

The challenge of geographic load balancing is to ensure that transaction requests from clients are directed to the most appropriate server load balancing device at the geographically distant data center. Geographic load distribution requires control points for all transaction requests destined to any data center. The point of control for a geographic load-distribution function resides within DNS. Most clients must contact a DNS server to get an IP address to request service from a server. Because, geographically replicated content and applications reside on servers with unique IP addresses, unique DNS responses can be provided to queries for the same URLs or applications based on a series of criteria. These different criteria are based on the availability of applications at different data centers and different metrics. The different metrics include proximity, weighted round robin, preferred data centers, load at the data center, etc. The different metrics are dynamically calculated and updated at distributed sites. Based on these different metrics and the availability of services, clients are directed to the best site.

Benefits

Redundancy, scalability, and high availability are the key benefits of multi-site load distribution. Site-to-site recovery enables businesses to provide redundancy in case of disasters at the primary data centers. Multi-site load distribution provides application high availability and scalability. Multi-Site load distribution provides these benefits by making individual sites look like a single server and getting application availability and load information from these servers. This makes it possible to deploy multiple inexpensive devices rather than one large expensive system, providing for incremental scalability and higher availability.

Hardware and Software Requirements

The table below lists different hardware and software required to support site-to-site recovery and multi-site load distribution. The GSS interoperates with the ACE and CSS. It also works with other server load balancing products, but some of the features, like the least loaded connections and shared keepalive features, cannot be used with other server load balancers. In subsequent sections of this document, interoperability of the GSS and the ACE is described.

Product	Release	Platforms
Global Site Selector (GSS)	2.0.2.0.0 (this software version was used for testing)	GSS-4492R
Application Control Engine (ACE)	1.6.1 (this software version was used for testing)	SLB complex for Catalyst 6K platforms
Cisco Network Registrar (CNR)	6.2.3.2 (this software version was used for testing)	

Design Details

Design Goals

The basic design goal is to be able to direct clients to appropriate data center based on the configured rules and the availability of the servers or services at the data center. The major design issues are:

- Redundancy
- High availability
- Scalability
- Security
- Other requirements as necessary

Redundancy

Within a data center, redundancy is achieved at different layers. It could be link redundancy or device redundancy. Redundancy provides a way of maintaining connectivity if there is a failure in the primary path. This is achieved by deploying devices that support stateful failover.

There are times when the entire data center might go out of service due to an unforeseen reason. In such cases, clients can be directed to a redundant data center. In the event of a data center fail over, it is difficult to provide stateful failover. Although, there is some impact due to the failure of primary data center, the impact is very small compared to not having a redundant design.

High Availability

High availability, from a global server load balancing point of view, is the ability to distribute the load among available data centers. If, for any reason such as a hardware or software failure or over loaded data center, it is determined that the new service requests cannot be handled by the data center, the request is directed to a data center that can really handle the service request. It is the constant monitoring of application availability and the load at a data center that helps in achieving high availability.

High availability is also prevalent at each layer of the network including Layer 2 and Layer 3. For a more detailed description of how high availability is achieved within a data center, refer to the *Data Center Networking: Infrastructure Architecture SRND:* http://www.cisco.com/en/US/docs/solutions/Enterprise/Data_Center/DC_Infra2_5/DCI_SRND_2_5_book.html.

Scalability

Scalability is an inherent element of distributed data center environment. Applications can be hosted at multiple data centers to distribute the load across multiple data centers and make the applications scalable and highly available. The design should be able to support the growth both in number of sites and the number of DNS records without performance degradation and without over-hauling the design. There are scaling limitations. The details are covered in Implementation Details, page 39.

Security

Security is deployed as an end-to-end service in the data center. Deploying request routing devices should not compromise security in the data center. For instance, the placement of authoritative DNS should ensure that security requirements are met because of the amount and type of overhead traffic needed to ensure application availability at each data center. Monitoring application availability might include determining both the health and load on the applications.

Other Requirements

Other design requirements include meeting client and application requirements. For a business to business client, the client has to stay with the same site as long as it is available for the length of the transaction period (site persistence). In the case of a client accessing a streaming application, the client should be directed to a topologically closest data center (proximity). Some other client and application requirements include directing clients to the data center based on round trip time, IGP, and BGP metrics. Ideally the design should be able to meet all these requirements.

Design Topologies

Cisco offers several products that are available for multi-site load distribution solutions. There are different topologies based on the products. All the topologies adhere to Cisco’s design recommendations and contain a layered approach. The layer which connects to dual service providers is known as the Internet edge (multi-homing to dual ISPs is covered in more detail in the *Data Center Networking: Internet Edge Design* SRND). At each layer, redundant devices are deployed for high availability. The core layer provides connectivity to branch offices, remote users and campus users. This document is focussed on the GSS interoperating with the ACE.

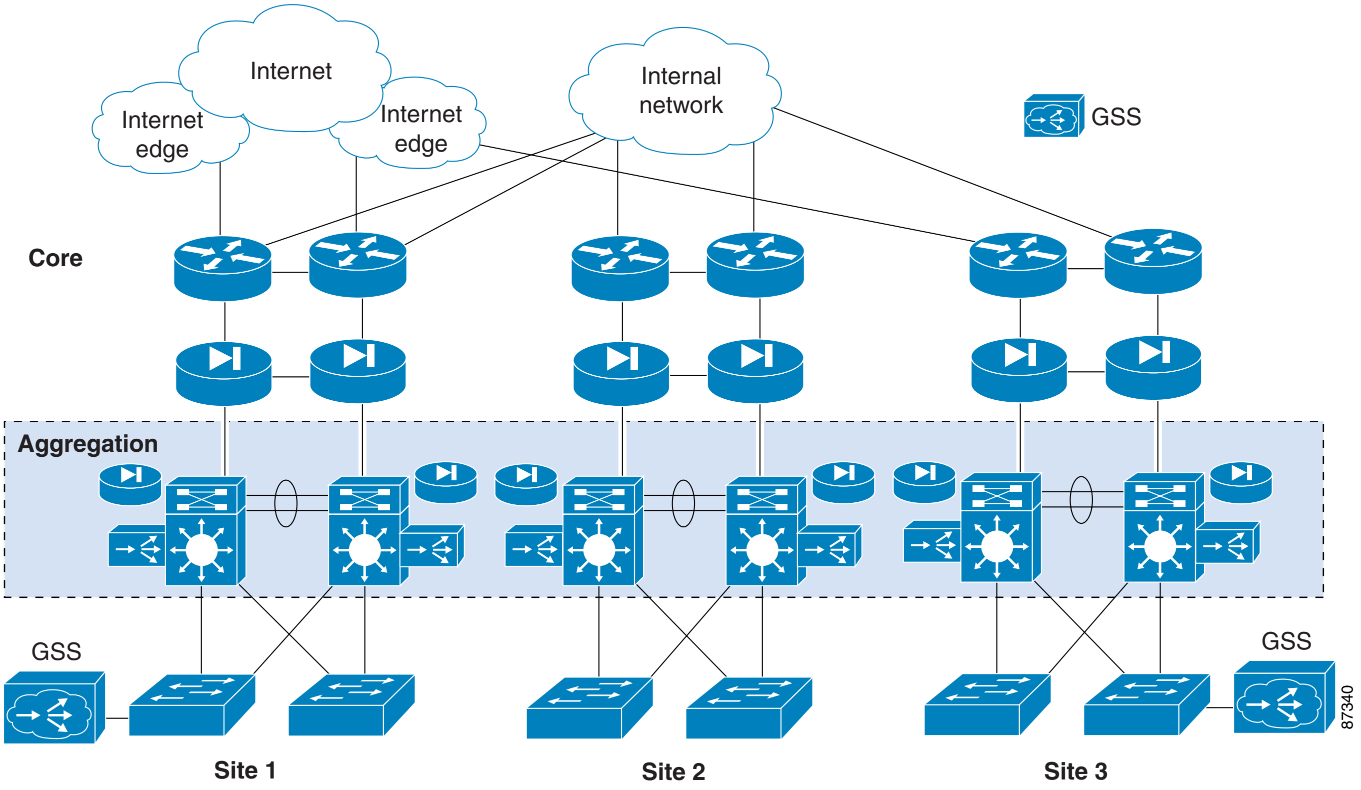
Specific topologies covered are site-to-site recovery and multi-site load distribution topologies. Both the topologies look similar except for some minor differences. Cisco recommends the GSS for both site-to-site load distribution and multi-site load distribution. Although the GSS has its limitations, it provides most required features in a single product.

Note The GSS does not support MX records, IGP and BGP metrics. MX records can be supported by forwarding requests to devices that handle MX records such as Cisco Network Registrar (CNR).

Multi-Site Load Distribution

Figure 4-1 depicts the topology for multi-site load distribution. The GSSes are overlaid on top of the existing data center infrastructures. More details about the data center topologies and the Internet edge can be found in the *Enterprise Data Center Infrastructure* SRND and the *Data Center Networking: Internet Edge* SRND.

Figure 13 Multi-site Load Distribution Using GSS



Site 1, Site 2, Site 3

There is no difference between the site-to-site recovery and multi-site load distribution topologies except that a GSS per site is not required to support multi-site load distribution. The GSS in Site 1 is the primary GSS and the GSS in Site 3 is the secondary GSS. There is no permanent session between the GSSes. But, after a configuration change on the primary GSS, the secondary GSS synchronizes with the primary GSS. Further, the configuration changes can only be made on the primary as long as the primary is up and running.

If the primary GSS goes down for some reason, the configuration changes can be made on the secondary. As in site-to-site recovery, the GSSes are connected to the access switch for the lack of good link redundancy mechanisms on the GSS. GSSes can be configured with different DNS rules. Each DNS rule can select a different predictor algorithm. Based on the DNS rules and the predictors used for the DNS rules, the GSSes, both primary and secondary, respond to DNS queries from the end users or clients. The responses lead the client into the appropriate data center.

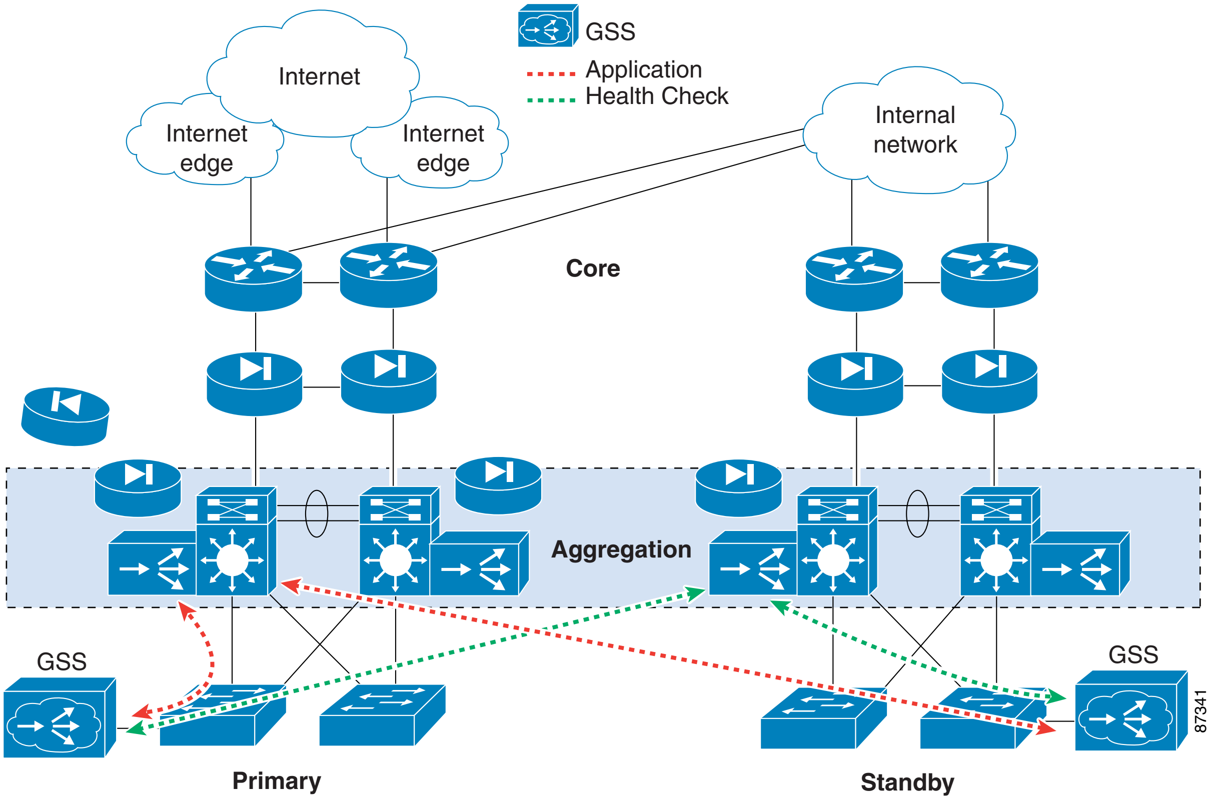
Implementation Details

Before getting into the details, this is an appropriate point to discuss the features common to both site-to-site recovery and multi-site load distribution. The first one to consider are the GSS health probes. It is important to note that the GSS has to send health probes to all sites or data centers or server load balancing devices to learn about application availability. The GSS also offers a shared keepalive. With the shared keepalive, the GSS sends out one request for all the VIPs in a specific data center and gets the response from all the VIPs in the same response. This is depicted in Figure 14.

Note These health probes have to traverse across the firewalls. The GSS configuration guide provides more information for deploying GSS behind firewalls.

Moreover, in case of multi-site load distribution, as the number of sites increase, each GSS has to send health probes to N data centers, N being the number of data centers. The amount of traffic is somewhat alleviated by the use of KAL-AP health probes.

Figure 14 Health Probes from GSS



Keep in mind that the key design goals, when deploying both site-to-site recovery and multi-site load distribution, are the following:

- Redundancy
- High availability
- Scalability

Redundancy

Typically, redundancy in a single site is provided by deploying a redundant device in active/standby mode. A request router, such as an ACE or a CSS, is typically connected to the aggregate switch; these devices can support both link and device redundancy. The GSS cannot be deployed in active/standby mode in a single site. Due to this limitation, the GSS can be deployed as in one of the following ways:

- Connected to the Aggregate Switch
- Connected to the Internet edge DMZ
- Connected to the Access Switch

In all of these cases, GSS is pointing to the default gateway, the active HSRP address on the aggregate switch or to the DMZ router. Logically the configuration is the same whether the GSS is connected to the aggregate or the access switch. If the GSS is connected to the aggregate switch, a Layer 3 and above failure does not break connectivity to the GSS. GSS will now point to the active HSRP address on the aggregate switch. However if the aggregate switch fails, the connectivity to GSS is broken.

The arguments in favor of connecting the GSS to the access switch are as follows:

- The probability of an aggregate switch failure is higher compared to an access switch simply because of the number of elements in the aggregate switch.
- With the GSS connected to the access layer, logically it resides outside the MSFC and has a lower probability of failure.

To configure the IP address on the GSS interface, use the following command:

```
gss1.ese-cdn.com# conf t
gss1.ese-cdn.com(config)# interface ethernet 0
gss1.ese-cdn.com(config-eth0)# ip address 172.25.99.100 255.255.255.0
```

Note While configuring the interface IP addresses, the global site selector services should be stopped using the **gss stop** command at the enable mode.

Now the default gateway on the GSS has to be configured. The default gateway points to the active HSRP address on the aggregate switches.

```
gss1.ese-cdn.com# conf t
gss1.ese-cdn.com(config)# ip default-gateway 172.25.99.1
```

Figure 15 Link and Device Redundancy

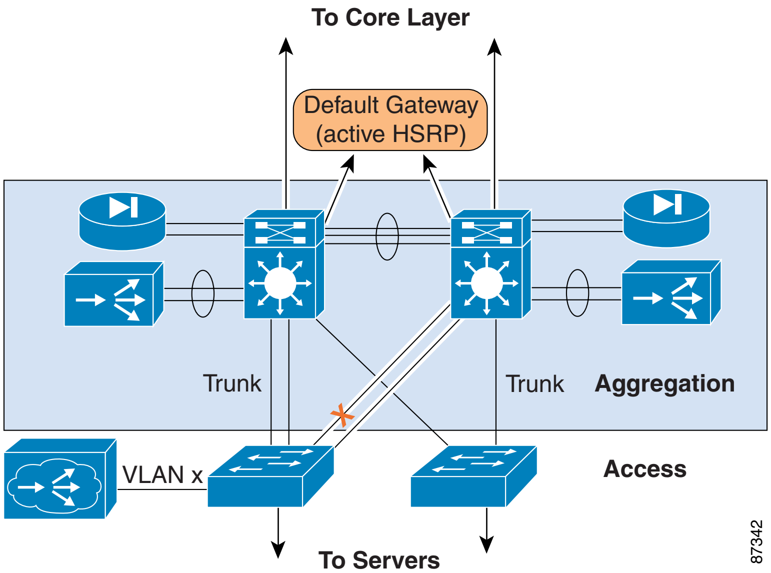


Figure 15 depicts the implementation details. The ACEs are deployed in active/standby configuration with the fault tolerant VLAN carried across the port channel between the two aggregate switches. The PIX firewalls are also deployed in active/standby configuration. With the aggregate switches and the access switches running spanning tree, one of the paths is blocked, as shown in Figure 15 with the red VLAN. Typically, the aggregate switch on the left is configured as the root for the spanning tree and the aggregate switch on the right is the secondary root for the spanning tree. With this topology, the GSS is deployed at the access switch and is part of the red VLAN.

Now a Layer 3 interface is created on both the aggregate switches for the red VLAN and are configured as part of the HSRP group. The aggregate switch on the right hand side is in standby mode. The default gateway on the GSS points to the active HSRP address. This topology minimizes the impact of aggregate switch failure. However, if the access switch fails, even though the spanning tree converges to provide redundancy to the server path, the GSS gets taken out of the picture.

Note There might be more than one client VLAN on the ACE. It is a good idea to put the GSS on a different VLAN.

Alternatively, the GSS can also be connected directly to the aggregate switch but in this case, if the link to the GSS fails, the GSS is out of the picture. With the GSS at the access layer, the GSS is protected from failures to the aggregate switch and failures to the links between the aggregate and the access switches.

High Availability

The secondary GSS deployed at the secondary data center also answers DNS queries. Typically, the upstream DNS round robins are between the primary and secondary GSSes. As long as the primary is active, it responds to DNS queries and directs the end users to the appropriate data center. If the primary GSS goes down for any reason, the secondary GSS continues to answer DNS queries.

Scalability

The GSS can scale up to 2000 authoritative domains and up to 8 GSSes can work together in a network. If there are more than 2 GSSes in the network, one of them is primary, the second one is standby and the remaining GSSes are configured as gss.

Basic Configuration

Before getting into implementation details, there are a few basic setup steps that must be done on the GSSes. These help in enabling the GUI on the GSS. Only the basic steps that are helpful are described in this section. More details about this are found in the configuration document.

All the content routing information is stored in a SQL database. The database files must be created on the GSSMs before the GUI can be accessed. These are the initial steps to configure a GSS.

Using the SETUP command script you can enter in all of the following information. For those wishing to do a step by step command line implementation the steps are detailed below.

Step 1 Initial configuration like the IP addresses for the interfaces, the default gateway, the host name and the name server is configured on the GSS. The name server has to configured on the GSS for the GSS to work.[1]

Step 2 Create the data base with the **gssm database create** command to enable the graphical user interface on the GSS. This command is executed in the enable mode. Also note that the database is enabled only on the primary and the standby GSS.

Step 3 Configure the node type on the GSS. The node type must be chosen for every GSS in the network. The different node types are primary, standby, or gss.

Step 4 Enable gss with the **gss enable gssm-primary** command. Again, this is done from the enable mode. To follow the activity on the gss, use the **show log follow** and **gss status** commands.

Step 5 Follow steps 1-4 for the standby GSS. In step 4, instead of gssm-primary, use the **gssm-standby** command to enable the GSS and specify the IP address of the primary GSS.

Step 6 Open a browser window and type https://<ip-address-of-gssm-primary> as the URL to access the GUI.

Step 7 The default username is admin and the password is default.

The next step is to configure the health probes, answers, answer group, domain lists and balance methods. This is explained in more detail in both site-to-site recovery and multi-site load distribution sections.

The information below assists you in understanding the relationship between different configuration rules, such as DNS rules, Domain lists etc. The DNS rules consist of the following objects. You can get to each object by clicking on DNS rules and then using the drop down menu on the left hand side.

- Source Address List—A list of addresses of local DNS. For site-to-site recovery, this can be set to accept all IP addresses. This represents the source that is requesting the IP address of the domain.
- Domain List—A list of domains. This represents the a list of domains, one of which matches the domain name requested.
- Answer Group—A group of resources from which the answers are provided.
- Balance Method—The global load balancing algorithm that is used to balance responses among the answer groups.

[1] Refer to the GSS configuration guide for more information on the name server.

- Answers—Configure different VIPs here along with the type of keep alive method used.
- Shared Keepalives—Specifies the IP address on the load balancer to which the KAL-AP health probes are sent.

Both site-to-site recovery and multi-site load distribution use health probes. The different types of health probes used are shared keepalive and ICMP. Shared keepalive is also called as KAL-AP. There are two types of KAL-Aps: KAL-AP by VIP and KAL-AP by tag. Shared keepalives can be set up either using KAL-AP by VIP or KAL-AP by tag. KAL-AP by VIP uses the VIP and KAL-AP by tag uses a domain string. For KAL-AP by tag, some additional configuration is required on the ACE, which is the load balancer. The tag specifies the sub-domain and the length of the tag has to be less than 64 characters. This is because the KAL-AP query limits the length of the tag to 64 characters. The idea behind using the domain as a tag is to probe the health of the VIP by domain name instead of the IP address (KAL-AP by VIP). This comes in handy if the addresses are being translated between the GSS and the load balancer.

The following is the required configuration on the ACE:

```
Probe icmp REAL_Servers
 ip address 10.10.100.1
interval 2
faildetect 1
passdetect interval 2
passdetect count 1

rserver host test
ip address 10.10.100.1
probe REAL_Servers
inservice

serverfarm host REAL_Servers
rserver test
inservice

Class-map VIP_200
2 match virtual-address 20.17.30.201 any

Policy-map type  loadbalance  http first-match  real.pol
Class class-default
Serverfarm REAL_Servers

Policy-map multi-match test
Class  VIP_200
Loadbalance vip inservice
Loadbalance policy real.pol
```

Note When setting up shared keepalives for ACE, the Primary IP address used can be either the IP address of the client VLAN or the alias IP address of the client VLAN. Also note that if the Content Services Switch (CSS) is used instead of ACE, use the circuit IP addresses on the CSS in the primary and secondary boxes of shared keep alive configuration.

Multi-Site Load Distribution

Figure 13 depicts a multi-site load distribution deployment with a primary and a standby GSS. The advantage of using GSS for multi-site load distribution is that it provides an integrated feature set compared to deploying global site load balancing using the CSS. However, on the flip side, the health probes are sent across the data centers.

Multi-site load distribution provides redundancy, high availability, and scalability. Deploying multi-site load distribution has similarities to site-to-site recovery deployments. Providing redundancy for GSSes in a multi-site load distribution is identical to deploying redundancy in site-to-site recovery.

Note Providing redundancy in a single site is identical to what was described earlier. Refer to Redundancy, page 40 for detailed information.

From a high availability and scalability perspective, multi-site load distribution offers more site selection methods and scales well due to the number of sites and number of GSSes that can be deployed.

When there are two or more sites to share the load, similar to a server farm, there are multiple predictor algorithms, balance methods, or site selection methods that can be used.

Site Selection Methods

The GSS, acting as an authoritative DNS, monitors the availability of different VIPs at different sites. Upon receiving a DNS request, the GSS responds with an A record of the active VIPs based on one of the following criteria.

- Round robin
- Weighted round robin
- Ordered list
- Least loaded
- Hashed
- Proximity
- Static
- Based on round trip time (Boomerang)

Least loaded, hashed, and proximity are covered in this section. Other balance methods are prevalent or have been discussed earlier in the document. Further, it was noted, in an earlier section, that the authoritative DNS is queried based on the round trip time. This is true in case of multi-site load distribution as well. The response for the query is really based on the different site selection methods. For example, if least loaded site selection method is configured on all the GSSes, regardless of which authoritative DNS is queried, the response is based on the load at different sites. If static proximity is used instead, the response is based on the IP address of the querying device.

Least Loaded

Clients are directed to a site with the least load. The definition of load is based on the load balancing device used at the data center.

- Calculate the maximum capacity for a given virtual server:

  max_capacity = For each inservice real add 10,000.
  (i.e., 1 inservice real=10000, 2 inservice reals = (2 X 10000) = 20000, etc.)
- Calculate the factor:
- Calculate DFP (Dynamic Feedback Protocol) Weight:

  factor = ((max_capacity - CurrentConnectionCount) << 10) / max_capacityweight = (65535 * factor) >> 10

  This returns a value in the range of 1 to 64K-1 with 65535 meaning MOST available.

- This weight has to be mapped to a range between 2-254 called capp_weight, with 2 being MOST available as follows:

  capp_weight = weight >> 8
  if capp_weight is less than 2 assign capp_weight = 2;

As an example, consider that there is one server in a server farm. This implies that the maximum number of connections is equal to 10000. i.e., max_capacity =10000.

Consider that there are 5000 connections going through the switch, the factor is calculated as follows:

```
Factor = ((10000 – 5000) << 10) / 10000

 = 5120000 / 10000 = 512
"<<" stands for left shift and  ">>" stands for right shift
weight = (65535 * 512) >> 10

 = 32767
capp_weight = 32767 >> 8; right shift 32767 by 8 bits, i.e., right shift
111111111111111 by 8 bits

= 127 in decimal or 1111111 in binary
```

This provides a measure of availability for different VIPs at different sites. When responding to the DNS query, the GSS looks at the available load information. The GSS responds with an A record with the IP address of the VIP, which has the lowest value. If there is more than one VIP with the same load, the GSS performs round robin between the VIPs.

It is important to note that the maximum number of connections allowed when using an ACE is about 4 million. To limit the number of connections to a maximum per server based on the server capacity, use maximum connections command option on the ACE for each real server. More details are provided in the configuration section.

Hashed

Using the source address and domain hash balance method, elements of the client's DNS proxy, IP address and the requesting client's domain, are extracted and used to create a unique value, referred to as a hash value. The unique hash value is attached to and used to identify a VIP that is chosen to serve the DNS query. The use of hash values makes it possible to “stick” traffic from a particular requesting client to a specific VIP, ensuring that future requests from that client are routed to the same VIP. This type of feature lets the client stick to a specific site. If there are two or more VIPs for the specified domain and the site sticky site goes out of service, the GSS picks an available VIP to go to based on the hash value and sticks to it.

Note The health probes are sent at regular intervals. If the sticky VIP goes down, the GSS learns the status of the sticky VIP on the next health probe. During this window, the GSS directs the clients to the sticky VIP.

Proximity

Clients matching a list of IP addresses in the source address list are directed to specific sites. This is called Static Proximity. The second category of proximity, called boomerang, is to direct clients to the site with the least round trip time between the requesting client (client’s DNS proxy) and the site.

Note GSS does not provide site selection methods based on IGP and BGP metrics.

Configuration

You configure the GSS with a GUI. It is difficult to show all the screens in this section to discuss the configurations. Refer to http://www.cisco.com/en/US/products/hw/contnetw/ps4162/index.html for a detailed description of the configuration.

The various objects of DNS rules have to be configured and tied together. You can get to each object by clicking on DNS rules and then using the drop down menu on the left hand side.

In this section a configuration outline is provided for different site selection or balance methods.

Step 1 Perform initial configurations as described above.

Step 2 On the web browser, use https to enter the IP address of your primary GSS (gssm primary), and login. Once you are on the GSS, the different parameters can be configured individually or by using the wizard.

Step 3 Configure the VIPs by selecting the “answers” option in the drop down menu for which health probes have to be sent.

The type of health probes used is also configured here. Different options available are 1. No health probe, 2. ICMP, 3. KAL-AP by VIP and Tag, 4. HTTP-HEAD.

Step 4 Select the answer groups and configure the members of the answer group. The members of the answer group are the VIPs.

Step 5 Select the domain list and configure the sub-domains.

Step 6 Optionally, the source list can also be configured if needed. If this option is not used, by default it applies to all requests.

Step 7 Select the DNS rules from the drop-down menu and tie all the information together. The way the DNS rules read when translated to simple english is “For all the clients that belong to this source list, looking for the sub-domain in the domain list, if the status is active, select one from this answer group based on the this balance method.

Step 8 Once the configuration is complete, click on monitoring to view the health information of all the different VIPs. The following sections describe the configurations for different balance methods.

Least Loaded Configuration

The GSS relies on the load balancing device to learn the load at a specific site. The load information is obtained by using the UDP based KAL-AP health probes. Since the load information is obtained from the load balancer, some configuration is required on the load balancers as well. This document uses the ACE to test the GSS. As a result, the configuration discussed here applies only to ACE. Configuration on CSS might not look the same.

Configuring the ACE

Step 1 Configure ACE for SLB either in bridged or routed mode.

Step 2 Configure the maximum number of connections on the real servers based on how many connections they can safely handle.

```
rserver host test
ip address 10.10.100.1
conn-limit max 1000 min 200
```

Step 3 Enable **capp udp**.

Configuring the GSS

Step 1 Follow the basic configuration steps for the initial configuration.

Step 2 Configure the shared keepalives and use KAL-AP by VIP.

For the ACE, only the primary address can be used if an alias is set up for the client side VLAN. (An alias IP is the virtual IP for an active standby setup). The ACE does respond to the KAL-AP requests to the alias IP.

Step 3 Configure the VIPs by selecting the Answers option in the drop down menu for the configured health probes.

Step 4 Select the answer groups and configure the members of the answer group. The members of the answer group are the VIPs.

Step 5 Select the domain list and configure the sub-domains.

Step 6 Optionally, the source list can also be configured if needed. If this option is not used, by default it applies to all requests from any source.

Step 7 Select the DNS rules from the drop down menu and tie all the information together. The way the DNS rules read when translated to simple english is “For all the clients which belong to this source list, looking for the sub-domain in the domain list, if the status is active, select one from this answer group based on the this balance method.”

Step 8 Click **Rule Builder** and enter all the relevant information (source address list, domain list and balance clause). For balance clause, choose least loaded balance method.

Note The ACE can handle up to 4 million connections. If the maximum connections are not setup on the real servers, the ACE does not report a higher load. This might lead to the servers being over loaded. Refer to the least load calculation procedures described above.

Hashed Configuration

Hashing is used for site stickiness based on the source address and/or destination domain. It should be noted that the source address used here is that of the local DNS or DNS proxy for the client. Although either one or both the options can be used for hashing, Cisco recommends that both the source address and the destination domain be used for site stickiness.

Now, based on the hash value, the client is handed an A record with the IP address of a specific VIP. As long as the VIP is alive and well, time and again, the client’s request to the same destination domain takes the client to the same VIP. If the VIP goes down or is not reachable, the client is directed to a different VIP.

The configuration steps are similar to what is described in the previous section with a few changes when it comes to selecting the balance method. To choose hashed balance method, select “Open Rule Builder” and enter all the information. Then choose Hashed and click on the by domain name and by source address boxes.

Proximity Configuration

Of the two proximity solutions supported by GSS, only static proximity configurations are provided in this document. Static proximity involves identifying the source and directing the client to a specific VIP or a group of VIPs based on a balance method. So, it is really a combination of source list and balance method. There is a possibility of static proximity for each one of the balance method.

As far as the steps themselves are concerned, they are exactly the same as the ones described earlier. Step 6. is not optional for proximity. The source address has to be configured to classify the clients into different groups. Different groups of clients can adopt different balance methods. For static proximity, using ordered list is preferred. By using ordered list, requests from a specific client for a specific sub-domain, can be directed towards a specific VIP as long as it is active. If the VIP goes down, clients can be directed to the second active VIP in the ordered list and so on.

More than one such rule can be configured with the same VIPs as members of the answer groups. The source address list and the order for the new rules can be different.

Conclusion

Unlike other global server load balancing devices, the GSS provides all the features in the same chassis. GSS also provides the decoupling between server load balancer and the global load balancing devices. GSS interoperates well with the ACE. GSS also provides most of the features and is easier to configure. GSS does support application high availability, load distribution, and business resilience.

Chapter 5—Site-to-Site Recovery Using IGP and BGP

Your organization can achieve substantial productivity gains, cost savings, and increases in business revenue by deploying a data center that is available 24X7. A highly available data center is important when deploying web-enabled applications that are available across different time zones or for supporting mission-critical applications. You can achieve redundancy and high availability for applications by deploying multiple data centers and distributing applications across those data centers. This document focuses on the design and deployment of distributed data centers for disaster recovery and business continuance.

This chapter provides design recommendations for achieving disaster recovery and business continuance by using distributed data centers. These recommendations are based on Cisco best practices and include components of Layer 2 and Layer 3 protocols, server load-balancing products, site load-distribution products, caching products, and security products and services.

Overview

Data center downtime results in revenue and productivity loss for the Enterprise. You can minimize downtime and assure business continuance by deploying distributed data centers and distributing the business applications and databases. When one of the primary data centers goes out of service, the standby data center supports the mission-critical applications, and this provides business resilience. Other benefits include application scalability, high availability and an improved end user experience. This design guide explains how to route end users to secondary data centers when a catastrophic event occurs at the primary data center.

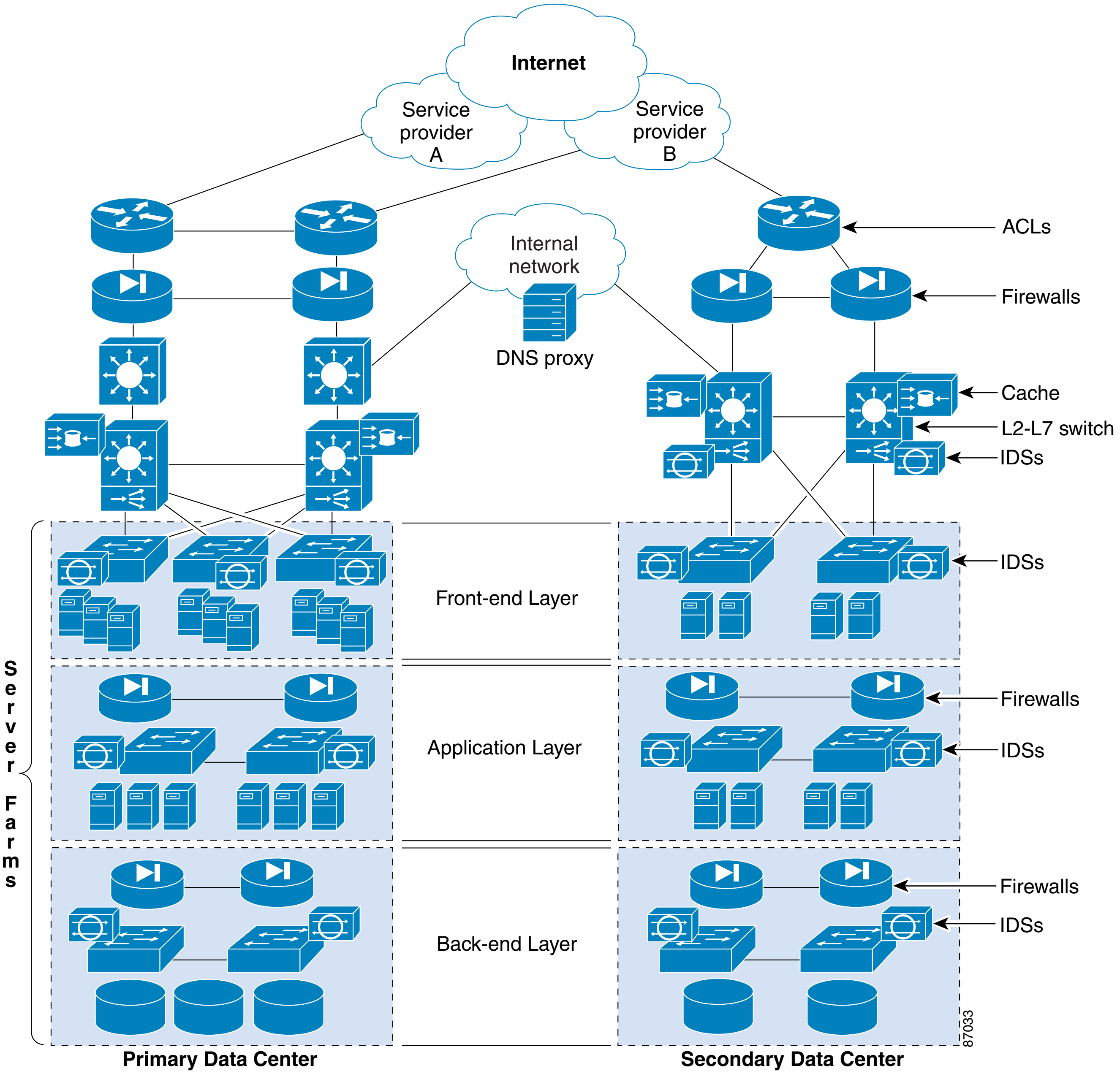
This document describes how BGP/IGP can be used to provide load sharing, disaster recovery, and business continuance solutions and identifies some possible distributed data center scenarios. No change to DNS is required for any of the solutions described in this document. All solutions are based on Route Health Injection (RHI), which injects a route based on the health of the server, and IP routing.

The architecture described in this paper uses two data centers with at least one data center connected to multiple Internet Service Providers (ISPs). In the active/standby scenario, the traffic flows to the active data center as long as the routes exist for the application servers at the active data center. Traffic is routed to the standby data center only upon failure of the application servers at the active data center.

Site-to-Site Recovery Topology

Disaster recovery solutions require at least two data centers: primary and secondary. A typical topology for a disaster recovery solution is shown in Figure 16.

Figure 16 Disaster Recovery Solution Topology



You can deploy this topology in two modes:

- Warm standby
- Hot standby

In a warm standby solution, the primary data center is the active data center receiving all client traffic. The secondary data center is the standby data center and receives no client traffic. In a warm standby solution, the applications at the secondary data center are active at all times but traffic is only processed by the secondary data center when the primary data center goes out of service.

In a hot standby solution, both sites are active. Each site is active for certain applications and acts as a standby for applications which are not active on that site. Hot Standby or Active-Active topology between data centers is possible only if the database, the back end network capacity and the application can support such an environment. Such an environment can be also be supported by IGP/BGP Site Selection mechanism as long as the same applications are not active at both sites. This is referred to as logical Active-Standby topology.

The following table summarizes the hardware and software required to implement the topology shown in Figure 16.

Product	Release	Platforms
Global Site Selector (GSS)	2.0.2.0.0 (this software version was used for testing)	GSS-4492R
Application Control Engine (ACE)	1.6.1 (this software version was used for testing)	SLB complex for Catalyst 6K platforms
Cisco 7200 Router	12.4(15)T1	VXR Router

Design Details

The data center and campus design documents provided by ESE, such as this one, specify how to design a multi-layer network and deploy application services. The multi-layer network model is fundamental to network design. providing a modular building block approach that increases ease of deployment, stability, and scalability. A modular approach is especially important for building a robust and scalable network.

In distributed data center design, the following are the three main factors for disaster recovery and business continuance.

- Design of the primary and standby data centers
- Back-end connectivity between the data centers for data replication
- Front-end intelligence for directing traffic to the active data center

This design document describes the front-end intelligence required for site-to-site recovery. There are two parts for the design which are discussed in this document.

- Route Health Injection (RHI) in the intranet for site-to-site recovery
- Extending RHI using BGP at the edge of the network to achieve site-to-site recovery Back-end connectivity between data centers is outside the scope of this paper.

Design Goals

Building a data center involves careful planning for capacity, redundancy, high availability, security and manageability. This describes the design of network failover at the data center front-end to ensure application accessibility at the standby data center.

Redundancy

Deploying redundant application servers in the server farm and using a content switch as a front-end achieves application redundancy within the data center.

You can achieve similar redundancy across multiple data centers by duplicating the data centers and creating front-end intelligence to route traffic to the active application servers. You must duplicate the three layers in the secondary data centers and replicate the state information and data across the two data centers. This ensures that your data centers are in sync at all times.

High Availability

High availability encompasses load distribution among multiple data centers as well as redundancy between data centers. High Availability addresses the elimination of congestion as well as single point of failure by deploying redundant networks and data centers. After you replicate data and deploy redundant applications at the standby data center, two tasks remain to achieve application high availability during a data center failure:

- Detect data center application failure
- Divert end users to the alternate data center(s) either by using DNS, or IGP and BGP mechanisms

Application Requirements

Successfully deploying a disaster recovery solution depends on understanding application requirements and end user connection mechanisms. End users connect to applications in the data center either by using DNS or hard coded IP addresses. Typically hard coded IP addresses are used in legacy applications. For either legacy or non-legacy applications, IP can be used as a site selection mechanism. When relying on IP for Site Selection, no changes are made to the DNS servers. Non-legacy applications tend to use DNS and are therefore supported by DNS Site Selection mechanism as well if required. Typically, different mechanisms may co-exist to support all Enterprise applications.

Security is built into data center architecture by using routers with ACLs, firewalls, and intrusion detection devices.

You should be able to increase the number of applications supported in a data center before deploying additional data centers. You must also ensure that the request routing devices have sufficient capacity. Disaster recovery fails if the request routing devices themselves become the bottleneck for end users.

Additional Design Goals

In addition to the goals mentioned, false alarms or false triggers are intolerable in a disaster recovery solution. False alarms can be caused by a device failure or a network failure. When request routing devices cannot probe the application servers, they assume that the application servers are down and trigger a failover to a standby data center.

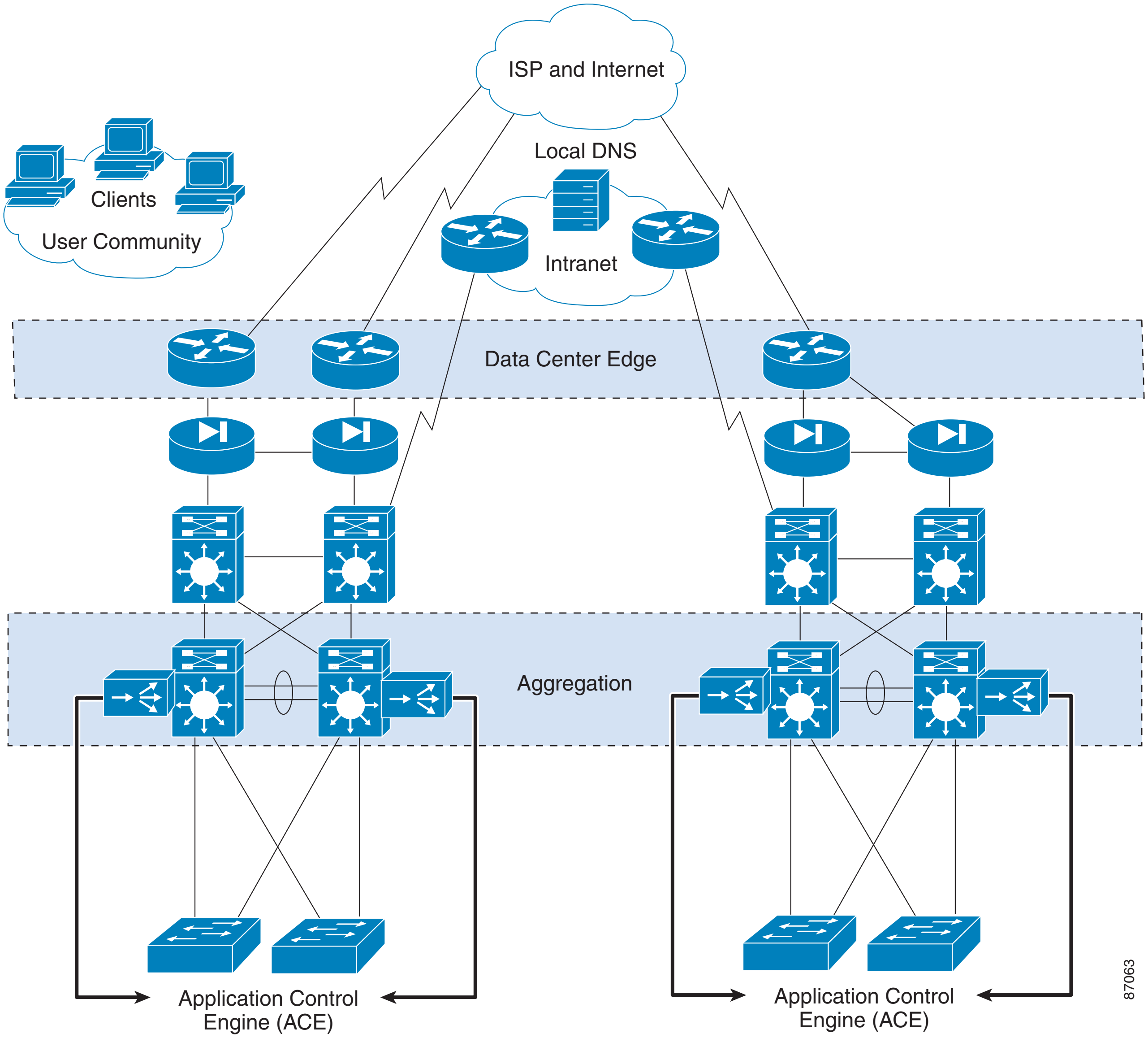
When configuring automatic failover to a secondary design center, you should be careful that the failover is not triggered by momentary problems in the primary data center network. The failover to the primary data center is typically triggered by a lack of response to a configurable number of health probes that are generated at configurable intervals. You should set the number of probes and the interval between probes in a way that ensures that failover to the secondary data center only occurs when a serious failure occurs.

Each Enterprise has its own tolerance limits for data center downtime, with financial institutions having the most stringent requirements. Other Enterprises may have tolerance limits of up to 4 hours. Enterprises with liberal tolerances may prefer manual failover to automatic failover because it gives network personnel time to inspect the situation and determine the appropriate response.

Design Recommendations

Understanding application requirements is critical because the applications used in the data center determine the topology of the solution. When there is a combination of legacy and non-legacy applications, the solution might use both DNS and IGP/BGP mechanisms. Keep the design goals in mind when deploying a disaster recovery solution and avoid false failures that trigger an unnecessary failover. Use ACE to activate route health injection (RHI), as shown in Figure 17.

Figure 17 Using ACEs with RHI



The ACEs are part of the aggregate switches, deployed in redundant mode within the data center. Unlike with the CSS, ACE uses routing to solve disaster recovery problems. ACE advertises host routes into the network from both data centers. Based on the metrics, the upstream router picks the best route.

Due to application requirements, you must use a combination of at least two technologies when both legacy and non-legacy applications are deployed in the data center. RHI provides support for disaster recovery solutions for legacy applications. Due to drawbacks in DNS caching, RHI is often preferred to DNS solution for internet facing applications. ACE and CSS support RHI but the solutions in this design guide use the ACE.

Advantages and Disadvantages of Using ACE

The topology using the ACE is recommended only to use RHI for supporting legacy applications that do not use DNS resolution to resolve the server IP addresses. The advantages of using RHI include the following:

- Quick convergence times.
- No external health probes.
- No exchange of information between sites about application availability.
- Ideal for active standby or warm standby disaster recovery solutions.

The disadvantages of using RHI include the following:

- If routes are summarized, a block of IP address is lost for each application.
- Can be used for site-to-site load distribution with certain restrictions like the data centers have to be located far apart.
- Use of Firewall Service Module (FWSM) between the ACE and MSFC prevents the use of RHI.

Site-to-Site Recovery using BGP

In an active/standby data center solution, the applications are hosted on both the data centers, however, only one of the data centers is active. All traffic goes to the active data center; traffic is routed to the standby data center only when the active data center fails.

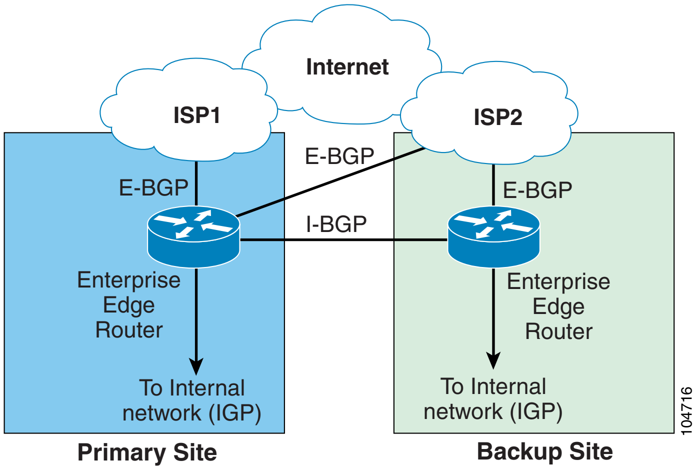
One way to implement an active/standby data center solution is by using the same IP addresses at two different locations within the Enterprise network and advertising the IP addresses with different metrics from each location. The metrics determine the path taken by traffic to and from the network clients. This may be preferable to a DNS solution because it avoids the vulnerabilities of DNS record caching. Specially for an internet facing application, this is more appealing. This section describes how you can use extend the RHI capability using BGP to direct clients to the primary site in an active/standby scenario, with no change is required to the DNS entries. The following scenarios are discussed below.

- AS Prepending
- BGP Conditional Advertisements

The first scenario is probably the simplest to implement. The second solution is more sophisticated because the secondary path is advertised conditionally if primary location goes out of service. To implement any of these solutions successfully, the cooperation of your ISP is required.

Figure 18 illustrates a simple topology for implementing site-to-site recovery solutions. The edge routers at both the primary and standby sites are connected to ISPs using E-BGP. Also, the two sites are interconnected by a link with I-BGP running between sites. The edge routers are running both IGP and BGP routing protocols. It is assumed that the rest of the internal network at both sites is running an IGP protocol and the IP prefix of interest shows up in the routers at both sites. These routes must be redistributed into BGP so that clients trying to get to this IP address are routed to the primary site as long as the IP address (application) is active there.

Figure 18 Multi-Site Topology for Site-to-site Recovery



AS Prepending

The BGP best path algorithm follows a series of steps to determine the best path to a specific destination. One of the steps involves BGP looking at the AS path to select the best route. AS prepending causes an AS to be added to an advertised route. The lower the number of the AS in the path list, the better the route is considered. For a disaster recovery solution using an active and standby site, the routes can be prepended with the same AS from the secondary site. When BGP goes through best path selection, the primary site will be chosen because it was advertised normally without a prepended AS.

AS prepending is suited for Enterprises who own an entire block of IP addresses. For other Enterprise customers, advertising the same IP with AS prepended from a secondary site might attract unwanted traffic.

The other caveat is that Cisco routers allow path selection based on the AS path to be turned off, even though the RFC mandates its use for path selection. The following section describes how to use BGP attributes as an alternative to AS prepending.

BGP Conditional Advertisements

Cisco's implementation of BGP allows conditional route advertisement, which can be used for site-to-site recovery. With this method, a certain condition must be met before advertisement occurs. A router at the secondary site monitors a list of prefixes from the primary site, and if the prefixes are missing from the BGP table, **then** it advertises a set of specified prefixes. There is no ambiguity because only the secondary site advertises routes. The secondary site learns the list of prefixes from I-BGP between sites.

To make this work, you configure a conditional advertisement on both the primary and secondary sites. The conditional advertisement at the primary site facilitates the conditional advertisement at the secondary site. If the routes are simply redistributed into BGP from IGP and advertised to the I-BGP peer, the secondary site will also advertise the route and this defeats the purpose of conditional advertisement. For this reason, the router at the primary site advertises the prefix to its I-BGP peer with the community set to “no-export.” This setting prevents the secondary site from advertising the route to its E-BGP peer. Also, the prefix is found in the BGP table so the condition required for advertisement is not met.

If both ISPs fail at the primary site, the conditional advertisement at the primary site router stops the I-BGP advertisements. This triggers the conditional advertisement at the secondary site router, which then advertises a more specific prefix. To implement this solution, make sure your ISP allows the advertisement of a block of IP addresses obtained from a different service provider.

Design Limitations

Both solutions advertise the same IP address from a secondary site based on certain criteria. If you do not own a block of IP addresses, your service provider must be willing to accept a route that belongs to a different ISP because the same IP address is being advertised from both sites. If you do own a block of IP addresses, the implementation is straightforward.

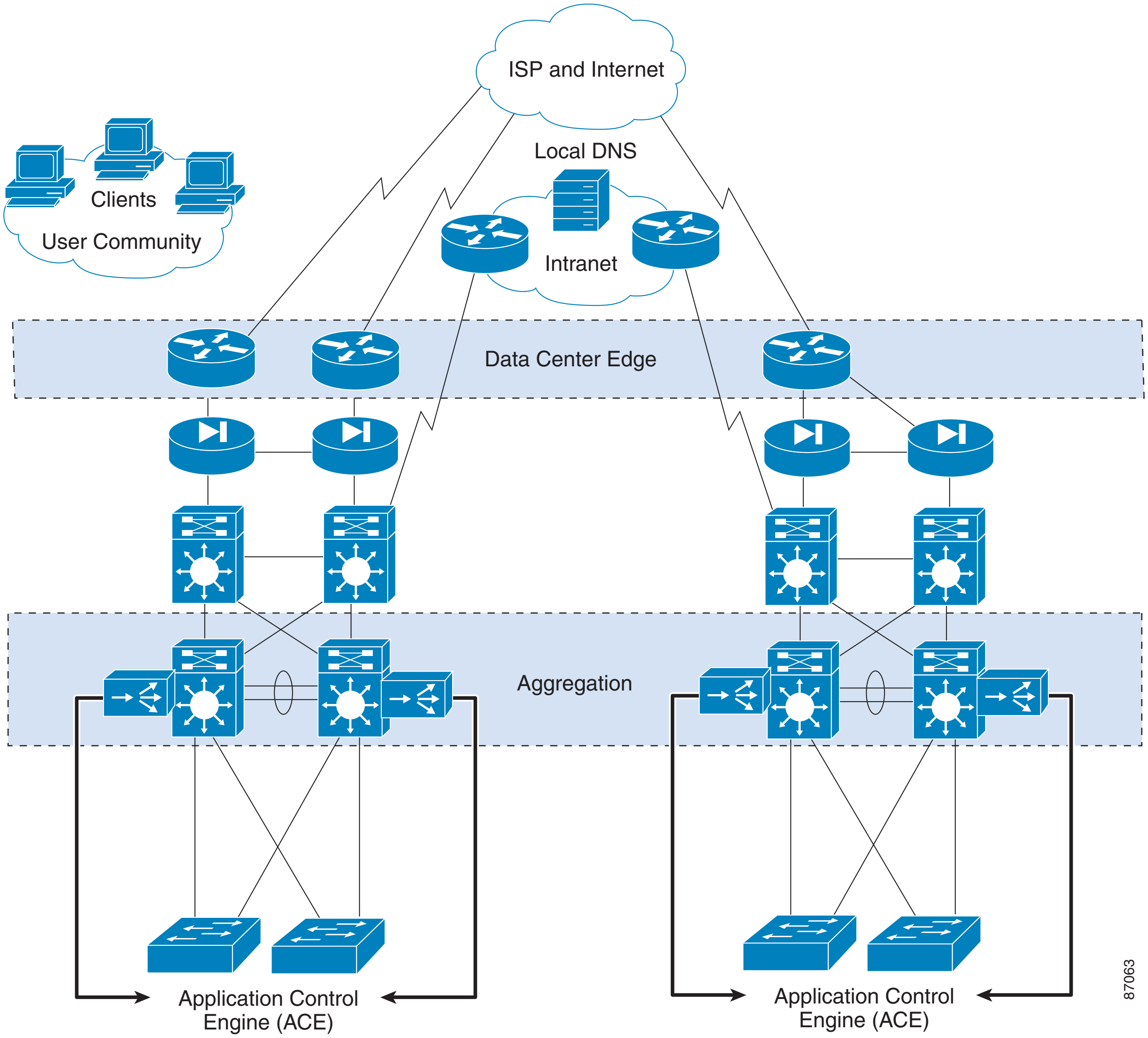
Exercise caution when using either of these solutions if the link connecting the primary and standby data centers has limited bandwidth, such as a serial link. Failure of both ISP links from the primary site will cause traffic to be directed to the secondary site through the ISP. The traffic then has to flow over a low-bandwidth link from the secondary site to the primary site, unless the active/standby roles are reversed. However, reversing the roles of the sites may not be desirable if the primary data center is just temporarily disconnected from the Internet.

Recovery Implementation Details Using RHI

Disaster recovery must provide intelligent IP address resolution for applications. The address resolution process provides information about the closest site that has the requested information. This process is called site selection. This section provides implementation details for site-to-site recovery using IGP/BGP.

Use this topology to provide disaster recovery solutions for legacy applications or where DNS changes are not desirable. A complete solution for an Enterprise that has both legacy and non-legacy applications can only be provided by using Site Selector and Route Health Injection. The RHI solution is depicted in Figure 19.

Figure 19 RHI using ACE



Configure RHI on the ACEs. As stated earlier, this topology is the only available option that provides redundancy and high availability in environments containing legacy applications that are accessed using static IP addresses.

Note ACE is deployed on service switches connected to the aggregation switches. For Enterprise data centers, ACE is deployed directly on the aggregation switches, subject to the availability of these switches and your willingness to use native images on aggregation switches.

In this method, two different data centers use the same VIP address. The VIP addresses are advertised as host routes from both data centers. The upstream router picks the best route and routes clients to the destination data center. The downside of this method is that routes cannot be summarized because of reliance on host routes. However, this method requires no other changes and converges quickly.

Under normal conditions, the end users or clients are request routed to the best route listed in the routing tables of the routers. When a catastrophic failure occurs, IP routing takes care of updating the routing table with the alternate route. In this topology, end user sessions time out during a catastrophic failure at the active data center. The clients must restart the application to get connected to the alternate data center.

High Availability

ACEs can be deployed in the data center in a redundant configuration, and this works similar to HSRP with one active ACE and one standby ACE. Refer to *Scaling Server Farms Design Document* for more details about configuring redundant ACEs. When primary and backup data centers are deployed, high availability is achieved as a result of routing changes.

Configuration Examples

There are three modes of operation for the content switch in a data center and RHI works in all three modes.

Step 1 Configure a working VIP by configuring the client side VLAN, server side VLAN, VLAN database, server farm and virtual server. Refer to *Scaling Server Farms Design Document* for more details about this configuration.

Step 2 Insert the VIP as a static route into the MSFC routing table by advertising the active VIP to the MSFC.

Step 3 Redistribute routes into OSPF.

Step 4 Make necessary routing changes by tuning the route metrics injected in Step 2.

Step 5 Repeat Steps 1-3 (at least) at the secondary data center.

Step 1 allows you to bring the VIP address online. In addition to this, tune the keepalive frequency for the real servers using instructions provided in the document.

Once the VIP address is online, the route can be injected into the MSFC routing table using Step 2. Perform Step 4 at one of the data centers, not both. For the sake of completeness, both steps are shown below.

Configuring the VLAN Interface Connected to the Core Routers

```
Router(config)# interface vlan 100
Router(config-if)# ip address 20.18.31.2 255.255.255.0
Router(config-if)# no ip redirects
Router(config-if)# no ip unreachables
Router(config-if)# no ip proxy-arp
```

Note If you have a fault tolerant configuration, you must configure **no ip proxy-arp** on the interface *before* you perform any subsequent steps. Turning off proxy ARP prevents the server ARPing for the VIP address from receiving a response from the secondary ACE before it receives a response from the primary ACE.

Configuring the Server Farm

```
rserver host test
ip address 20.40.30.10
probe REAL_Servers
inservice

serverfarm host REAL_Servers
rserver test
inservice

Class-map VIP_200
2 match virtual-address 20.40.30.10 any
Configure the client-side VLAN as follows:

Interface vlan 100  ip address 20.18.31.150 255.255.255.0

alias 20.18.31.6 255.255.255.0
```

The upstream VIP address on the aggregate switch is 20.18.31.2. This is the interface through which the client connections are established. It is important to use the alias when there is a redundant ACE. With RHI, use the alias even if there is no redundant ACEs (this is a workaround for caveat CSCdz28212).

Configuring the Server-Side VLAN

```
Interface vlan 30
Ip address 20.40.30.1 255.255.255.0
```

The default gateway address on the real servers is 20.40.30.1.

Configuring the Virtual Server

```
serverfarm host REAL_Servers
rserver test
inservice

Class-map VIP_200
2 match virtual-address 20.17.30.201 any

Policy-map type  loadbalance  http first-match  real.pol
Class class-default
Serverfarm REAL_Servers
```

Injecting the Route into the MSFC Routing Table

Use the advertise active command to notify the MSFC that there is a VIP address available over the client VLAN. The MSFC then injects this static route into the routing table.

```
Policy-map multi-match test
Class  VIP_200
Loadbalance vip inservice
Loadbalance vip advertise active
Loadbalance policy real.pol


Interface vlan 100
Service-policy input rhi-configuration
Ip route inject vlan 100
```

Redistributing Routes into OSPF

```
Router(config)# router ospf 1
Router(config-router)# redistribute static subnets
```

Changing Route Metrics

This step is needed only at one of the data centers. It changes the metric so that the upstream routers carry only the best routes. A metric of 10 is used in this example.

```
Router(config)# router ospf 1
Router(config-router)# redistribute static metric 10 subnets
```

The following example shows the routing table after the completing these configuration steps. The route is injected into the routing table as a static route. After you complete the configuration at the primary data center, the static route is redistributed into OSPF with the configured metric.

```
Router# sh ip route

Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     20.0.0.0/8 is variably subnetted, 17 subnets, 3 masks
O 20.19.0.0/16 is a summary, 1d19h, Null0
O 20.18.30.0/24 [110/2] via 20.18.31.1, 2d08h, Vlan100
C 20.18.31.0/24 is directly connected, Vlan100
O IA 20.17.31.0/24 [110/15] via 20.18.31.1, 2d08h, Vlan100
O IA 20.17.30.0/24 [110/14] via 20.18.31.1, 2d08h, Vlan100
C 20.40.30.0/24 is directly connected, Vlan30
O 20.18.100.0/24 [110/3] via 20.18.31.1, 2d08h, Vlan100
O 20.18.99.0/24 [110/3] via 20.18.31.1, 2d08h, Vlan100
O IA 20.17.99.0/24 [110/15] via 20.18.31.1, 2d08h, Vlan100
S 20.19.30.200/32 [1/0] via 20.18.31.6, Vlan100
```

These static routes are redistributed. The following example shows the routing table in one of the upstream routers.

```
Router# sh ip route

Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is 20.17.40.2 to network 0.0.0.0
```

```
     20.0.0.0/8 is variably subnetted, 16 subnets, 2 masks
O E2 20.19.0.0/16 [110/10] via 20.17.50.2, 00:13:23, Serial1/0
O IA 20.18.30.0/24 [110/10001] via 20.17.50.2, 00:13:23, Serial1/0
O IA 20.18.31.0/24 [110/10001] via 20.17.50.2, 00:13:23, Serial1/0
O IA 20.17.31.0/24 [110/12] via 20.17.40.2, 00:13:23, BVI10
O IA 20.17.30.0/24 [110/11] via 20.17.40.2, 00:13:23, BVI10
O IA 20.40.30.0/24 [110/10002] via 20.17.50.2, 00:13:23, Serial1/0
O IA 20.17.33.0/24 [110/11] via 20.17.40.2, 00:13:24, BVI10
O 20.17.35.0/24 [110/11] via 20.17.40.2, 00:13:24, BVI10
C 20.17.41.0/24 is directly connected, BVI20
```

Repeat these steps at the standby or secondary data center.

Routing Advertisements in RHI

RHI works by injecting a host route into the routing table based on the availability of the VIP address. Typically, the VIP addresses in a data center are in the same subnet as the client VLAN. This helps avoid configuring static routes on the aggregation switch and advertising static routes. But with RHI, the secondary data center has to have a VIP address in a different subnet, which is the same as for the primary data center.

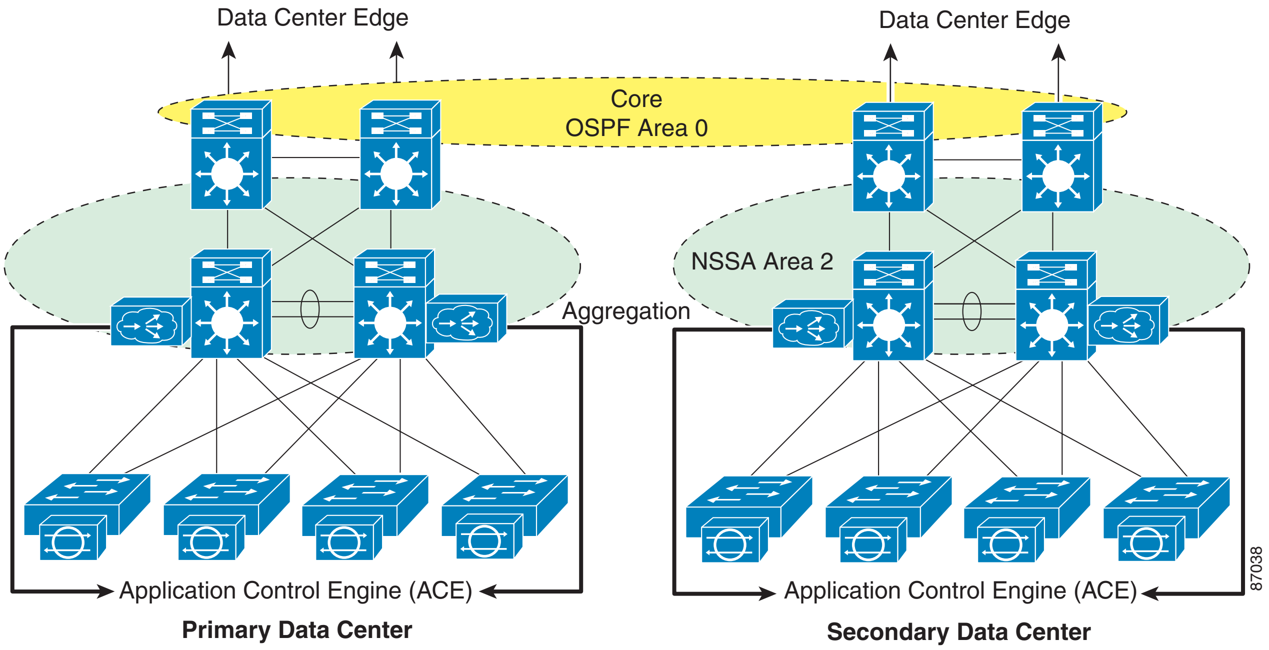
The VIP address has to be advertised with a metric other than the one used for the primary data center VIP address. Notice the highlighted routes in the routing table shown above.

The infrastructure design document provides several recommendations about how the data center area should be configured. If you use RHI, the route has to advertised from the data center. If you use RHI, configure the data centers as a not-so-stubby-area (NSSA) and configure the data center in area X, where X could be 0 or anything else.

Case 1

Use a different subnet for all the VIP addresses in a data center other than the client VLAN subnet. For example, all the network addresses for Layer 3 interfaces might start with 20.17.x.x in the primary data center and with 20.18.x.x in the secondary data center. Then, use a network starting with 20.19.x.x for all VIP addresses.

Figure 20 RHI using ACE



The advantage of doing this is that it allows you to configure summarization on the aggregate switch for the VIP addresses. Also, configure the data center aggregation switches in an NSSA, which helps if you are summarizing these routes. Even if the routes are summarized, in case of a failure of any single application, all the applications have to be failed over to the secondary data center. The other alternative is to put each application in a different subnet and summarize the routes so the failure of one VIP address will not affect the other VIP addresses.

Case 2

Put both data centers in a single OSPF area (area 0) and use host routes. For example, all the network addresses for Layer 3 interfaces might start with 20.17.x.x in the primary data center and with 20.18.x.x in the secondary data center. Then, use a network starting with 20.17.x.x for all VIP addresses. This scenario is simple to configure and works well if host routes are advertised. However, it looses the flexibility of Case 1.

Case 1 is recommended if you use RHI, because RHI supports route summarization when connecting to multiple service providers.

Restrictions and Limitations

The following caveats apply for RHI regardless of whether you use Case 1 or Case 2.

- For intranet applications, host routes can be advertised and the failover from active site to standby site within few seconds. However, the applications in the data center should be capable of supporting such quick failover if such quick failover is desired. However for internet facing applications, summarization will lead to different solutions. These different scenarios are outlined in the BGP section below.
- Scalability may become an issue. If the number of applications supported grows, the number of host routes in the Enterprise network grow linearly. Summarization also creates scalability problems due to loss of different subnets.

- If a firewall service module (FWSM) is used in between the ACE and the MSFC, and the ACE and MSFC are not on the same VLAN, RHI breaks. However, this limitation will be removed by future releases of FWSM that can be deployed transparently.

Recovery Implementation Details using BGP

To use BGP for site-to-site recovery, inject a virtual IP (VIP) as a host route into the routing table using route health injection (RHI)¨ Typically, the data center network is set up as a stub network or an not-so-stubby are (NSSA). Figure 5-6 illustrates how a route is injected from the data center, advertised into the Enterprise core network and redistributed at the edge router. Solutions differ based on how the routes are advertised from the edge to the ISP. Figure 5-6 also provides an illustration of the routing domains and how the test topology was set up.

The configuration for all the solutions in this section use the following procedure.

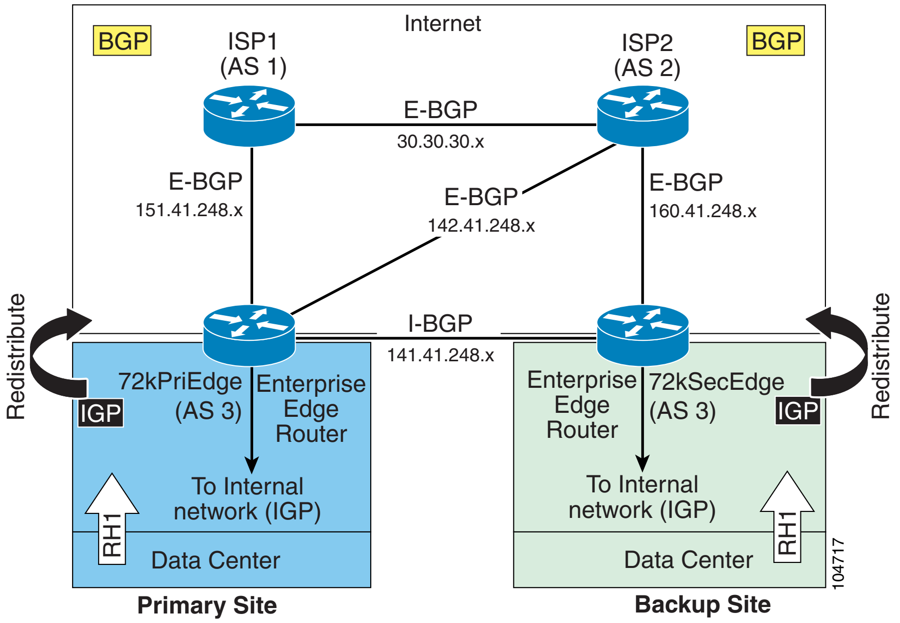
Step 1 Redistribute routes into BGP.

Step 2 Configure filters to selectively advertise routes to BGP peers as necessary.

Step 3 Perform configuration required for the specific solution.

Note that in all the topologies, the primary site is multi-homed to two ISPs, which requires using MED to send a lower metric with the routing updates to ISP1, and to send a higher metric to ISP2 from the primary site edge router. In addition, you must set weights to prefer a specific ISP as the next-hop router for outgoing traffic.

Figure 21 ***Routing Domains for Site-to-site Recovery***



AS Prepending

To implement this solution, complete the following steps:

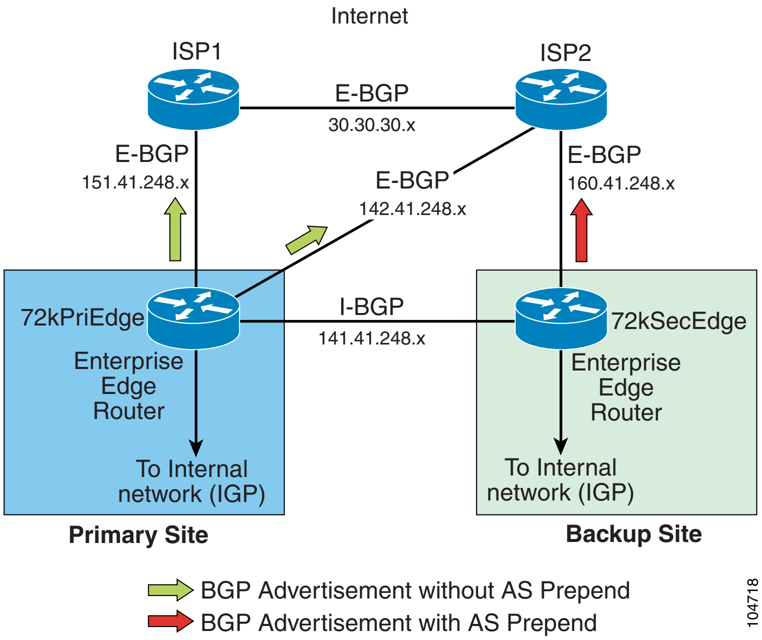
Step 1 Redistribute IGP routes into BGP at both sites.

Step 2 Configure route maps or distribute lists to filter OSPF routes if necessary.

Step 3 Configuring route maps to perform AS prepending at the second site.

Figure 22 depicts the topology for AS prepending, which assumes that the routes of interest are injected into IGP using RHI.

Figure 22 AS Prepending



Primary Site Configuration

The two ISP neighbors are in AS 1 and AS 2, and the IP addresses are 151.41.248.129 and 142.41.248.130 respectively. The primary site is in AS 3 and has an I-BGP connection to the router in the standby site. This is shown in the following configuration (the remote AS is configured as 3).

```
router bgp 3

  bgp log-neighbor-changes

  redistribute ospf 1 route-map OspfRouteFilter

  neighbor 141.41.248.130 remote-as 3

  neighbor 142.41.248.130 remote-as 2

  neighbor 151.41.248.129 remote-as 1

  no auto-summary

!
```

The OSPF routes are redistributed into BGP using the **redistribute** command. A route map is also configured which is used to selectively advertise the routes into BGP. There are different ways that you can do this. This implementation uses the **prefix-list** command. The following shows the route maps and the prefix lists:

```
ip prefix-list OspfRoute seq 10 permit 130.34.0.0/16 le 32
ip prefix-list OspfRoute seq 15 permit 20.20.0.0/16 le 32
route-map OspfRouteFilter permit 10

 match ip address prefix-list OspfRoute
!
```

The match statement in the route map matches all the IP addresses in the prefix list and selectively redistributes OSPF routes into BGP. The ip prefix-list command is configured as shown in the example above. The first address 130.34.0.0 is shown for demonstration purposes only. Note that the prefix of interest is 20.20.0.0. This is the prefix that will be advertised from both the primary site and the secondary site with different AS paths. There is no need for AS prepending because this is the primary site.

Standby Site Configuration

The standby site configuration is similar except for the addition of AS prepending.

```
router bgp 3
  no synchronization
  bgp log-neighbor-changes
  redistribute ospf 1 route-map OspfRoutes
  neighbor 141.41.248.129 remote-as 3
  neighbor 160.41.248.130 remote-as 2
  neighbor 160.41.248.130 route-map AddASnumbers out
  no auto-summary

!
…
ip prefix-list DR-Applications seq 10 permit 140.36.0.0/16 le 32
ip prefix-list DR-Applications seq 15 permit 140.40.0.0/16 le 32
ip prefix-list DR-Applications seq 20 permit 20.20.0.0/16 le 32
!
…
route-map OspfRoutes permit 10

 match ip address prefix-list DR-Applications
!
```

The configuration for redistribution is similar to the configuration for the primary site. Note that the prefix of interest is 20.20.0.0. As shown in the configuration above, an additional route map (AddASnumbers) is configured for all outbound advertisements. This route map is used when advertising routes to the neighbor 160.41.248.130. This route map is shown below.

```
route-map AddASnumbers permit 10
 match ip address prefix-list DR-Applications
 set as-path prepend 3 3 3

!
```

Notice that the match command matches the prefix list, and the set command prepends the AS. When the route is advertised to the ISP, it shows up with multiple paths. Following is the output from the show ip bgp command on the second ISP router. Notice the prefix of interest.

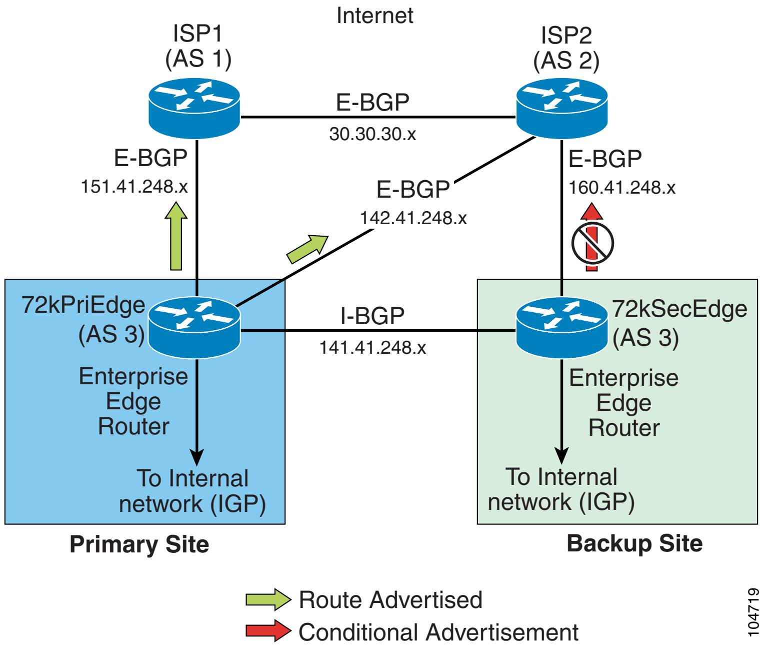
```
72k-ISP2# sh ip bgp
BGP table version is 47, local router ID is 160.41.248.130
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,

 r RIB-failure
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network  Next Hop       Metric LocPrf Weight Path
*> 10.10.0.4/30  160.41.248.129 0 0 3 ?

* 20.20.20.0/24  160.41.248.129 20 0 3 3 3 3 ?

* 30.30.30.129 0 1 3 ?
*> 142.41.248.129 20 0 3 ?

* 130.34.248.128/26


 30.30.30.129 0 1 3 ?
*> 160.41.248.129 0 3 ?
…
```

Three alternate paths show up as a result of dual-homing on the primary site. The prefix advertised from the secondary site has a longer AS path. The best path is indicated by “>” next to the asterisk on the third line and this is the prefix advertised by the primary site without AS prepending.

BGP Conditional Advertisement

Figure 23 illustrates conditional advertisement from two different sites.

Figure 23 BGP Conditional Advertisement



There are two different ways to use BGP conditional advertisement:

- Using the **advertise-map** command.

- Using the **aggregate-address** and **suppress-map** commands.

Both of these methods conditionally advertise routes based on the availability of routes in the BGP table. The implementation details for the second method are not discussed in this document. The difference between the two methods is that the two IP addresses are the same when using the **advertise-map** command while they are different when using the **aggregate-address** and **suppress-map** commands.

To implement BGP conditional advertisement using the **advertise-map** command, complete the following steps.

Step 1 Redistribute IGP routes into BGP at both sites.

Step 2 Conditionally advertise the prefix of interest from the primary site into I-BGP.

Step 3 Advertise the prefix of interest with a longer mask (more specific route) to the E-BGP peer of the secondary site router if the prefix of interest is not in the BGP table of the secondary site router.

Note When the topology changes, it takes about 80 seconds for the BGP tables to become stable with default timers. CSCdu37363 is the bug ID filed to fix the I-BGP quick failover problem.

Primary Site Configuration

This section provides the BGP configuration for the primary site.

It is recommended to use redundant links for I-BGP connection. The details are not discussed in this document because this is a well understood concept. Also, this disaster recovery solutions in this document use interface IP addresses rather than loopback IP addresses for E-BGP and I-BGP peering.

```
router bgp 3
 no synchronization
 bgp log-neighbor-changes
 network 142.41.248.128 mask 255.255.255.192
 network 151.41.248.128 mask 255.255.255.192
 redistribute ospf 1 route-map OspfRouteFilter
 neighbor 141.41.248.130 remote-as 3
 neighbor 141.41.248.130 next-hop-self
 neighbor 141.41.248.130 send-community
 neighbor 141.41.248.130 advertise-map ADV exist-map NON
 neighbor 142.41.248.130 remote-as 2
 neighbor 142.41.248.130 distribute-list 10 out
 neighbor 151.41.248.129 remote-as 1
 neighbor 151.41.248.129 distribute-list 10 out
 no auto-summary

!
```

Note When RHI is used, routes show up as E 2 routes in the routing table. With BGP, the specific type of route has to be identified when redistributing routes. However, if prefix lists or distribute lists are used in the configuration, the route type does not matter.

AS prepending was illustrated using the prefix-list command. For this configuration, the distribute-list command is used for each neighbor to prevent unwanted routes from being advertised. The neighbor 141.41.248.130 send-community command advertises the prefix with the community. This community is used at the standby site router to make decisions about advertising the prefix. Because you do not want to advertise this route to the E-BGP peer of the secondary router, set the community to no-export using the following commands:

```
route-map ADV permit 10
 match ip address prefix-list Adv
 set community no-export

!
```

The second route map is as shown below:

```
route-map NON permit 10
 match ip address prefix-list Non
!
```

The prefix-lists, Non and Adv, are as shown below:

```
ip prefix-list Adv seq 5 permit 20.20.0.0/16
!
ip prefix-list Non seq 10 permit 142.41.248.128/26
ip prefix-list Non seq 15 permit 151.41.248.128/26
!
```

The prefix of interest here is 20.20.0.0/16.

The distribute list refers to an access-list, which has a list of all the prefixes that have to be filtered.

```
access-list 10 deny  3.3.3.0
access-list 10 permit 20.20.20.0

access-list 10 permit 130.34.0.0
access-list 10 permit 142.41.0.0
access-list 10 permit 151.41.0.0
access-list 10 permit 130.34.0.0 0.0.255.255
```

Standby Site Configuration

You must complete similar configuration at the standby site. You must also configure conditional advertisements on the standby site router. The configuration at the standby site router is as follows:

```
router bgp 3
 no synchronization
 bgp log-neighbor-changes
 redistribute ospf 1 route-map OspfRoutes
 neighbor 141.41.248.129 remote-as 3
 neighbor 141.41.248.129 next-hop-self
 neighbor 141.41.248.129 distribute-list 11 out
 neighbor 160.41.248.130 remote-as 2
 neighbor 160.41.248.130 distribute-list 2 out
 neighbor 160.41.248.130 advertise-map ADV non-exist-map NON
 no auto-summary

!
```

Remember that route maps, distribute lists, and prefix lists can be used at the secondary site to control redistribution and peer advertisements. The conditional advertisement is provided by the following command:

```
neighbor 160.41.248.130 advertise-map ADV non-exist-map NON
```

This command advertises the prefix specified in ADV, if the prefix is missing from the NON route map. The route map configuration is as follows:

```
route-map NON permit 10

 match ip address prefix-list Non
!
route-map ADV permit 10

 match ip address prefix-list Adv
!
```

The prefix list configuration is as follows:

```
ip prefix-list Adv seq 5 permit 20.20.20.0/24
!
ip prefix-list Non seq 10 permit 20.20.0.0/16
!
```

The prefix in the list, Non, represents the advertisement from the primary. If this prefix is missing from the BGP table, the prefix specified in Adv is advertised. The prefix in Adv is more specific than the prefix in Non. The configuration for redistribution of OSPF routes into BGP is as follows:

```
ip prefix-list DR-Applications seq 10 permit 140.36.0.0/16 le 32
ip prefix-list DR-Applications seq 15 permit 140.40.0.0/16 le 32
ip prefix-list DR-Applications seq 20 permit 20.20.0.0/16 le 32
ip prefix-list DR-Applications seq 25 deny 10.10.0.0/16 le 32
!
route-map OspfRoutes permit 10

 match ip address prefix-list DR-Applications
!

Make sure that you modify the OSPF weight at the standby site when redistributing routes
into BGP so it will not take precedence over a route learned from I-BGP peer. The required
commands are as follows: route-map OspfRoutes permit 10
 match ip address prefix-list DR-Applications
 set weight 0

!
```

The BGP tables on the ISP routers are shown below:

```
72k-ISP2# sh ip bgp

BGP table version is 140, local router ID is 160.41.248.130

Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
     r RIB-failure, S Stale

Origin codes: i - IGP, e - EGP, ? - incomplete

   Network  Next Hop       Metric LocPrf Weight Path

* 20.20.0.0/16  30.30.30.129 0 1 3 ?
*> 142.41.248.129 20 0 3 ?

* 130.34.248.128/26


 30.30.30.129 0 1 3 ?
*> 142.41.248.129 12 0 3 ?
*> 142.41.248.128/26

 0.0.0.0 0 32768 i
*> 160.41.248.128/26
```

```
 0.0.0.0 0 32768 i
72k-ISP2#
```

Note Note that the standby site does not advertise the prefix of interest as long as the ISP links to the primary site are up. BGP points to the primary site as the best path. The other path that was learned on this ISP router was over the link between the two ISPs.

When one ISP at the primary site goes down, the conditional advertisement is not triggered at the standby site. The following shows the BGP table at the second ISP, when one of the ISPs went down at the primary site:

```
72k-ISP2#sh ip bgp
BGP table version is 140, local router ID is 160.41.248.130
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,

     r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete


     Network   Next Hop                      Metric LocPrf Weight Path
*> 20.20.0.0/16  142.41.248.129              20     0 3 ?
*> 130.34.248.128/26
                         142.41.248.129      12     0 3 ?
*> 142.41.248.128/26
                0.0.0.0                      0      32768 i
*> 160.41.248.128/26
                0.0.0.0                      0      32768 i
```

The following shows the BGP table when both ISP links to the primary site go down:

```
72k-ISP2#sh ip bgp

BGP table version is 102, local router ID is 160.41.248.130

Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
     r RIB-failure, S Stale

Origin codes: i - IGP, e - EGP, ? - incomplete

   Network  Next Hop       Metric LocPrf Weight Path

*> 20.20.20.0/24  160.41.248.129 20 0 3 ?

*> 160.41.248.128/26

 0.0.0. 0 32768 i
```

Restrictions and Limitations

As discussed earlier, there is no problem distinguishing different applications with host routes. Host routes work well for intranet applications. However, for Internet-facing applications, routes will be summarized at the edge. Route summarization leads into the problem of distinguishing different applications. With route summarization, there is possibility of black holing traffic for the application that just failed. With conditional advertisements, failure of an application in a subnet causes the failover of all the applications. This might be acceptable based on how server redundancy is provided in each data center and other issues and disaster recovery policies.

If the applications should recover at the speed of IGP convergence, the first possible solution for such a scenario is as follows:

- Use high bandwidth link between sites
- Use host routes in the intranet
- Run IGP between sites

You can use this solution along with BGP conditional advertisements or AS prepending. With this solution, regardless of how the traffic comes in, either through the primary site or the standby site, due to host routes, the traffic will be routed to the appropriate data center using the link between data centers.

The failover from active to standby data center is closely associated with the applications and database running in the data center. Failover to standby data center can be achieved in seconds using IGP/BGP. The applications in the standby data centers should also support such a quick failover. More often, there are business policies put in place for identifying disaster situations and that will consume more time. More over, due to high availability design within each data center unless there is a disaster, the application will still be available.

The second solution is to put the applications in different subnets so that failure of a specific application will result in routing the traffic to a standby site. This solution will involve the following steps.

- Put the applications in different subnets
- Use conditional BGP advertisements
- No need for high bandwidth link between data centers

With this solution, the convergence speed depends on E-BGP convergence which is about 2 minutes. However, based on the number of applications and the summarization, scalability becomes an issue since more address space is used for this solution. Also, this is closely tied to availability of applications at the standby data centers which is dependent on availability of data.

Conclusion

All steps must be taken to protect and preserve data. Planning for disaster recovery and business continuance is the key to minimizing losses and increasing productivity.

Catastrophic failures can happen at any point of the network. The success of disaster recovery solutions depends on good design of the Layer 2 network, the Layer 3 network, the data center. A sound Layer 2 and Layer 3 design helps to minimize network failures. Built on this foundation, good content switching practices in a data center help reduce server inefficiency and application downtime. All this makes site-to-site recovery much more efficient and eliminates the chance of failover due to network instability.

Chapter 6—Site-to-Site Load Distribution Using IGP and BGP

Distributed data centers provide business continuance, disaster recovery, and load-sharing solutions between multiple data centers. Organizations have different requirements that must be met through a business continuance and disaster recovery solution. Some organizations use DNS for managing business continuance and load sharing between two data centers. However, for some organizations that do not want to make changes to DNS can use IGP/BGP mechanisms for the solution. This design document introduces reference topologies that use IGP/BGP for achieving site-to-site load distribution between data centers.

This document is intended for engineers, engineering managers, and network operations staff who need to familiarize themselves with the principles and complexities of implementing a site-to-site recovery and load distribution solution based on IGP/BGP.

Overview

This document describes how BGP/IGP can be used to provide load sharing, disaster recovery, and business continuance solutions and identifies some possible distributed data center scenarios. No change to DNS is required for any of the solutions described in this document. All solutions are based on Route Health Injection (RHI) and IP routing. For more information about RHI, see the "Route Health Injection" section on page 6-21.

The architecture described in this paper uses two data centers with at least one data center connected to multiple Internet Service Providers (ISPs). The scenario discussed is referred to as Site-to-site load distribution (active/active data centers). The active/active scenario includes two different implementations, differing in how the applications are deployed at the two data centers:

- Some active applications are deployed at each data center to provide a logical active/standby configuration for each application.
- All active applications are deployed at both sites. BGP routes traffic to the closest data center edge and IGP routes the traffic to the closest data center. Load balancing occurs over an internal link between sites.

Design Details

The basic design goal is to be able to direct clients to appropriate data center based on the data center availability or services availability at the data center. The major design issues are:

- Redundancy
- High availability
- Scalability
- Security
- Other requirements as necessary

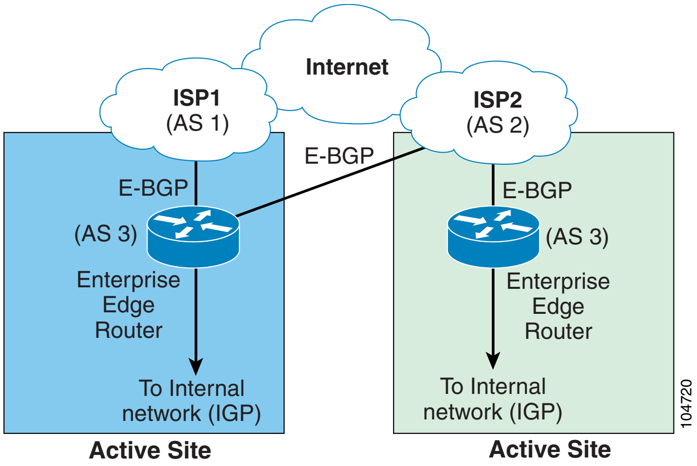
Most of these design requirements are explained in the Site Selection documents for multi site load distribution. In case of site-to-site load distribution, site selection is based on the best routes in the routing table. Site selection is not based on the configuration or the load at specific sites. All the design information for Active/Active sites is discussed in the following section.

Active/Active Site-to-Site Load Distribution

For site-to-site load balancing you must use an active/active scenario in which both data centers host active applications. Applications can be active concurrently or the applications can be hosted in a logical active/standby mode. Logical active/standby mode means that some applications will be active on one site while those same applications will be in standby mode at the second site. Different applications will be active at the second site while the same applications will be in standby mode in the first site.

Support for logical active/standby mode depends on the capabilities of specific applications and databases. If this mode is supported, you can use IGP mechanisms to route traffic to the data center that is logically active for a specific application. Figure 24 illustrates a topology for site-to-site load balancing. Notice that in Figure 24 there is no link between sites. Each site concurrently hosts applications and serves clients.

Figure 24 Topology for Site-to-Site Load Balancing



In this scenario, it is not necessary to implement AS prepending or BGP conditional advertisements. It is sufficient if the same route is advertised from two different sites. This simplifies the design as only internal IGP has to be configured based on how the applications are hosted.

The second topology for site-to-site load balancing is similar to Figure 24, except that a link is required between sites with enough bandwidth to carry the traffic between the sites.

In an active/active data center solution, the applications are hosted on both the data centers and both the data centers serve client requests. An Application Control Engine (ACE) is installed on the Catalyst 6500 in the data center. Both the data centers use the same virtual IP address (VIP) to represent the server cluster for each application, and the same subnet is advertised to the Internet with BGP. The routing infrastructure directs any client request to the topologically nearest site. This mechanism is referred to as an Anycast mechanism, and it offers three major advantages:

- **Proximity—**Clients terminate at the closest data center.
- **Site Stickiness—**Clients achieve "stickiness" and do not bounce between sites as a result of DNS or Microsoft Internet Explorer shortcomings.
- **Load Distribution—**Clients are dynamically distributed between available sites.

Note Note that the solutions discussed in this document require that the routing infrastructure be stable. Route flapping could cause long-lived application sessions (such as TCP) to break.

Implementation Details for Active/Active Scenarios

This section includes the following topics:

- OSPF Route Redistribution and Summarization
- BGP Route Redistribution and Route Preference

- Load Balancing Without IGP Between Sites
- Subnet-Based Load Balancing Using IGP Between Sites
- Application-Based Load Balancing Using IGP Between Sites
- Using NAT in Active/Active Scenarios

Each active/active scenario provides disaster recovery and load distribution, utilizing IP Anycast, BGP, IGP and RHI. Each active/active scenario requires the same two initial steps, which are described in the first two topics in this section.

Subnet-based and application-based load balancing require a link, which makes it possible to route traffic between sites within seconds if either data center fails.

In the subnet-based design, the two sites are active at the same time, and the clients that get routed to a specific site terminate their connection to the data center at that specific site. If the data center fails at that site, the traffic is routed to the second site through the internal IGP. If the entire site fails, including the edge routers, external clients can still reach the second data center through the external network. The subnet-based implementation also provides a quick way to bring down the site for maintenance.

The application-based design logically partitions the data centers into active/standby sites. If the application environment requires it, the route metrics can be changed in the data center so that a specific VIP is active at one site and is on standby at the second site. This design also helps to load balance traffic for different applications between two sites.

Network address translation (NAT) can be applied to all three designs, but has been tested for this paper with the application-based design.

OSPF Route Redistribution and Summarization

When redistributing RHI static routes into OSPF, use metric-type 1 for adding the internal cost to the external cost. This is essential in designs with IGP between sites. It is best to summarize the host routes on the MultiLayer Switch Feature Card (MSFC) in the same chassis as the ACE.

```
cat6K_l# sh run | beg router ospf

router ospf 1

 log-adjacency-changes

 summary-address 24.24.24.0 255.255.255.0

 redistribute static metric-type 1 subnets

network 10.0.0.0 0.0.0.255 area 0
 network 10.4.0.16 0.0.0.3 area 0
 network 10.4.1.0 0.0.0.255 area 0
 network 10.6.0.16 0.0.0.3 area 0
 network 130.40.248.0 0.0.0.255 area 0

!

cat6K_l# sh ip route ospf
     140.40.0.0/26 is subnetted, 1 subnets
O IA 140.40.248.128 [110/4] via 10.0.0.129, 01:32:02, Vlan10
     141.41.0.0/26 is subnetted, 1 subnets
O IA 141.41.248.128 [110/2] via 10.0.0.129, 01:32:07, Vlan10
     24.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
```

```
O 24.24.24.0/24 is a summary, 00:02:21, Null0
     130.34.0.0/26 is subnetted, 1 subnets
```

The edge router receives the default route from the ISP and propagates it down to the data center MSFC. The OSPF configuration on the edge router looks like the following:

```
72k-edgePriDC# sh run | beg router ospf

router ospf 1
 log-adjacency-changes
 network 10.0.0.0 0.0.0.255 area 0
 default-information originate

!
```

Notice that in this configuration, the **always** keyword is *not* used.

BGP Route Redistribution and Route Preference

The edge routers distribute OSPF into their BGP process using a prefix-list and each updates their neighbor with these routes (most importantly 24.24.24.0/24).

This is slightly complicated for the edge router in this example*,* 72kPriEdge*,* as it has links to both ISP1 and ISP2 to which the other edge router, 72kSecEdge, is connected. From 72kPriEdge, use MED to configure a lower metric for the route updates sent to ISP1 than for ISP2. As a result, ISP2 will always prefer 72kSecEdge for reaching 24.24.24.0/24.

You also have to set the weight for the inbound routes (0.0.0.0/0.0.0.0 route) so that ISP1 has a higher weight (2000) than ISP2. As a result, 72kPriEdge will use the router at ISP1 for its default (next-hop) router.

BGP Configuration of Primary Site Edge Router

```
72k-edgePriDC# sh run | beg router bgp

router bgp 3
 no synchronization
 bgp log-neighbor-changes
 network 142.41.248.128 mask 255.255.255.192
 network 151.41.248.128 mask 255.255.255.192
 redistribute ospf 1 route-map OspfRouteFilter
 neighbor 142.41.248.132 remote-as 2
 neighbor 142.41.248.132 route-map WEIGHT-IN in
 neighbor 142.41.248.132 route-map ISP2-OUT out
 neighbor 151.41.248.131 remote-as 1
 neighbor 151.41.248.131 route-map WEIGHT-IN in
 neighbor 151.41.248.131 route-map ISP1-OUT out
 no auto-summary

!

ip as-path access-list 2 permit ^2$
!
ip prefix-list OspfRoute seq 10 permit 130.34.0.0/16 le 32
ip prefix-list OspfRoute seq 15 permit 20.20.20.0/24
ip prefix-list OspfRoute seq 20 permit 24.0.0.0/8 le 32
access-list 10 permit 20.20.0.0 0.0.255.255
access-list 10 permit 130.34.0.0 0.0.255.255
```

```
access-list 10 permit 142.41.0.0 0.0.255.255
access-list 10 permit 151.41.0.0 0.0.255.255
access-list 10 permit 24.24.24.0 0.0.0.255
!
route-map ISP1-OUT permit 10

 match ip address 10

 set metric 20
!
route-map ISP2-OUT permit 10

 match ip address 10

 set metric 30
!
route-map WEIGHT-IN permit 10

 match as-path 2

 set weight 200
!
route-map WEIGHT-IN permit 20

 set weight 2000
!
route-map OspfRouteFilter permit 10

 match ip address prefix-list OspfRoute
!
```

BGP Configuration of Secondary Site Edge Router

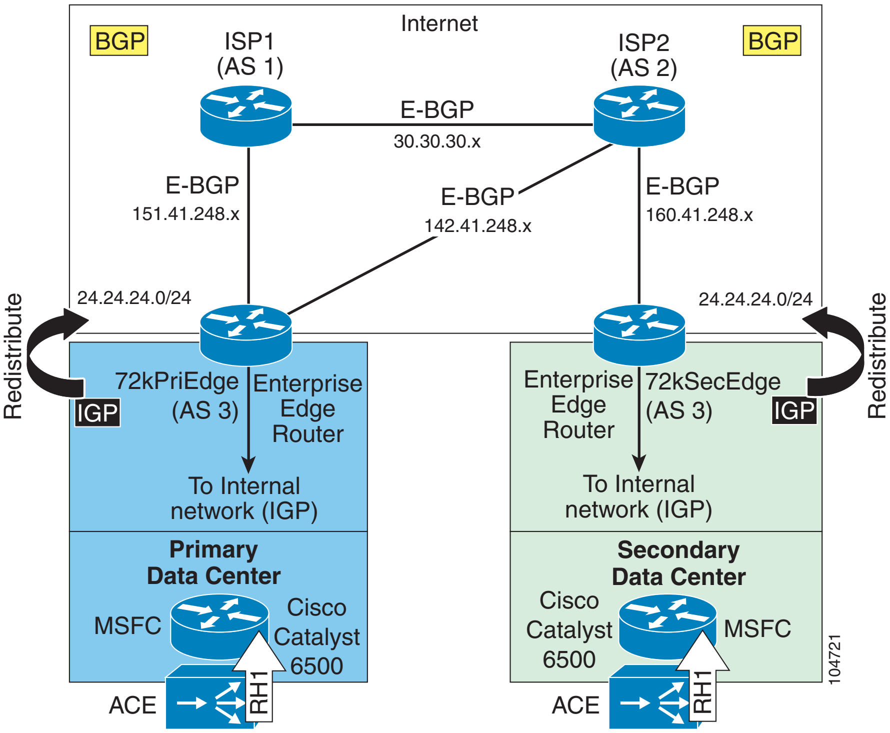
```
72k-edgeSecDC# sh run | beg router bgp

router bgp 3
 no synchronization
 bgp log-neighbor-changes
 network 160.41.248.128 mask 255.255.255.192
 redistribute ospf 1 route-map OspfRouteFilter
 neighbor 160.41.248.132 remote-as 2
 neighbor 160.41.248.132 route-map ISP2-OUT out
 no auto-summary

!
ip prefix-list OspfRoute seq 10 permit 140.40.0.0/16 le 32
ip prefix-list OspfRoute seq 15 permit 20.20.20.0/24
ip prefix-list OspfRoute seq 20 permit 24.0.0.0/8 le 32
!
access-list 10 permit 20.20.0.0 0.0.255.255
access-list 10 permit 24.24.24.0 0.0.0.255
!
!
route-map ISP2-OUT permit 10

 match ip address 10
 set metric 20
!
!
route-map OspfRouteFilter permit 10

 match ip address prefix-list OspfRoute
!

!
```

Load Balancing Without IGP Between Sites

Before using the instructions in this section to implement the load balancing solution without IGP, complete the general configuration described in OSPF Route Redistribution and Summarization, page 74 and BGP Route Redistribution and Route Preference, page 75.

Figure 25 illustrates load balancing solution without IGP, which is the simplest design discussed in this paper. In this design, the benefits of IP Anycast can be obtained without making any physical changes to the network environment.

Figure 25 Load Balancing without IGP



In this design, ACE is used in the both data centers for server load balancing. RHI is enabled so that ACE injects a host static route into the MSFC on the same Catalyst 6500 chassis. These routes are redistributed and summarized in OSPF. OSPF is redistributed into BGP on the edge routers. The link between 72kPriEdge and ISP2 is only used for redundancy. This shows how to control incoming and outgoing routes in BGP using weights and to prefer routes using MEDs.

Routes During Steady State

```
Cat6k-ISP1# sh ip bgp 24.24.24.0
BGP routing table entry for 24.24.24.0/24, version 40
Paths: (2 available, best #1)

  Advertised to non peer-group peers:

  30.30.30.132

 3


    151.41.248.129 from 151.41.248.129 (151.41.248.129)
      Origin incomplete, metric 20, localpref 100, valid, external, best
 2 3
    30.30.30.132 from 30.30.30.132 (160.41.248.132)
      Origin incomplete, localpref 100, valid, external

72k-ISP2# sh ip bgp 24.24.24.0
BGP routing table entry for 24.24.24.0/24, version 27
Paths: (3 available, best #3, table Default-IP-Routing-Table)

  Advertised to non peer-group peers:
  30.30.30.131 142.41.248.129
 1 3

    30.30.30.131 from 30.30.30.131 (151.41.248.131)
      Origin incomplete, localpref 100, valid, external
 3
    142.41.248.129 from 142.41.248.129 (151.41.248.129)
      Origin incomplete, metric 30, localpref 100, valid, external
 3
    160.41.248.130 from 160.41.248.130 (160.41.248.130)
      Origin incomplete, metric 20, localpref 100, valid, external, best
```

Routes After All Servers in Primary Site Are Down

Notice the 24.24.24.0/24 routes below that point to 151.41.248.129 on ISP1 and those that point to 142.41.248.129 on ISP2 are removed. This is because the primary site edge router stopped receiving routes from the connected MSFC. These changes were triggered by ACE removing the routes from the MSFC when it determined that the servers were down.

In our testing with minimal routes in the ISP routers, these BGP routes were removed in less then 5 seconds. Convergence would be higher in a production network.

```
Cat6k-ISP1# sh ip bgp 24.24.24.0
BGP routing table entry for 24.24.24.0/24, version 42
Paths: (1 available, best #1)
Flag: 0x820

  Advertised to non peer-group peers:
  151.41.248.129
 2 3

    30.30.30.132 from 30.30.30.132 (160.41.248.132)
      Origin incomplete, localpref 100, valid, external, best
```

```
72k-ISP2#sh ip bgp 24.24.24.0
BGP routing table entry for 24.24.24.0/24, version 27
Paths: (1 available, best #1, table Default-IP-Routing-Table)

  Advertised to non peer-group peers:
  30.30.30.131 142.41.248.129
 3

    160.41.248.130 from 160.41.248.130 (160.41.248.130)
      Origin incomplete, metric 20, localpref 100, valid, external, best
```

Limitations and Restrictions

The limitations and restrictions you should consider when implementing this design include the following:

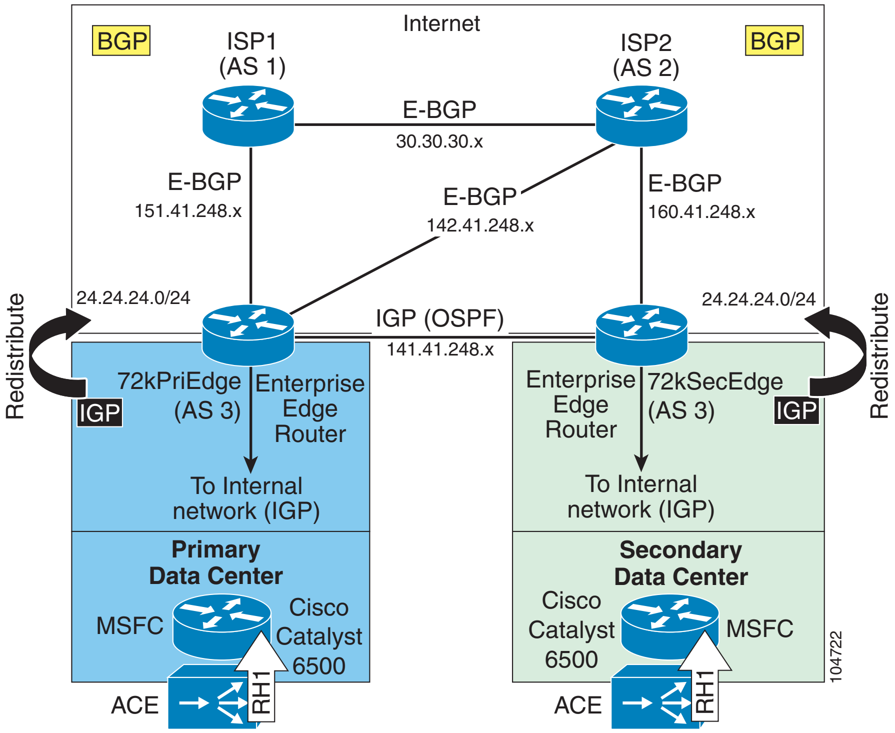
- All applications that have virtual IPs within the announced subnet (24.24.24.0, in the example) must have front-end/back-end servers in production at both sites.
- A site can not be easily taken out of service for maintenance. Routes need to be pulled from the BGP routing table for the site to be taken out of service. This can be done by on of the following methods:
- Change **router bgp** configurations on the Edge router
- Change **router ospf** configurations on the Edge router or Data Center MSFC
- Take all the relevant virtual servers out of service (**no inservice**) on the ACEs
- Long-lived TCP applications may break due to instability in the routing infrastructure.
- If there is a firewall service module (FWSM) in the data center between the ACE and the MSFC, RHI will not work.

Subnet-Based Load Balancing Using IGP Between Sites

Before using the instructions in this section to implement the load balancing solution without IGP, complete the general configuration described in OSPF Route Redistribution and Summarization, page 74 and BGP Route Redistribution and Route Preference, page 75.

Figure 26 illustrates subnet-based load balancing solution with a link between the sites and both edge routers running IGP. This design is an enhancement over the scenario without a link between the sites.

Figure 26 Subnet-Based Load Balancing with IGP Running between Sites



Changing IGP Cost for Site Maintenance

The main advantage of this solution compared to the one without a link between sites, is that you can change an active/active design to an active/standby design on the fly by changing the OSPF cost of the link. This allows you to take the secondary data center out of service by a simple IGP cost modification.

For example, if the secondary data center needs to be brought down for maintenance without stopping BGP updates, just increase the OSPF cost on the link connecting the edge router to the internal MSFC. The traffic that comes in from ISP2 to the secondary data center, can be forwarded to the primary data center over the link between sites. Internet clients and ISPs are not aware of any route changes. In most environments, this link between sites would have higher bandwidth than the path through the ISP.

The configuration for this scenario is as follows:

```
72k-edgeSecDC# sh ip route 24.24.24.0

Routing entry for 24.24.24.0/24
  Known via "ospf 1", distance 110, metric 21, type extern 1
  Redistributing via bgp 3
  Advertised by bgp 3 route-map OspfRouteFilter
  Last update from 10.10.0.6 on FastEthernet4/0, 00:10:27 ago
  Routing Descriptor Blocks:

 * 10.10.0.6, from 140.40.248.130, 00:10:27 ago, via FastEthernet4/0
      Route metric is 21, traffic share count is 1

72k-edgeSecDC# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
72k-edgeSecDC(config)# interface FastEthernet4/0
72k-edgeSecDC(config-if)# ip address 10.10.0.130 255.255.255.0
```

```
72k-edgeSecDC(config-if)# ip ospf cost 5
72k-edgeSecDC(config-if)#^Z


72k-edgeSecDC# sh ip route 24.24.24.0

Routing entry for 24.24.24.0/24
  Known via "ospf 1", distance 110, metric 22, type extern 1
  Redistributing via bgp 3
  Advertised by bgp 3 route-map OspfRouteFilter
  Last update from 141.41.248.129 on FastEthernet5/0, 00:00:12 ago
  Routing Descriptor Blocks:

 * 141.41.248.129, from 130.40.248.130, 00:00:12 ago, via FastEthernet5/0
      Route metric is 22, traffic share count is 1
```

Routes During Steady State

The following are the steady state routes for all the relevant devices.

```
72k-edgePriDC# sh ip route 24.24.24.0

Routing entry for 24.24.24.0/24
  Known via "ospf 1", distance 110, metric 21, type extern 1
  Redistributing via bgp 3
  Advertised by bgp 3 route-map OspfRouteFilter
  Last update from 10.0.0.6 on FastEthernet1/1, 00:08:57 ago
  Routing Descriptor Blocks:

 * 10.0.0.6, from 130.40.248.130, 00:08:57 ago, via FastEthernet1/1
      Route metric is 21, traffic share count is 1

72k-edgePriDC# sh ip bgp 0.0.0.0
BGP routing table entry for 0.0.0.0/0, version 6
Paths: (2 available, best #2, table Default-IP-Routing-Table)

  Not advertised to any peer
 2
    142.41.248.132 from 142.41.248.132 (160.41.248.132)
      Origin IGP, localpref 100, weight 200, valid, external
 1
    151.41.248.131 from 151.41.248.131 (151.41.248.131)
      Origin IGP, localpref 100, weight 2000, valid, external, best

Cat6k-ISP1# sh ip bgp 24.24.24.0
BGP routing table entry for 24.24.24.0/24, version 18
Paths: (2 available, best #1)

  Advertised to non peer-group peers:
  30.30.30.132
 3

    151.41.248.129 from 151.41.248.129 (151.41.248.129)
      Origin incomplete, metric 20, localpref 100, valid, external, best
 2 3
    30.30.30.132 from 30.30.30.132 (160.41.248.132)
      Origin incomplete, localpref 100, valid, external
```

```
72k-ISP2# sh ip bgp 24.24.24.0
BGP routing table entry for 24.24.24.0/24, version 7
Paths: (3 available, best #3, table Default-IP-Routing-Table)
Multipath: eBGP

  Advertised to non peer-group peers:
  30.30.30.131 142.41.248.129
 1 3

    30.30.30.131 from 30.30.30.131 (151.41.248.131)
      Origin incomplete, localpref 100, valid, external
 3
    142.41.248.129 from 142.41.248.129 (151.41.248.129)
      Origin incomplete, metric 30, localpref 100, valid, external
 3
    160.41.248.130 from 160.41.248.130 (160.41.248.130)
      Origin incomplete, metric 20, localpref 100, valid, external, best


72k-edgeSecDC# sh ip route 24.24.24.0

Routing entry for 24.24.24.0/24
  Known via "ospf 1", distance 110, metric 22, type extern 1
  Redistributing via bgp 3
  Advertised by bgp 3 route-map OspfRouteFilter
  Last update from 141.41.248.129 on FastEthernet5/0, 00:37:05 ago
  Routing Descriptor Blocks:

 * 141.41.248.129, from 130.40.248.130, 00:37:05 ago, via FastEthernet5/0
      Route metric is 22, traffic share count is 1

72k-edgeSecDC# sh ip bgp 0.0.0.0
BGP routing table entry for 0.0.0.0/0, version 27
Paths: (1 available, best #1, table Default-IP-Routing-Table)

  Not advertised to any peer
 2
    160.41.248.132 from 160.41.248.132 (160.41.248.132)

      Origin IGP, localpref 100, valid, external, best
72k-edgeSecDC#
72k-edgeSecDC#sh run int f4/0
Building configuration...

Current configuration : 100 bytes
!
interface FastEthernet4/0

 ip address 10.10.0.130 255.255.255.0
 ip ospf cost 5
 duplex full

end
```

Test Cases

Following are several test cases that were conducted to verify this design. Notice how the routes change in each test case. All the traceroutes were conducted from a host connected to ISP 1.

Test Case 1—Primary Edge Link (f2/0) to ISP1 Goes Down

The route disappeared from the ISP router within 5 seconds after a direct link failure.

```
72k-edgePriDC# sh ip bgp 0.0.0.0
BGP routing table entry for 0.0.0.0/0, version 8
Paths: (1 available, best #1, table Default-IP-Routing-Table)

  Not advertised to any peer
 2
    142.41.248.132 from 142.41.248.132 (160.41.248.132)
      Origin IGP, localpref 100, weight 200, valid, external, best

Cat6k-ISP1# sh ip bgp 24.24.24.0
BGP routing table entry for 24.24.24.0/24, version 20
Paths: (1 available, best #1)

  Not advertised to any peer
 2 3
    30.30.30.132 from 30.30.30.132 (160.41.248.132)
      Origin incomplete, localpref 100, valid, external, best

72k-ISP2# sh ip bgp 24.24.24.0
BGP routing table entry for 24.24.24.0/24, version 7
Paths: (2 available, best #2, table Default-IP-Routing-Table)
Multipath: eBGP

  Advertised to non peer-group peers:
  30.30.30.131 142.41.248.129
 3

    142.41.248.129 from 142.41.248.129 (151.41.248.129)
      Origin incomplete, metric 30, localpref 100, valid, external
 3
    160.41.248.130 from 160.41.248.130 (160.41.248.130)
      Origin incomplete, metric 20, localpref 100, valid, external, best

3500AP# traceroute 24.24.24.1
Type escape sequence to abort.
Tracing the route to 24.24.24.1

  1 55.55.1.1 3 msec 3 msec 2 msec
2 30.30.30.132 2 msec 5 msec
  3 160.41.248.130 3 msec 2 msec 3 msec

  4 141.41.248.129 2 msec 3 msec 5 msec
  5 10.0.0.6 2 msec 3 msec 3 msec
6  *  * *

(ISP1)
(ISP2)
(Secondary Datacenter)
(Primary Datacenter)
(Cat6k in Primary Datacenter)
```

Test Case 2—Primary Edge Link (f2/0) to ISP1 and Link (f3/0) to ISP2 Goes Down

```
72k-edgePriDC# sh ip route 0.0.0.0
```

```
Routing entry for 0.0.0.0/0, supernet
  Known via "ospf 1", distance 110, metric 1, candidate default path
  Tag 1, type extern 2, forward metric 1
  Redistributing via bgp 3
  Last update from 141.41.248.130 on FastEthernet0/0, 00:00:45 ago
  Routing Descriptor Blocks:

 * 141.41.248.130, from 160.41.248.130, 00:00:45 ago, via FastEthernet0/0
      Route metric is 1, traffic share count is 1
      Route tag 1


72k-edgeSecDC# sh ip route 24.24.24.0

Routing entry for 24.24.24.0/24
  Known via "ospf 1", distance 110, metric 22, type extern 1
  Redistributing via bgp 3
  Advertised by bgp 3 route-map OspfRouteFilter

Last update from 141.41.248.129 on FastEthernet5/0, 00:52:32 ago
  Routing Descriptor Blocks:

 * 141.41.248.129, from 130.40.248.130, 00:52:32 ago, via FastEthernet5/0
      Route metric is 22, traffic share count is 1

Cat6k-ISP1# sh ip bgp 24.24.24.0
BGP routing table entry for 24.24.24.0/24, version 20
Paths: (1 available, best #1)

  Not advertised to any peer
 2 3
    30.30.30.132 from 30.30.30.132 (160.41.248.132)
      Origin incomplete, localpref 100, valid, external, best

72k-ISP2# sh ip bgp 24.24.24.0
BGP routing table entry for 24.24.24.0/24, version 7
Paths: (1 available, best #1, table Default-IP-Routing-Table)
Multipath: eBGP

  Advertised to non peer-group peers:
  30.30.30.131
 3

    160.41.248.130 from 160.41.248.130 (160.41.248.130)
      Origin incomplete, metric 20, localpref 100, valid, external, best

3500AP# traceroute 24.24.24.1

Type escape sequence to abort.
Tracing the route to 24.24.24.1 1 55.55.1.1 0 msec 3 msec 0 msec
(ISP1)2 30.30.30.132 3 msec 3 msec 2 msec
(ISP2)3 160.41.248.130 3 msec 3 msec 3 msec
(Secondary Site Edge Router)4 141.41.248.129 2 msec 3 msec 2 msec
(Primary Site Edge Router )5 10.0.0.6 6 msec 2 msec 3 msec
(Primary Data Center)6  *  *
 *
```

Test Case 3—Primary Data Center ACE Goes Down

```
cat6K_l# sh ip route static
72k-edgePriDC# sh ip bgp 0.0.0.0
BGP routing table entry for 0.0.0.0/0, version 17
Paths: (2 available, best #2, table Default-IP-Routing-Table)

  Not advertised to any peer
 2
    142.41.248.132 from 142.41.248.132 (160.41.248.132)
      Origin IGP, localpref 100, weight 200, valid, external
 1
    151.41.248.131 from 151.41.248.131 (151.41.248.131)
      Origin IGP, localpref 100, weight 2000, valid, external, best

72k-edgePriDC# sh ip route 24.24.24.0

Routing entry for 24.24.24.0/24
  Known via "ospf 1", distance 110, metric 26, type extern 1
  Redistributing via bgp 3
  Advertised by bgp 3 route-map OspfRouteFilter
  Last update from 141.41.248.130 on FastEthernet0/0, 00:01:04 ago
  Routing Descriptor Blocks:

 * 141.41.248.130, from 140.40.248.130, 00:01:04 ago, via FastEthernet0/0
      Route metric is 26, traffic share count is 1

72k-edgeSecDC# sh ip bgp 0.0.0.0
BGP routing table entry for 0.0.0.0/0, version 27
Paths: (1 available, best #1, table Default-IP-Routing-Table)

Not advertised to any peer
 2
    160.41.248.132 from 160.41.248.132 (160.41.248.132)

      Origin IGP, localpref 100, valid, external, best
72k-edgeSecDC#
72k-edgeSecDC#
72k-edgeSecDC#sh ip route 24.24.24.0
Routing entry for 24.24.24.0/24

  Known via "ospf 1", distance 110, metric 25, type extern 1
  Redistributing via bgp 3
  Advertised by bgp 3 route-map OspfRouteFilter
  Last update from 10.10.0.6 on FastEthernet4/0, 00:01:49 ago
  Routing Descriptor Blocks:

 * 10.10.0.6, from 140.40.248.130, 00:01:49 ago, via FastEthernet4/0
      Route metric is 25, traffic share count is 1

Cat6k-ISP1# sh ip bgp 24.24.24.0
BGP routing table entry for 24.24.24.0/24, version 27
Paths: (2 available, best #1)

  Advertised to non peer-group peers:
  30.30.30.132
 3

    151.41.248.129 from 151.41.248.129 (151.41.248.129)
      Origin incomplete, metric 20, localpref 100, valid, external, best
 2 3
```

```
    30.30.30.132 from 30.30.30.132 (160.41.248.132)
      Origin incomplete, localpref 100, valid, external

72k-ISP2# sh ip bgp 24.24.24.0
BGP routing table entry for 24.24.24.0/24, version 7
Paths: (3 available, best #3, table Default-IP-Routing-Table)
Multipath: eBGP

  Advertised to non peer-group peers:
  30.30.30.131 142.41.248.129
 3

    142.41.248.129 from 142.41.248.129 (151.41.248.129)
      Origin incomplete, metric 30, localpref 100, valid, external
 1 3
    30.30.30.131 from 30.30.30.131 (151.41.248.131)
      Origin incomplete, localpref 100, valid, external
 3
    160.41.248.130 from 160.41.248.130 (160.41.248.130)
      Origin incomplete, metric 20, localpref 100, valid, external, best


3500AP# traceroute 24.24.24.1

Type escape sequence to abort.
Tracing the route to 24.24.24.1

  1 55.55.1.1 2 msec 3 msec 2 msec (ISP1)
  2 151.41.248.129 3 msec 2 msec 0 msec (Primary Site Edge Router )
  3 141.41.248.130 0 msec 2 msec 0 msec (Secondary Site Edge Router)
  4 10.10.0.6 2 msec 2 msec 3 msec (Secondary Data Center)
 5  *  * *
```

Limitations and Restrictions

- All applications that have virtual IPs within the announced subnet (24.24.24.0, in the example) must have front-end/back-end servers in production at both sites.
- Long-lived TCP applications may break due to instability in the routing infrastructure.
- For RHI to function, ACE needs to share a VLAN with the MSFC in the same Catalyst 6500 chassis.
- The internal path between the sites should not have any firewalls. If a firewall is present, it should be either a Layer 2 firewall or a Layer 3 firewall capable of running IGP protocol (such as OSPF).

Application-Based Load Balancing Using IGP Between Sites

This application-based design is a further enhancement over the subnet-based design. Before using the instructions in this section to implement the application-based load balancing solution with IGP, complete the general configuration described in OSPF Route Redistribution and Summarization, page 74 and BGP Route Redistribution and Route Preference, page 75.

In this design we move the RHI host route summarization up to the BGP layer on the edge routers. The RHI host route for each application is given a different weight so that some applications are active on the primary site while others are active in the secondary site. You can keep an instance of the application at both sites for backup purposes, as required.

Weights are used on the data center MSFC to control active/standby behavior for the virtual IP for each. Local active applications are given a lower metric compared to a remote application. For example, the primary site in this example is primary for 24.24.24.1 and 24.24.24.2 while the secondary site is standby. The secondary site is primary for 24.24.24.3 while the primary site is standby for this route and the associated application.

Configuration on all devices stays the same as in the previous scenario except for the configurations shown below.

Configuration on Primary Site

Primary Data Center Catalyst 6500

```
!

router ospf 1
 log-adjacency-changes
 redistribute static metric-type 1 subnets route-map REMOTE-APP
 network 10.0.0.0 0.0.0.255 area 0
 network 10.4.0.16 0.0.0.3 area 0
 network 10.4.1.0 0.0.0.255 area 0
 network 10.6.0.16 0.0.0.3 area 0
 network 130.40.248.0 0.0.0.255 area 0

!
access-list 11 permit 24.24.24.3
!
route-map REMOTE-APP permit 10

 match ip address 11

 set metric 30
!
route-map REMOTE-APP permit 30

 set metric 20
!
```

Primary Data Center Edge Router

```
!

router ospf 1
 log-adjacency-changes
 network 10.0.0.0 0.0.0.255 area 0
 network 141.41.248.128 0.0.0.127 area 1
 default-information originate

!

router bgp 3
 no synchronization
 aggregate-address 24.24.24.0 255.255.255.0 summary-only
 <SNIP>
 no auto-summary
```

```
!
!
```

Configuration on Secondary Site

Secondary Data Center Catalyst 6500

```
!

router ospf 1
 log-adjacency-changes
 redistribute static metric-type 1 subnets route-map REMOTE-APP
 network 10.10.0.0 0.0.0.255 area 1
 network 10.14.0.16 0.0.0.3 area 1
 network 10.14.1.0 0.0.0.255 area 1
 network 10.16.0.16 0.0.0.3 area 1
 network 140.40.248.0 0.0.0.255 area 1

!
!
access-list 11 permit 24.24.24.1
access-list 11 permit 24.24.24.2
!
route-map REMOTE-APP permit 10

 match ip address 11

 set metric 30
!
route-map REMOTE-APP permit 30

 set metric 20
!
!
```

Secondary Data Center Edge Router

```
router ospf 1
 log-adjacency-changes
 network 10.10.0.0 0.0.0.255 area 1
 network 141.41.248.128 0.0.0.127 area 1
 default-information originate

!

router bgp 3
 no synchronization
 aggregate-address 24.24.24.0 255.255.255.0 summary-only
 <SNIP>
 no auto-summary

!
!
```

Routes During Steady State

Primary Edge Router

```
72k-edgePriDC# sh ip route | in 24.24.24
B 24.24.24.0/24 [200/0] via 0.0.0.0, 00:14:45, Null0
O E1 24.24.24.1/32 [110/21] via 10.0.0.6, 00:14:41, FastEthernet1/1
O E1 24.24.24.2/32 [110/21] via 10.0.0.6, 00:14:41, FastEthernet1/1

O E1 24.24.24.3/32 [110/22] via 141.41.248.130, 00:14:41, FastEthernet0/0
```

Secondary Edge Router

```
72k-edgeSecDC# sh ip route | in 24.24.24
B 24.24.24.0/24 [200/0] via 0.0.0.0, 00:15:17, Null0
O E1 24.24.24.1/32 [110/22] via 141.41.248.129, 00:15:13, FastEthernet5/0
O E1 24.24.24.2/32 [110/22] via 141.41.248.129, 00:15:13, FastEthernet5/0
O E1 24.24.24.3/32 [110/21] via 10.10.0.6, 00:15:13, FastEthernet4/0
```

Test Case 1—Servers Down at Primary Site

The following IGP route changed within 5 seconds. The BGP route change also took 5 seconds.

Primary Edge Router

```
72k-edgePriDC# sh ip route | in 24.24.24
B 24.24.24.0/24 [200/0] via 0.0.0.0, 00:16:50, Null0
O E1 24.24.24.1/32 [110/32] via 141.41.248.130, 00:00:04, FastEthernet0/0
O E1 24.24.24.2/32 [110/32] via 141.41.248.130, 00:00:04, FastEthernet0/0
O E1 24.24.24.3/32 [110/22] via 141.41.248.130, 00:00:04, FastEthernet0/0
```

Secondary Edge Router

```
72k-edgeSecDC# sh ip route | in 24.24.24
B 24.24.24.0/24 [200/0] via 0.0.0.0, 00:16:59, Null0
O E1 24.24.24.1/32 [110/31] via 10.10.0.6, 00:00:14, FastEthernet4/0
O E1 24.24.24.2/32 [110/31] via 10.10.0.6, 00:00:13, FastEthernet4/0
O E1 24.24.24.3/32 [110/21] via 10.10.0.6, 00:00:13, FastEthernet4/0

3500AP# traceroute 24.24.24.3
Type escape sequence to abort.
Tracing the route to 24.24.24.3

  1 55.55.1.1 0 msec 0 msec 2 msec
  2 151.41.248.129 0 msec 0 msec 2 msec
  3 141.41.248.130 3 msec 3 msec 0 msec
  4 10.10.0.6 5 msec 2 msec 0 msec
5  *  * *
```

```
(ISP1)
(Primary Site Edge Router)
(Secondary site Edge Router)
(Secondary Data Center)
```

Limitations and Restrictions

- Long lived TCP applications may break due to instability in the routing infrastructure.
- For RHI to work, the ACE needs to share a VLAN with the MSFC in the same Catalyst 6500 chassis.
- The internal path between the sites should not have any firewalls. If a firewall is present, it should be either a Layer 2 firewall or a Layer 3 firewall capable of running IGP protocol (such as OSPF).

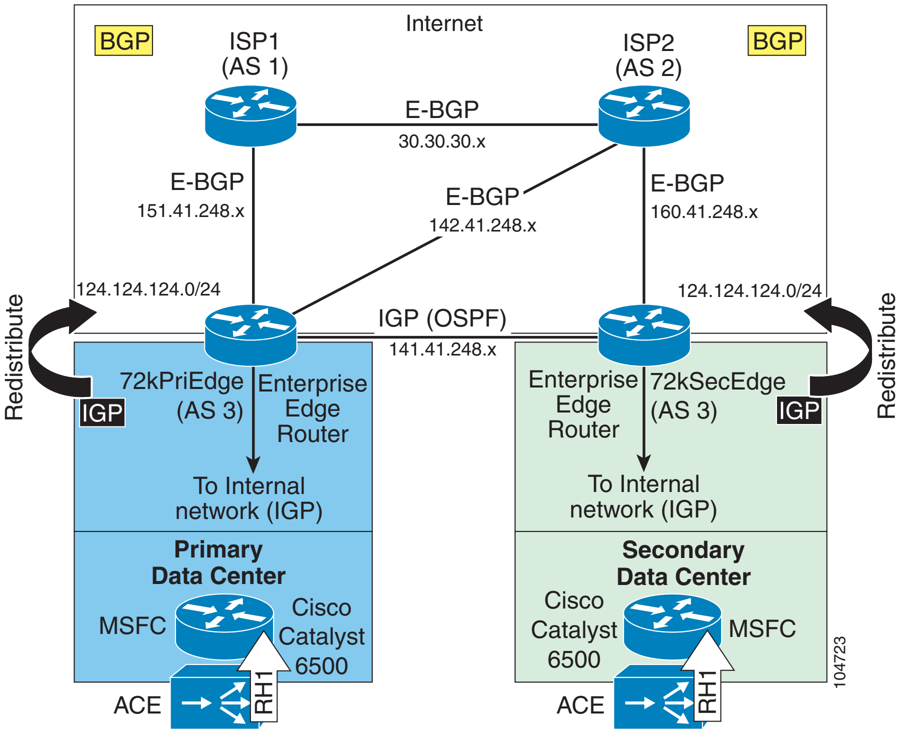
Using NAT in Active/Active Load Balancing Solutions

Figure 27 illustrates the changes in application-based design required for a NAT-based solution. This option is an enhancement that can be used in all the active/active load balancing scenarios described in this chapter.

In this example, the virtual IPs used to represent the applications are private IPs that are one-to-one NATed to public IP addresses. For example, the private address 24.24.24.1 is NATed to the public address 124.124.124.1. In a production network, the 24.24.24.0 subnet would be in the 10.0.0.0 address space.

Note To implement NAT on a design without IGP running between the sites, you have to use appropriate conditional advertisements when redistributing the configured static route into BGP. Otherwise, traffic may be black-holed.

Figure 27 Application-Based Load Balancing Sites with IGP Running Between Data Centers



All configuration stays the same except for the addition of IOS NAT configuration and the redistribution of the public IP addresses into BGP. For example, in the configuration below, 124.124.124.0 is advertised to the external world by redistributing a NULL0 static route into BGP.

Primary Site Edge Router Configuration

```
interface FastEthernet0/0
 description "To 5/0 72kEdge(IBGP)"
 ip address 141.41.248.129 255.255.255.192
 ip nat inside
 duplex full

!

interface FastEthernet1/1
 description To cat6k-l f2/26
 ip address 10.0.0.129 255.255.255.0
 ip nat inside
 duplex full

!
interface FastEthernet2/0
 description "To 2/13 Cat6k(ISP1)"

ip address 151.41.248.129 255.255.255.192
 ip nat outside
 duplex full

!
```

```
interface FastEthernet3/0
 description "To 2/1 72k(ISP2)"
 ip address 142.41.248.129 255.255.255.192
 ip nat outside
 duplex full

!
!
router bgp 3

 no synchronization
 bgp log-neighbor-changes
 redistribute static
 <SNIP>
 no auto-summary

!
ip nat inside source static 24.24.24.1 124.124.124.1
ip nat inside source static 24.24.24.2 124.124.124.2
ip nat inside source static 24.24.24.3 124.124.124.3
!
ip route 124.124.124.0 255.255.255.0 Null0
!
!
```

Secondary Site Edge Router Configuration

```
interface FastEthernet3/0
 description "To 3/0 72k(ISP2)"
 ip address 160.41.248.130 255.255.255.192
 ip nat outside
 duplex full

!

interface FastEthernet4/0
 ip address 10.10.0.130 255.255.255.0
 ip nat inside
 duplex full

!

interface FastEthernet5/0
 description "To 0/0 72kPriDC"
 ip address 141.41.248.130 255.255.255.192
 ip nat inside
 duplex full

!
!
router bgp 3

 no synchronization
 redistribute static
 <SNIP>
 no auto-summary

!
ip route 124.124.124.0 255.255.255.0 Null0
!
!
```

```
ip nat inside source static 24.24.24.1 124.124.124.1
ip nat inside source static 24.24.24.2 124.124.124.2
ip nat inside source static 24.24.24.3 124.124.124.3
!
72k-edgeSecDC#
```

Steady State Routes

```
72k-edgePriDC# sh ip bgp 0.0.0.0
BGP routing table entry for 0.0.0.0/0, version 57
Paths: (2 available, best #1, table Default-IP-Routing-Table)

  Not advertised to any peer
 1
    151.41.248.131 from 151.41.248.131 (151.41.248.131)
      Origin IGP, localpref 100, weight 2000, valid, external, best
 2
    142.41.248.132 from 142.41.248.132 (160.41.248.132)

      Origin IGP, localpref 100, weight 200, valid, external
72k-edgePriDC#
72k-edgePriDC#
72k-edgePriDC#sh ip route | in 24.24.24.
O E1 24.24.24.1 [110/21] via 10.0.0.6, 1d10h, FastEthernet1/1
O E1 24.24.24.2 [110/21] via 10.0.0.6, 1d10h, FastEthernet1/1
O E1 24.24.24.3 [110/22] via 141.41.248.130, 1d10h, FastEthernet0/0

72k-edgePriDC# sh ip route | in 124.124.124.
S 124.124.124.0 is directly connected, Null0

Cat6k-ISP1# sh ip bgp
BGP table version is 6, local router ID is 151.41.248.131
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network  Next Hop       Metric LocPrf Weight Path

* 124.124.124.0/24 30.30.30.132  0 2 3 ?
*> 151.41.248.129 20 0 3 ?

* 142.41.248.128/26


 30.30.30.132 0 0 2 i
*> 151.41.248.129 20 0 3 i
*> 151.41.248.128/26

 0.0.0.0 0 32768 i

* 30.30.30.132 0 2 3 i

* 151.41.248.129 20 0 3 i


*> 160.41.248.128/26
 30.30.30.132 0 0 2 i

Cat6k-ISP1# sh ip bgp 124.124.124.0
BGP routing table entry for 124.124.124.0/24, version 2
Paths: (2 available, best #2)
```

```
  Advertised to non peer-group peers:
  30.30.30.132
 2 3

    30.30.30.132 from 30.30.30.132 (160.41.248.132)
      Origin incomplete, localpref 100, valid, external
 3
    151.41.248.129 from 151.41.248.129 (151.41.248.129)
      Origin incomplete, metric 20, localpref 100, valid, external, best

72k-ISP2# sh ip bgp
BGP table version is 16, local router ID is 160.41.248.132
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,

     r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network  Next Hop       Metric LocPrf Weight Path

* 124.124.124.0/24 30.30.30.131  0 1 3 ?

* 142.41.248.129 30 0 3 ?
*> 160.41.248.130 20 0 3 ?

* 142.41.248.128/26


30.30.30.131 0 1 3 i
*> 0.0.0.0 0 32768 i

*            142.41.248.129  30 0 3 i

* 151.41.248.128/26


 30.30.30.131 0 0 1 i
*>           142.41.248.129  30 0 3 i
*> 160.41.248.128/26

 0.0.0.0 0 32768 i

72k-ISP2# sh ip bgp 124.124.124.0
BGP routing table entry for 124.124.124.0/24, version 3
Paths: (3 available, best #3, table Default-IP-Routing-Table)

  Advertised to non peer-group peers:
  30.30.30.131 142.41.248.129
 1 3

    30.30.30.131 from 30.30.30.131 (151.41.248.131)
      Origin incomplete, localpref 100, valid, external
 3
    142.41.248.129 from 142.41.248.129 (151.41.248.129)
      Origin incomplete, metric 30, localpref 100, valid, external
 3
    160.41.248.130 from 160.41.248.130 (160.41.248.130)
      Origin incomplete, metric 20, localpref 100, valid, external, best

72k-edgeSecDC# sh ip bgp 0.0.0.0
BGP routing table entry for 0.0.0.0/0, version 25
Paths: (1 available, best #1, table Default-IP-Routing-Table)
```

```
  Not advertised to any peer
 2
    160.41.248.132 from 160.41.248.132 (160.41.248.132)
      Origin IGP, localpref 100, valid, external, best

72k-edgeSecDC# sh ip route | in 24.24.24.
O E1 24.24.24.1 [110/22] via 141.41.248.129, 1d10h, FastEthernet5/0
O E1 24.24.24.2 [110/22] via 141.41.248.129, 1d10h, FastEthernet5/0
O E1 24.24.24.3 [110/21] via 10.10.0.6, 1d10h, FastEthernet4/0

72k-edgeSecDC# sh ip route | in 124.124.124.
S 124.124.124.0 is directly connected, Null0

3500AP# traceroute 124.124.124.1

Type escape sequence to abort.
Tracing the route to 124.124.124.1


  1 55.55.1.1 5 msec 2 msec 3 msec
  2 151.41.248.129 2 msec 2 msec 3 msec
  3 10.0.0.6 3 msec 6 msec 2 msec
4  *  * *

(ISP1)
(Primary Site Edge Router)
(Primary Site Data Center)
```

Routes When Servers in Primary Data Center Goes Down

There are no route changes on either ISP.

```
72k-edgePriDC# sh ip route | in 24.24.24.
O E1 24.24.24.1 [110/32] via 141.41.248.130, 00:00:31, FastEthernet0/0
O E1 24.24.24.2 [110/32] via 141.41.248.130, 00:00:31, FastEthernet0/0
O E1 24.24.24.3 [110/22] via 141.41.248.130, 00:00:31, FastEthernet0/0

72k-edgeSecDC# sh ip route | in 24.24.24.
O E1 24.24.24.1 [110/31] via 10.10.0.6, 00:00:46, FastEthernet4/0
O E1 24.24.24.2 [110/31] via 10.10.0.6, 00:00:46, FastEthernet4/0

O E1 24.24.24.3 [110/21] via 10.10.0.6, 00:00:46, FastEthernet4/0

3500AP# traceroute 124.124.124.1

Type escape sequence to abort.
Tracing the route to 124.124.124.1

  1 55.55.1.1 6 msec 5 msec 3 msec
  2 151.41.248.129 2 msec 3 msec 0 msec
  3 141.41.248.130 6 msec 2 msec 3 msec
  4 10.10.0.6 2 msec 3 msec 2 msec

5

(ISP1)
(Primary Site Edge Router)
(Secondary Site Edge Router)
```

```
(Secondary Site Data Center)

* * *
```

Route Health Injection

The **advertise active** command available with virtual servers tells the ACE to install a host route in the MSFC on the same Catalyst 6500 chassis only if the virtual server is in Operational state. A virtual server is in the Operational state when at least one of the servers in the same server farm is Operational. Extensive probing is available on the ACE to check the health of the server and the appropriate application daemon that runs on the server. For RHI to work, MSFC and ACE **must** share a client-side VLAN (VLAN26 in this example). It is important to remember that on the ACE module that you must also use the route inject command on the vlan interface for the route to be inserted.

The following is a minimal RHI configuration used in testing.

```
Configuration commands for Admin context:

Context1
  allocate-interface vlan 26


Configuration commands for  context1:

access-list LB_ALLOW_VIPS extended permit tcp any 24.24.24.1 255.255.255.255 eq www
access-list LB_ALLOW_VIPS extended permit tcp any 24.24.24.2 255.255.255.255 eq www
access-list LB_ALLOW_VIPS extended permit tcp any 24.24.24.3 255.255.255.255 eq www


probe icmp ICMP
  faildetect 2
  interval 5


rserver host 130-34-248-129
  inservice
  ip address 130.34.248.129


serverfarm host RHI-TEST
  probe ICMP
  rserver 130-34-248-129
    inservice


parameter-map type http RHI-TEST-1_HTTP
  persistence-rebalance
parameter-map type http RHI-TEST-2_HTTP
  persistence-rebalance
parameter-map type http RHI-TEST-3_HTTP
  persistence-rebalance


class-map type management match-any TO-CP-POLICY
  match protocol http any
  match protocol telnet any
  match protocol icmp any
class-map match-all RHI-TEST-1
  match virtual-address 24.24.24.1 tcp eq www
class-map match-all RHI-TEST-2
```

```
  match virtual-address 24.24.24.2 tcp eq www
class-map match-all RHI-TEST-3
  match virtual-address 24.24.24.3 tcp eq www


policy-map type management first-match TO-CP-POLICY
  class TO-CP-POLICY
    permit
policy-map type loadbalance first-match RHI-TEST-1
  class class-default
    serverfarm RHI-TEST
policy-map type loadbalance first-match RHI-TEST-2
  class class-default
    serverfarm RHI-TEST
policy-map type loadbalance first-match RHI-TEST-3
  class class-default
    serverfarm RHI-TEST
policy-map multi-match POLICY548280
  class RHI-TEST-1
    loadbalance vip advertise active
    appl-parameter http advanced-options RHI-TEST-1_HTTP
    loadbalance policy RHI-TEST-1
    loadbalance vip inservice
    loadbalance vip icmp-reply active
  class RHI-TEST-2
    loadbalance vip advertise active
    appl-parameter http advanced-options RHI-TEST-2_HTTP
    loadbalance policy RHI-TEST-2
    loadbalance vip inservice
    loadbalance vip icmp-reply active
  class RHI-TEST-3
    loadbalance vip advertise active
    appl-parameter http advanced-options RHI-TEST-3_HTTP
    loadbalance policy RHI-TEST-3
    loadbalance vip inservice
    loadbalance vip icmp-reply active


interface vlan 26
  ip address 10.16.0.2 255.255.255.0
  access-group input LB_ALLOW_VIPS
  alias 10.16.0.3 255.255.255.0
  service-policy input POLICY548280
  service-policy input TO-CP-POLICY
  no shutdown
interface vlan 14
  ip address 130.34.248.161 255.255.255.192
  no shutdown


ft group 1
  priority 110
ft peer 1




ip route 0.0.0.0 255.255.255.0 10.16.0.1
```

The following is the configuration on the interface on the MSFC that connects to the ACE.

```
interface Vlan26
```

```
 ip address 10.16.0.1 255.255.255.0
end

cat6K_l# sh mod c 4 vlan id 26 detail
vlan IP address  IP mask type



26    10.16.0.2          255.255.255.0SERVER
 ALIASES

  IP address     IP mask



  --------------------------------

  10.16.0.3  255.255.255.0
```

The following shows the static route in the MSFC routing table pointing to the Alias on the ACE. An Alias is a shared IP address, similar to a Hot Standby Router Protocol (HSRP) group IP address.

```
cat6K_l#cat6K_l# sh ip route static

     24.0.0.0/32 is subnetted, 3 subnets
S 24.24.24.1 [1/0] via 10.16.0.3, Vlan26
S 24.24.24.2 [1/0] via 10.16.0.3, Vlan26
S 24.24.24.3 [1/0] via 10.16.0.3, Vlan26
```

Glossary

C

- **Content and Application Peering Protocol (CAPP)**—Enables distributed CSS to exchange global proximity information in real time. The use of CAPP ensures that all content requests draw on complete and accurate proximity information. In a session between two Content Services Switches serving Global Load Balancing function to exchange information about the site and make decisions on redirecting clients to the appropriate sited based on the information.
- **Content Routing—**The ability of the network to take a client request and redirect it to an appropriate resource for servicing. There are a number of ways to accomplish this, some proprietary some not. The three most common ways of accomplishing this are DNS-based redirection, HTTP-based redirection, and Route Health Injection.
- **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.

D

- **Distributed Data Center**—Consists of more than one data centers connected together by WAN/LAN or high speed transport layer to provide redundant data center services. These distributed data centers also share load between them.

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

- **Health Checks or Health Probes**—Used by the server load balancing and global 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.
- **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.

N

- **NAS (Network Attached Storage)**—A central data storage system that is attached to the network that it serves. A File Server with internal or external storage is a simple example.
- **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.

O

- **Origin Web Server**—Core of Content Networking. Base from where web services are sourced.

R

- **Real Server**—Physical server providing the services behind the virtual server to the clients.

S

- **SAN (Storage Area Network)**—A dedicated, centrally managed, secure information infrastructure that enables any-to-any interconnection of servers and storage systems. SANs are typically built using the SCSI and Fibre Channel (SCSI-FCP) protocols.
- **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 off loading 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.
- **Source of Authority (SOA** —The primary DNS server for a particular domain.
- **SRDF (Symmetrix Remote Data Facility)**—EMC Symmetrix Remote Data Facility enables real-time data replication between processing environments. This can be the same data center or separated by longer distances.
- **Stateful Failover**—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.
- **Stateless Failover**—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.
- **Storage Array (SA)** —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).

T

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

U

- **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.

W

- **Weighted Round Robin** —When weights are assigned to different sites and clients are directed to sites in a round robin fashion based on the weights assigned to different sites, the sites with the highest weight take end up with more clients.
- **Wavelength** —The distance between points of corresponding phase of two consecutive cycles of a wave. In DWDM systems, wavelength is also called lambda.