Data Center Fundamentals

Understand Data Center network design and infrastructure architecture, including load balancing, SSL, and security

Mauricio Arregoces, CCIE® No. 3285
Maurizio Portolani

ciscopress.com

Data Center Fundamentals

Mauricio Arregoces

Maurizio Portolani

Published by:
Cisco Press
800 East 96th Street
Indianapolis, IN 46240 USA

ISBN: 1-58705-023-4

Library of Congress Cataloging-in-Publication Number: 2001086631

Printed in the United States of America 6 7 8 9 0

Sixth Printing August 2009

Trademark Acknowledgments

All terms mentioned in this book that are known to be trademarks or service marks have been appropriately capitalized. Cisco Press or Cisco Systems, Inc., cannot attest to the accuracy of this information. Use of a term in this book should not be regarded as affecting the validity of any trademark or service mark.

Warning and Disclaimer

This book is designed to provide information about Data Center technologies. Every effort has been made to make this book as complete and as accurate as possible, but no warranty or fitness is implied.

The information is provided on an "as is" basis. The authors, Cisco Press, and Cisco Systems, Inc., shall have neither liability nor responsibility to any person or entity with respect to any loss or damages arising from the information contained in this book or from the use of the discs or programs that may accompany it.

The opinions expressed in this book belong to the author and are not necessarily those of Cisco Systems, Inc.

Feedback Information

At Cisco Press, our goal is to create in-depth technical books of the highest quality and value. Each book is crafted with care and precision, undergoing rigorous development that involves the unique expertise of members from the professional technical community.

Readers' feedback is a natural continuation of this process. If you have any comments regarding how we could improve the quality of this book or otherwise alter it to better suit your needs, you can contact us through e-mail at feedback@ciscopress.com. Please make sure to include the book title and ISBN in your message.

We greatly appreciate your assistance.

Corporate and Government Sales

Cisco Press offers excellent discounts on this book when ordered in quantity for bulk purchases or special sales. For more information, please contact:

U.S. Corporate and Government Sales 1-800-382-3419 corpsales@pearsontechgroup.com

For sales outside of the U.S. please contact: **International Sales** 1-317-581-3793 international@pearsontechgroup.com

Publisher	John Wait
Editor-In-Chief	John Kane
Cisco Representative	Anthony Wolfenden
Cisco Press Program Manager	Nannette M. Noble
Production Manager	Patrick Kanouse
Development Editors	Christopher Cleveland Betsey Henkels
Senior Project Editor	Sheri Cain
Copy Editors	Krista Hansing, Kris Simmons
Technical Editors	Mario Baldi, Robert Batz, Mark Gallo, Ron Hromoko, Fabio Maino, Scott Van de Houten, Stefano Testa, Brian Walck
Team Coordinator	Tammi Barnett
Cover Designer	Louisa Adair
Composition	Octal Publishing, Inc.
Indexers	Tim Wright, Eric Schroeder
Proofreader	Angela Rosio

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

European Headquarters
Cisco Systems International BV
Haarlerbergpark
Haarlerbergweg 13-19
1101 CH Amsterdam
The Netherlands
www-europe.cisco.com
Tel: 31 0 20 357 1000
Fax: 31 0 20 357 1100

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
www.cisco.com
Tel: 408 526-7660
Fax: 408 527-0883

Asia Pacific Headquarters
Cisco Systems, Inc.
Capital Tower
168 Robinson Road
#22-01 to #29-01
Singapore 068912
www.cisco.com
Tel: +65 6317 7777
Fax: +65 6317 7799

Introduction

Data Centers are complex systems encompassing a wide variety of technologies that are constantly evolving. Designing and maintaining a Data Center network requires skills and knowledge that range from routing and switching to load balancing and security, including the essential knowledge of servers and applications.

This books addresses both fundamental information such as the protocols used by switches and routers; the protocols used in application environments; the network technology used to build the Data Center infrastructure and secure, scale, and manage the application environments; and design best practices. We hope this book becomes your Data Center reference on protocols, technology, and design.

Motivation for Writing This Book

While speaking to networkers abroad on the topic of server load balancing, we realized that we could only convey the benefits of the technology by explaining application layer information and describing the larger design issues common in application environments.

Often through discussions with customers, the subjects related to load balancing take a back seat as issues of integration with the entire Data Center take the forefront. This book attempts to cover the breadth and depth of the Data Center IP network. The storage network and distributed Data Center topics will be the subjects of other books.

Having designed campus and Data Center networks, and having developed and supported technologies that are often referred to as *content networking* (load balancing, Secure Socket Layer [SSL] offloading, and DNS routing), we felt the need for a book that described these topics in a single place and focused on what is relevant to the Data Center. This area is what this book is about: it is an all-encompassing view of Data Centers from routing and switching technologies to application-aware technologies.

Who Should Read This Book

This book is intended for any person or organization seeking to understand Data Center networks: the fundamental protocols used by the applications and the network, the typical network technologies, and their design aspects. The book is meant to be both a reference on protocols and technology and a design and implementation guide for personnel responsible for planning, designing, implementing, and operating Data Center networks.

Chapter Organization

This book has six parts. This book is designed to be read in order from the overview of the Data Center environment, through the server farms and infrastructure protocols, to security and load-balancing concepts, before you reach the Data Center design chapters. This organization also allows you to go directly to the desired chapter if you already know the information in the previous chapters.

Part I, "An Introduction to Server Farms," includes chapters that contain an overview of the architecture of Data Centers, servers, and applications. This part also introduces the security and load-balancing technology:

- Chapter 1, "Overview of Data Centers," presents Data Center environments, the Data Center architecture, and services that are used as a guide to the rest of the book.
- Chapter 2, "Server Architecture Overview," explores the architecture of servers. This chapter covers topics such as how servers process TCP and User Datagram Protocol (UDP) traffic, how processes and threads are used, and server health.

- Chapter 3, "Application Architectures Overview," explores the application environments and how applications are architected. This chapter includes discussions on the relation between the application architectures and the design of the Data Center, the n-tier model, HTML and XML, user-agent technologies, web server technologies, and clustering technologies. This chapter introduces application concepts that are developed in Chapter 18 and Chapter 19.
- Chapter 4, "Data Center Design Overview," discusses the types of server farms on Data Centers, generic and alternative Layer 2 and Layer 3 designs, multitier designs, high availability, Data Center services, and trends that might affect Data Center designs.
- Chapter 5, "Data Center Security Overview," discusses threats, vulnerabilities and common attacks, network security devices such as firewalls and intrusion detection systems (IDSs), and other fundamental security concepts such as cryptography; VPNs; and authentication, authorization and accounting (AAA).
- Chapter 6, "Server Load-Balancing Overview," discusses reasons for load balancing, fundamental load-balancing concepts, high-availability considerations, and generic load-balancing architectures. The fundamental load-balancing concepts include Layer 4 and Layer 5 load balancing, session tracking, session persistence, and server health.

Part II, "Server Farm Protocols," explores the fundamental protocols used in server farms:

- Chapter 7, "IP, TCP, and UDP," explores the protocol headers details and their relevance to network design issues.
- Chapter 8, "HTTP and Related Concepts," discusses key concepts such as Uniform Resource Identifiers (URIs) and URLs, Multipurpose Internet Mail Extension (MIME) and its relation to HTTP entities, and HTTP header details. Chapter 8 provides additional information on the operation of HTTP, the different versions and their performance characteristics.
- Chapter 9, "SSL and TLS," discusses SSL operations with specific focus on SSL session establishment, ciphersuites, and SSL performance considerations. Chapter 15 provides additional information on the public-key infrastructure (PKI), certificates, and more security-related aspects of SSL.
- Chapter 10, "DNS Essentials and Site-Selection Considerations," explores how the DNS namespace is organized, the DNS components in the Internet, how the DNS resolution process works, DNS configuration options, DNS server placement in the network, and how to use DNS to distribute application requests to multiple Data Centers.
- Chapter 11, "Streaming Protocols Overview," discusses HTTP and real streaming, the use of TCP and UDP in streaming, analog and digital video, coders-decoders (codecs), packetization, the streaming transport formats, unicast, multicast and stream splitting, and encoding mechanisms.

Part III, "Infrastructure Protocols," explores the fundamental Layer 2 and Layer 3 protocols as well as IBM Data Center technologies:

- Chapter 12, "Layer 2 Protocol Essentials," discusses Ethernet frame types; the difference between unicast, multicast, and broadcast frames; physical layer characteristics of Ethernet technologies; jumbo frames; trunks and channels; and a variety of spanning-tree concepts. Chapter 20 provides the design best practices applied to the concepts described in this chapter.
- Chapter 13, "Layer 3 Protocol Essentials," discusses the Address Resolution Protocol (ARP); gateway redundancy protocols such as Hot Standby Router Protocol (HSRP), VRRP and GLBP; and routing-protocol essentials for Open Shortest Path First (OSPF) and Enhanced Interior Gateway Routing Protocol (EIGRP). Chapter 20 provides the design best practices applied to the concepts described in this chapter.

- Chapter 14, "IBM Data Center Technology," discusses mainframe attachment options, IBM networking, Systems Network Architecture (SNA) switching, Sysplex, TN3270, and current IBM Data Center designs.

Part IV, "Security and Server Load Balancing," explores the security protocols and technology, load-balancing operations, server health management, session tracking and cookies, and persistence mechanisms on load balancers:

- Chapter 15, "Security Protocols and Technologies," discusses cryptography, U.S. government–related topics about cryptography, PKI, transport security protocols (SSL and IP Security [IPSec]), authentication protocols and technologies, and network management security. This chapter also complements Chapter 9 with regards to the security design aspects of SSL and introduces the concept of SSL VPNs.
- Chapter 16, "Load-Balancing Modes and Predictors," discusses the load-balancing modes of operation, server load-balancing algorithms, and cache farm load-balancing algorithms.
- Chapter 17, "Server Health Management," discusses server health management through load balancers, SNMP, server failure detection and checking, in-band and out-of-band probes, and case studies on server checking for web hosting and e-commerce applications.
- Chapter 18, "Session Tracking and Cookies," explores the concept of user sessions from an application point of view. This chapter explains nonpersistent cookies, cookies in general, how servers track user sessions, session persistence on clusters of servers, and the challenges of dealing with HTTP and HTTPS. Chapter 19 further expands the topic of session persistence in load-balancing deployments.
- Chapter 19, "Persistence Mechanisms on Load Balancers," explains session persistence in relation to load balancing; discusses key persistence mechanisms, including source-IP sticky, cookie-URL sticky, HTTP redirection sticky, and SSL sticky; and presents a case study using an e-commerce application. Chapter 19 is based on the applications introduced in Chapter 3 and Chapter 18.

Part V, "Data Center Design," explores the details behind designing the Data Center infrastructure, the integration of security into the infrastructure design, and the performance of Data Center devices:

- Chapter 20, "Designing the Data Center Infrastructure," discusses router switching paths, essential Data Center design concepts, and the design best practices of the infrastructure by explaining the configuration of Layer 2 and Layer 3 features and protocols that are described in Chapter 12 and 13.
- Chapter 21, "Integrating Security into the Infrastructure," discusses the concept of security zones and how to design application security at the Internet Edge and at intranet server farms. This chapter explains alternative designs and how to implement secure management.
- Chapter 22, "Performance Metrics of Data Center Devices," discusses the Data Center traffic patterns and performance metrics of various Data Center devices, including proposed metrics for devices for which there are none and no standard methodology exists (such as load balancers and SSL offloaders).

Part VI, "Appendixes," is the final part of this book:

- Appendix A, "Character Sets," covers multiple character sets, including ASCII, the extended ASCII sets, and the ISO-8859-1 set.
- Appendix B, "HTTP Header Fields," explains the details of HTTP header fields that were not described in Chapter 8.
- Appendix C, "Video Encoding Mechanisms," explains the removal of special and temporal redundancy in codecs with special focus on MPEG.
- Appendix D, "Loopback Interface Configuration Procedures," provides an explanation about configuring a machine with multiple IP addresses used as loopbacks for certain load-balancing modes of operation.

- Appendix E, “Configuring Servers to Insert Cookies,” examines several alternatives for configuring cookie insertion on web servers.
- Appendix F, “Client-Side and Server-Side Programming,” provides excerpts of client-side programs to help you understand the differences and similarities between JavaScripts, Java applets, and ActiveX controls. The section on server-side programming explains the differences between CGI, servlets, and Active Server Pages (ASP) in terms of operating-system implications (threads versus processes). This appendix explains the adoption of certain technologies in today’s enterprise applications and the performance and availability implications.

This chapter covers the following topics:

- Types of server farms and Data Centers
- Data Center topologies
- Fully redundant Layer 2 and Layer 3 designs
- Fully redundant Layer 2 and Layer 3 designs with services

CHAPTER 4

Data Center Design Overview

This chapter focuses on three main properties of Data Center architectures: scalability, flexibility, and high availability. Data Centers are rapidly evolving to accommodate higher expectations for growth, consolidation, and security. Although the traditional Layer 2 and Layer 3 designs have not changed drastically over the last few years, stringent demands for uptime and service availability, coupled with new technology and protocols, make the design efforts more challenging and demanding.

Demands for scalability, flexibility, and high availability can be summarized as follows:

- **Scalability**—The Data Center must support fast and seamless growth without major disruptions.
- **Flexibility**—The Data Center must support new services without a major overhaul of its infrastructure.
- **High availability**—The Data Center must have no single point of failure and should offer predictable uptime (related to hard failures).

NOTE A hard failure is a failure in which the component must be replaced to return to an operational steady state.

Scalability translates into the capability to sustain rapid growth in performance, the number of devices hosted in the Data Center, and the amount and quality of the services offered. Higher performance implies tolerance to very short-term changes in traffic patterns without packet loss and longer-term plans mapping growth trends to the capacity of the Data Center.

Scalability on the number of hosted devices refers to being capable of seamlessly adding more ports for servers, routers, switches, and any other service devices, such as server load balancers, firewalls, IDSs, and SSL offloaders. Higher density also includes slot density because the number of slots ultimately determines the potential growth of the system.

Flexibility translates into designs that accommodate new service offerings without requiring the complete redesign of the architecture or drastic changes outside the normal periods scheduled for maintenance. The approach to flexibility is a modular design in which the characteristics of the modules are known, and the steps to add more modules are simple.

High availability translates into a fully redundant architecture in which all possible hard failures are predictable and deterministic. This implies that each possible component's failure has a predetermined failover and fallback time, and that the worst-case scenario for a failure condition is still within the acceptable failover limits and is within the requirements as measured from an application availability viewpoint. This means that although the time of failure and recovery of a network component should be predictable and known, the more important time involves the user's perception of the time to recover application service.

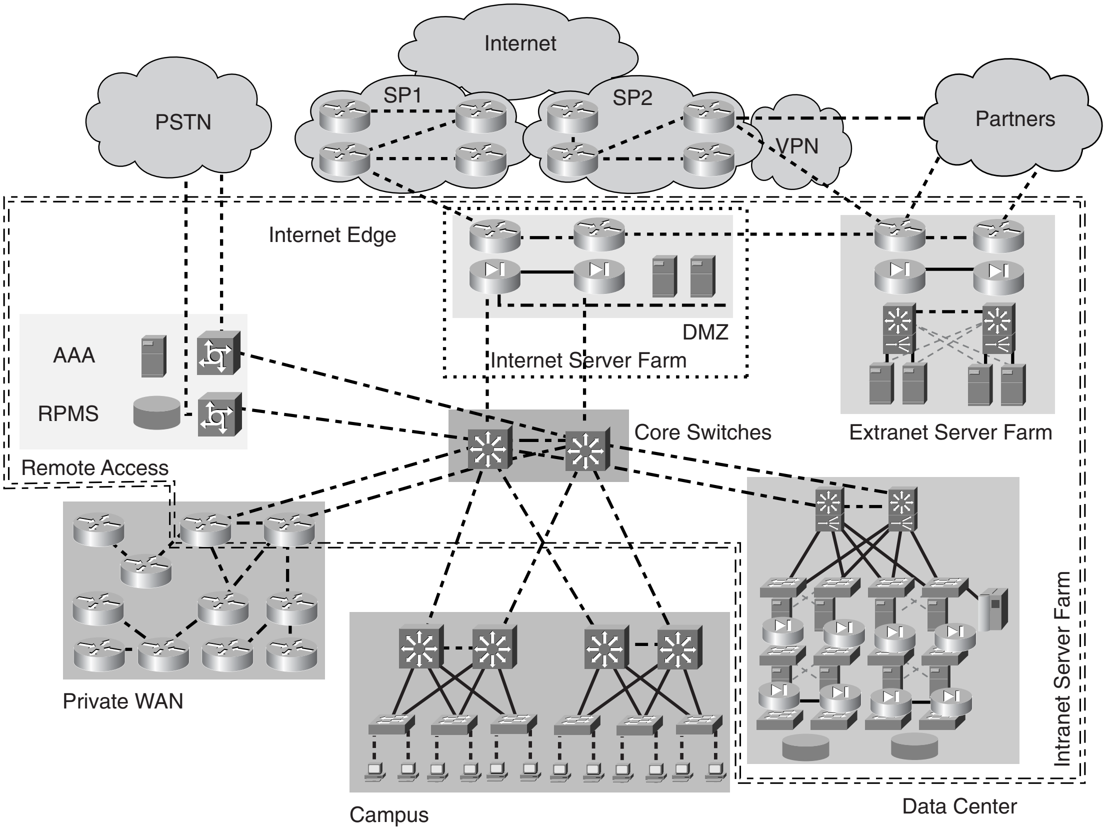
NOTE After a failure, the recovery time could be measured from the perspective of the Layer 2 environment (the spanning tree) or from a Layer 3 perspective (the routed network), yet the application availability ultimately matters to the user. If the failure is such that the user connection times out, then, regardless of the convergence time, the network convergence does not satisfy the application requirements. In a Data Center design, it is important to measure recovery time from the perspectives of both the network and the application to ensure a predictable network recovery time for the user (application service).

Figure 4-1 presents an overview of the Data Center, which, as a facility, includes a number of the building blocks and components of the larger enterprise network architecture.

This books deals primarily with the engineering of application environments and their integration to the remaining enterprise network. Different types of server farms support the application environments, yet this book focuses on understanding, designing, deploying, and maintaining the server farms supporting intranet application environments. The actual engineering of the different server farm types—Internet, extranet, and intranet server farms—does not vary much from type to type; however, their integration with the rest of the architecture is different. The design choices that differ for each type of server farm are the result of their main functional purpose. This leads to a specific location for their placement, security considerations, redundancy, scalability, and performance. In addition to the server farm concepts, a brief discussion on the types of server farms further clarifies these points.

NOTE The figures in this chapter contain a wide variety of Cisco icons. Refer to the section, "Icons Used in This Book" (just before the "Introduction") for a list of icons and their descriptions.

Figure 4-1 *Overview of Data Center Topology*



Types of Server Farms and Data Centers

As depicted in Figure 4-1, three distinct types of server farms exist:

- Internet
- Extranet
- Intranet

All three types reside in a Data Center and often in the same Data Center facility, which generally is referred to as the *corporate Data Center* or *enterprise Data Center*. If the sole purpose of the Data Center is to support Internet-facing applications and server farms, the Data Center is referred to as an *Internet Data Center*.

Server farms are at the heart of the Data Center. In fact, Data Centers are built to support at least one type of server farm. Although different types of server farms share many architectural requirements, their objectives differ. Thus, the particular set of Data Center requirements depends on which type of server farm must be supported. Each type of server farm has a distinct set of infrastructure, security, and management requirements that must be addressed in the design of the server farm. Although each server farm design and its specific topology might be different, the design guidelines apply equally to them all. The following sections introduce server farms.

Internet Server Farms

As their name indicates, Internet server farms face the Internet. This implies that users accessing the server farms primarily are located somewhere on the Internet and use the Internet to reach the server farm. Internet server farms are then available to the Internet community at large and support business-to-consumer services. Typically, internal users also have access to the Internet server farms. The server farm services and their users rely on the use of web interfaces and web browsers, which makes them pervasive on Internet environments.

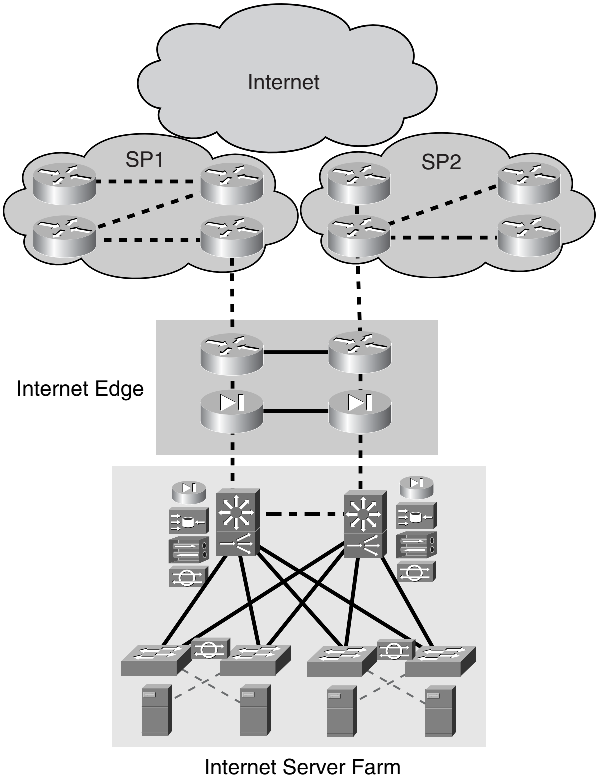
Two distinct types of Internet server farms exist. The dedicated Internet server farm, shown in Figure 4-2, is built to support large-scale Internet-facing applications that support the core business function. Typically, the core business function is based on an Internet presence or Internet commerce.

In general, dedicated Internet server farms exist to sustain the enterprise's e-business goals. Architecturally, these server farms follow the Data Center architecture introduced in Chapter 1, "Overview of Data Centers," yet the details of each layer and the necessary layers are determined by the application environment requirements. Security and scalability are a major concern in this type of server farm. On one hand, most users accessing the server farm are located on the Internet, thereby introducing higher security risks; on the other hand, the number of likely users is very high, which could easily cause scalability problems.

The Data Center that supports this type of server farm is often referred to as an Internet Data Center (IDC). IDCs are built both by enterprises to support their own e-business infrastructure and by service providers selling hosting services, thus allowing enterprises to collocate the e-business infrastructure in the provider's network.

The next type of Internet server farm, shown in Figure 4-3, is built to support Internet-based applications in addition to Internet access from the enterprise. This means that the infrastructure supporting the server farms also is used to support Internet access from enterprise users. These server farms typically are located in the demilitarized zone (DMZ) because they are part of the enterprise network yet are accessible from the Internet. These server farms are referred to as DMZ server farms, to differentiate them from the dedicated Internet server farms.

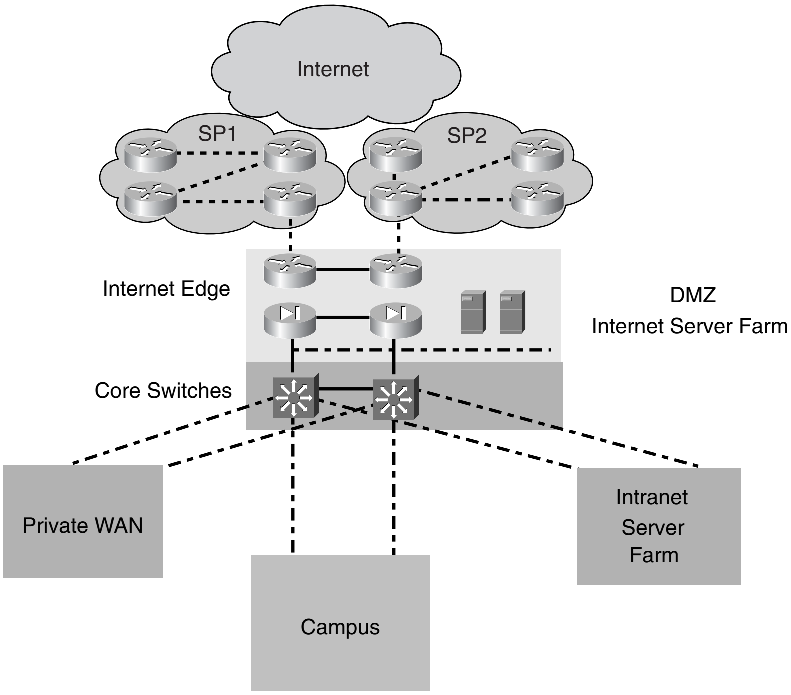
Figure 4-2 *Dedicated Internet Server Farms*



These server farms support services such as e-commerce and are the access door to portals for more generic applications used by both Internet and intranet users. The scalability considerations depend on how large the expected user base is. Security requirements are also very stringent because the security policies are aimed at protecting the server farms from external users while keeping the enterprise's network safe. Note that, under this model, the enterprise network supports the campus, the private WAN, and the intranet server farm.

NOTE Notice that Figure 4-3 depicts a small number of servers located on a segment off the firewalls. Depending on the requirements, the small number of servers could become hundreds or thousands, which would change the topology to include a set of Layer 3 switches and as many Layers 2 switches for server connectivity as needed.

Figure 4-3 *DMZ Server Farms*

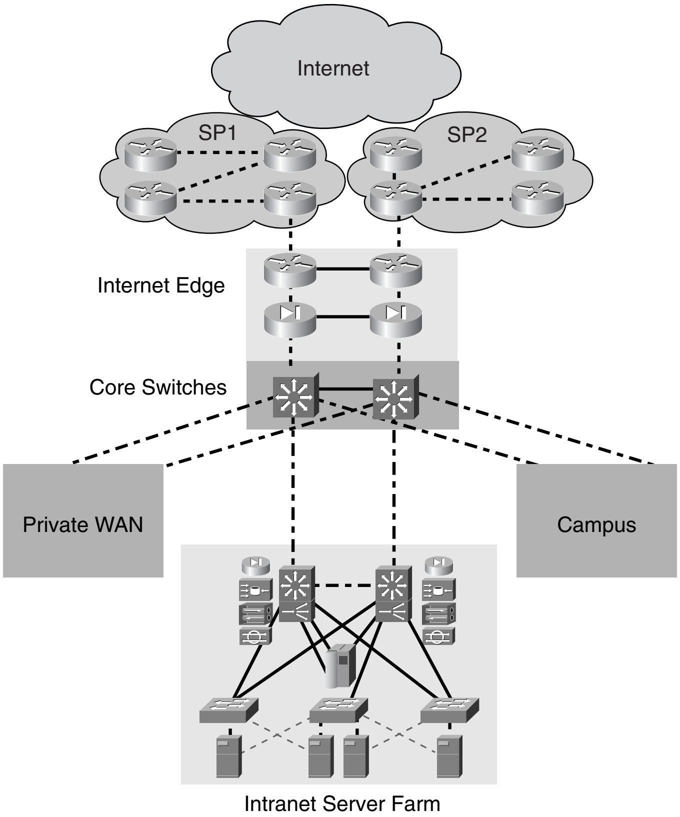


Intranet Server Farms

The evolution of the client/server model and the wide adoption of web-based applications on the Internet was the foundation for building intranets. Intranet server farms resemble the Internet server farms in their ease of access, yet they are available only to the enterprise's internal users. As described earlier in this chapter, intranet server farms include most of the enterprise-critical computing resources that support business processes and internal applications. This list of critical resources includes midrange and mainframe systems that support a wide variety of applications. Figure 4-4 illustrates the intranet server farm.

Notice that the intranet server farm module is connected to the core switches that form a portion of the enterprise backbone and provide connectivity between the private WAN and Internet Edge modules. The users accessing the intranet server farm are located in the campus and private WAN. Internet users typically are not permitted access to the intranet; however, internal users using the Internet as transport have access to the intranet using virtual private network (VPN) technology.

Figure 4-4 *Intranet Server Farms*



The Internet Edge module supports several functions that include the following:

- Securing the enterprise network
- Controlling Internet access from the intranet
- Controlling access to the Internet server farms

The Data Center provides additional security to further protect the data in the intranet server farm. This is accomplished by applying the security policies to the edge of the Data Center as well as to the applicable application tiers when attempting to harden communication between servers on different tiers. The security design applied to each tier depends on the architecture of the applications and the desired security level.

The access requirements of enterprise users dictate the size and architecture of the server farms. The growing number of users, as well as the higher load imposed by rich applications, increases the demand placed on the server farm. This demand forces scalability to become a critical design criterion, along with high availability, security, and management.

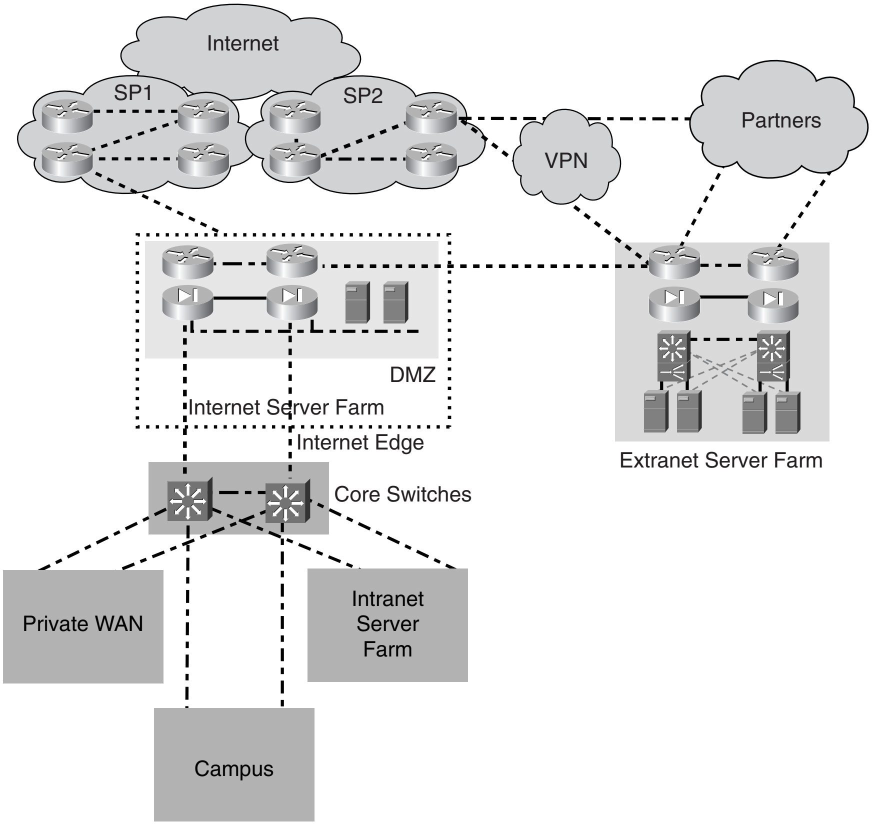
Extranet Server Farms

From a functional perspective, extranet server farms sit between Internet and intranet server farms. Extranet server farms continue the trend of using web-based applications, but, unlike Internet- or intranet-based server farms, they are accessed only by a selected group of users that are neither Internet- nor intranet-based. Extranet server farms are mainly available to business partners that are considered external yet trusted users. The main purpose for extranets is to improve business-to-business communication by allowing faster exchange of information in a user-friendly and secure environment. This reduces time to market and the cost of conducting business. The communication between the enterprise and its business partners, traditionally supported by dedicated links, rapidly is being migrated to a VPN infrastructure because of the ease of the setup, lower costs, and the support for concurrent voice, video, and data traffic over an IP network.

As explained previously, the concept of extranet is analogous to the IDC, in that the server farm is at the edge of the enterprise network. Because the purpose of the extranet is to provide server farm services to trusted external end users, there are special security considerations. These security considerations imply that the business partners have access to a subset of the business applications but are restricted from accessing the rest of the enterprise network. Figure 4-5 shows the extranet server farm. Notice that the extranet server farm is accessible to internal users, yet access from the extranet to the intranet is prevented or highly secured. Typically, access from the extranet to the intranet is restricted through the use of firewalls.

Many factors must be considered in the design of the extranet topology, including scalability, availability, and security. Dedicated firewalls and routers in the extranet are the result of a highly secure and scalable network infrastructure for partner connectivity, yet if there are only a small number of partners to deal with, you can leverage the existing Internet Edge infrastructure. Some partners require direct connectivity or dedicated private links, and others expect secure connections through VPN links. The architecture of the server farm does not change whether you are designing Internet or intranet server farms. The design guidelines apply equally to all types of server farms, yet the specifics of the design are dictated by the application environment requirements.

Figure 4-5 *Extranet Server Farms*



The following section discusses the types of Data Centers briefly mentioned in this section.

Internet Data Center

Internet Data Centers (IDCs) traditionally are built and operated by service providers, yet enterprises whose business model is based on Internet commerce also build and operate IDCs. The architecture of enterprise IDCs is very similar to that of the service provider IDCs, but the requirements for scalability are typically lower because the user base tends to be smaller and there are fewer services compared with those of SP IDCs hosting multiple customers.

In fact, the architecture of the IDC is the same as that presented in Figure 4-2. An interesting consideration of enterprise IDCs is that if the business model calls for it, the facilities used by the Data Center could be collocated in a service provider Data Center, but it remains under the control of the enterprise. This typically is done to lower the costs associated with building the server farm and reducing a product's time to market by avoiding building a Data Center internally from the ground up.

Corporate Data Center

Corporate or enterprise Data Centers support many different functions that enable various business models based on Internet services, intranet services, or both. As a result, support for Internet, intranet, and extranet server farms is not uncommon. This concept was depicted in Figure 4-1, where the Data Center facility supports every type of server farm and also is connected to the rest of the enterprise network—private WAN, campus, Internet Edge, and so on. The support of intranet server farms is still the primary target of corporate Data Centers.

Enterprise Data Centers are evolving, and this evolution is partly a result of new trends in application environments, such as the n-tier, web services, and grid computing, but it results mainly because of the criticality of the data held in Data Centers.

The following section discusses the typical topologies used in the architecture of the Data Center.

Data Center Topologies

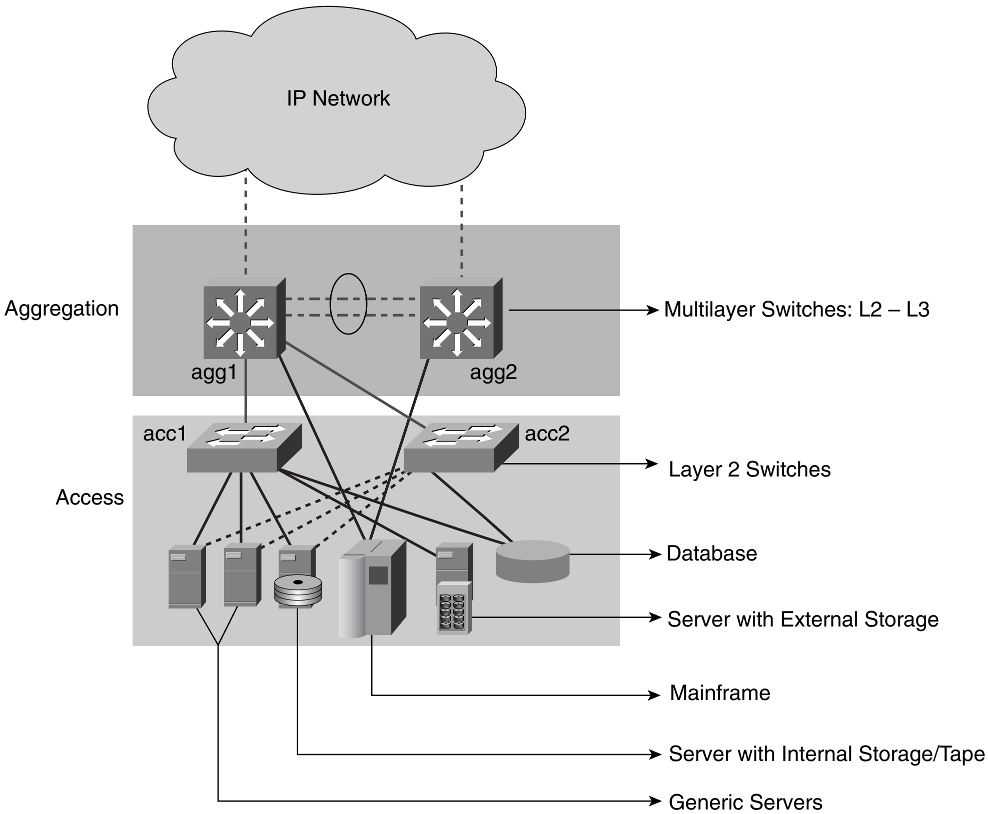
This section discusses Data Center topologies and, in particular, the server farm topology. Initially, the discussion focuses on the traffic flow through the network infrastructure (on a generic topology) from a logical viewpoint and then from a physical viewpoint.

Generic Layer 3/Layer 2 Designs

The generic Layer 3/Layer 2 designs are based on the most common ways of deploying server farms. Figure 4-6 depicts a generic server farm topology that supports a number of servers.

NOTE Notice that the distribution layer now is referred to as the aggregation layer resulting from becoming the aggregation point for most, if not all, services beyond the traditional Layer 2 and Layer 3.

Figure 4-6 *Generic Server Farm Design*



The highlights of the topology are the aggregation-layer switches that perform key Layer 3 and Layer 2 functions, the access-layer switches that provide connectivity to the servers in the server farm, and the connectivity between the aggregation and access layer switches.

The key Layer 3 functions performed by the aggregation switches are as follows:

- Forwarding packets based on Layer 3 information between the server farm and the rest of the network
- Maintaining a "view" of the routed network that is expected to change dynamically as network changes take place
- Supporting default gateways for the server farms

The key Layer 2 functions performed by the aggregation switches are as follows:

- Spanning Tree Protocol (STP) 802.1d between aggregation and access switches to build a loop-free forwarding topology.
- STP enhancements beyond 802.1d that improve the default spanning-tree behavior, such as 802.1s, 802.1w, Uplinkfast, Backbonefast, and Loopguard. For more information, refer to Chapter 12, “Layer 2 Protocol Essentials.”
- VLANs for logical separation of server farms.
- Other services, such as multicast and ACLs for services such as QoS, security, rate limiting, broadcast suppression, and so on.

The access-layer switches provide direct connectivity to the server farm. The types of servers in the server farm include generic servers such as DNS, DHCP, FTP, and Telnet; mainframes using SNA over IP or IP; and database servers. Notice that some servers have both internal disks (storage) and tape units, and others have the storage externally connected (typically SCSI).

The connectivity between the two aggregation switches and between aggregation and access switches is as follows:

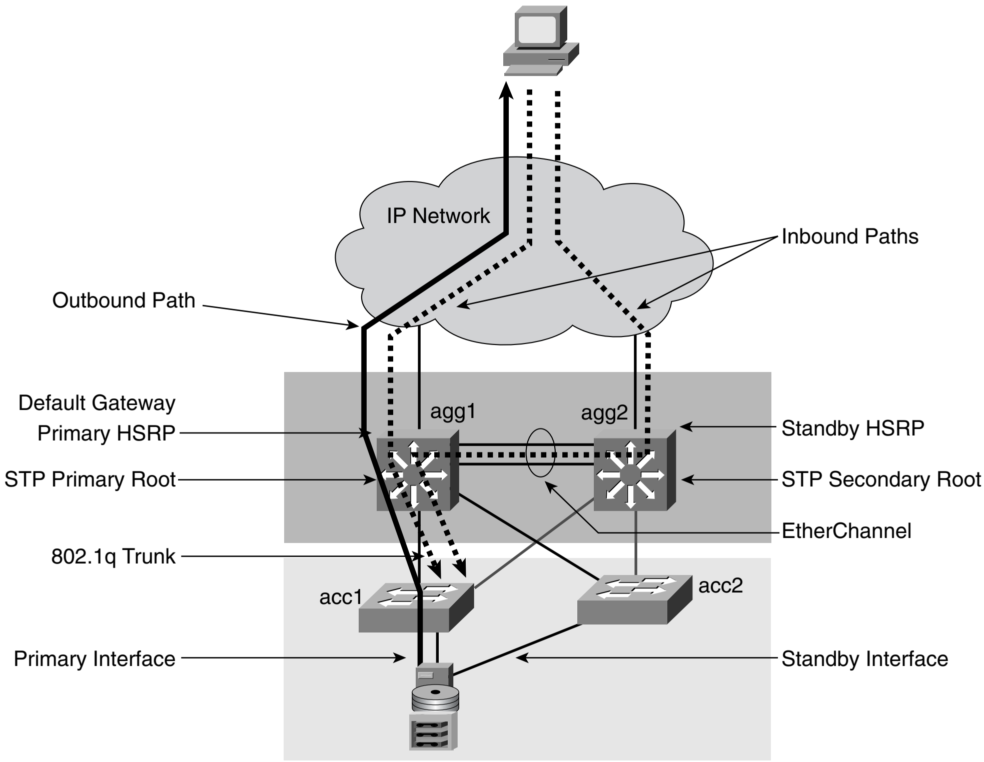
- EtherChannel between aggregation switches. The channel is in trunk mode, which allows the physical links to support as many VLANs as needed (limited to 4096 VLANs resulting from the 12-bit VLAN ID).
- Single or multiple links (EtherChannel, depending on how much oversubscription is expected in the links) from each access switch to each aggregation switch (uplinks). These links are also trunks, thus allowing multiple VLANs through a single physical path.
- Servers dual-homed to different access switches for redundancy. The NIC used by the server is presumed to have two ports in an active-standby configuration. When the primary port fails, the standby takes over, utilizing the same MAC and IP addresses that the active port was using. For more information about dual-homed servers, refer to Chapter 2, “Server Architecture Overview.”

The typical configuration for the server farm environment just described is presented in Figure 4-7.

Figure 4-7 shows the location for the critical services required by the server farm. These services are explicitly configured as follows:

- agg1 is explicitly configured as the STP root.
- agg2 is explicitly configured as the secondary root.
- agg1 is explicitly configured as the primary default gateway.
- agg2 is explicitly configured as the standby or secondary default gateway.

Figure 4-7 *Common Server Farm Environment*



NOTE The explicit definition of these critical functions sets the primary and alternate paths to and from the server farm. Notice that there is no single point of failure in the architecture, and the paths are now deterministic.

Other STP services or protocols, such as UplinkFast, are also explicitly defined between the aggregation and access layers. These services/protocols are used to lower convergence time during failover conditions from the 802.d standard of roughly 50 seconds to 1 to 3 seconds.

In this topology, the servers are configured to use the agg1 switch as the primary default gateway, which means that outbound traffic from the servers follows the direct path to the agg1 switch. Inbound traffic can arrive at either aggregation switch, yet the traffic can reach the server farm only through agg1 because the links from agg2 to the access switches are not forwarding (blocking). The inbound paths are represented by the dotted arrows, and the outbound path is represented by the solid arrow.

The next step is to have predictable failover and fallback behavior, which is much simpler when you have deterministic primary and alternate paths. This is achieved by failing every component in the primary path and recording and tuning the failover time to the backup component until the requirements are satisfied. The same process must be done for falling back to the original primary device. This is because the failover and fallback processes are not the same. In certain instances, the fallback can be done manually instead of automatically, to prevent certain undesirable conditions.

NOTE When using 802.1d. if the primary STP root fails and the secondary takes over, when it comes back up, it automatically takes over because it has a lower priority. In an active server farm environment, you might not want to have the STP topology change automatically, particularly when the convergence time is in the range of 50 seconds. However, this behavior is not applicable when using 802.1w, in which the fallback process takes only a few seconds.

Whether using 802.1d or 802.1w, the process is automatic, unlike when using HSRP, in which the user can control the behavior of the primary HSRP peer when it becomes operational again through the use of preemption. If preemption is not used, the user has manual control over when to return mastership to the initial master HSRP peer.

The use of STP is the result of a Layer 2 topology, which might have loops that require an automatic mechanism to be detected and avoided. An important question is whether there is a need for Layer 2 in a server farm environment. This topic is discussed in the following section.

For more information about the details of the Layer 2 design, see Chapter 20, “Designing the Data Center Infrastructure.”

The Need for Layer 2 at the Access Layer

Access switches traditionally have been Layer 2 switches. This holds true also for the campus network wiring closet. This discussion is focused strictly on the Data Center because it has distinct and specific requirements, some similar to and some different than those for the wiring closets.

The reason access switches in the Data Center traditionally have been Layer 2 is the result of the following requirements:

- When they share specific properties, servers typically are grouped on the same VLAN. These properties could be as simple as ownership by the same department or performance of the same function (file and print services, FTP, and so on). Some servers that perform the same function might need to communicate with one another, whether as a result of a clustering protocol or simply as part of the application function. This communication exchange should be on the same subnet and sometimes is possible only on the same subnet if the clustering protocol heartbeats or the server-to-server application packets are not routable.
- Servers are typically dual-homed so that each leg connects to a different access switch for redundancy. If the adapter in use has a standby interface that uses the same MAC and IP addresses after a failure, the active and standby interfaces must be on the same VLAN (same default gateway).
- Server farm growth occurs horizontally, which means that new servers are added to the same VLANs or IP subnets where other servers that perform the same functions are located. If the Layer 2 switches hosting the servers run out of ports, the same VLANs or subnets must be supported on a new set of Layer 2 switches. This allows flexibility in growth and prevents having to connect two access switches.
- When using stateful devices that provide services to the server farms, such as load balancers and firewalls, these stateful devices expect to see both the inbound and outbound traffic use the same path. They also need to constantly exchange connection and session state information, which requires Layer 2 adjacency. More details on these requirements are discussed in the section, "Access Layer," which is under the section, "Multiple Tier Designs."

Using just Layer 3 at the access layer would prevent dual-homing, Layer 2 adjacency between servers on different access switches, and Layer 2 adjacency between service devices. Yet if these requirements are not common on your server farm, you could consider a Layer 3 environment in the access layer. Before you decide what is best, it is important that you read the section titled "Fully Redundant Layer 2 and Layer 3 Designs with Services," later in the chapter. New service trends impose a new set of requirements in the architecture that must be considered before deciding which strategy works best for your Data Center.

The reasons for migrating away from a Layer 2 access switch design are motivated by the need to drift away from spanning tree because of the slow convergence time and the operation challenges of running a controlled loopless topology and troubleshooting loops when they occur. Although this is true when using 802.1d, environments that take advantage of 802.1w combined with Loopguard have the following characteristics: They do not suffer from the same problems, they are as stable as Layer 3 environments, and they support low convergence times.

NOTE The STP standard 802.1d has limitations in addressing certain conditions in addition to its convergence time, yet a fair amount of spanning tree–related problems are the result of misconfiguration or rogue STP devices that appear on the network and "bridge" between Layer 2 domains. More information on this topic is presented in Chapter 12.

The next section discusses an alternate solution for a topology with spanning tree that does not present the STP problems or limitations.

Alternate Layer 3/Layer 2 Designs

Figure 4-8 presents an alternate Layer 3/Layer 2 design resulting from the need to address STP limitations.

Figure 4-8 *Loopless Topology*

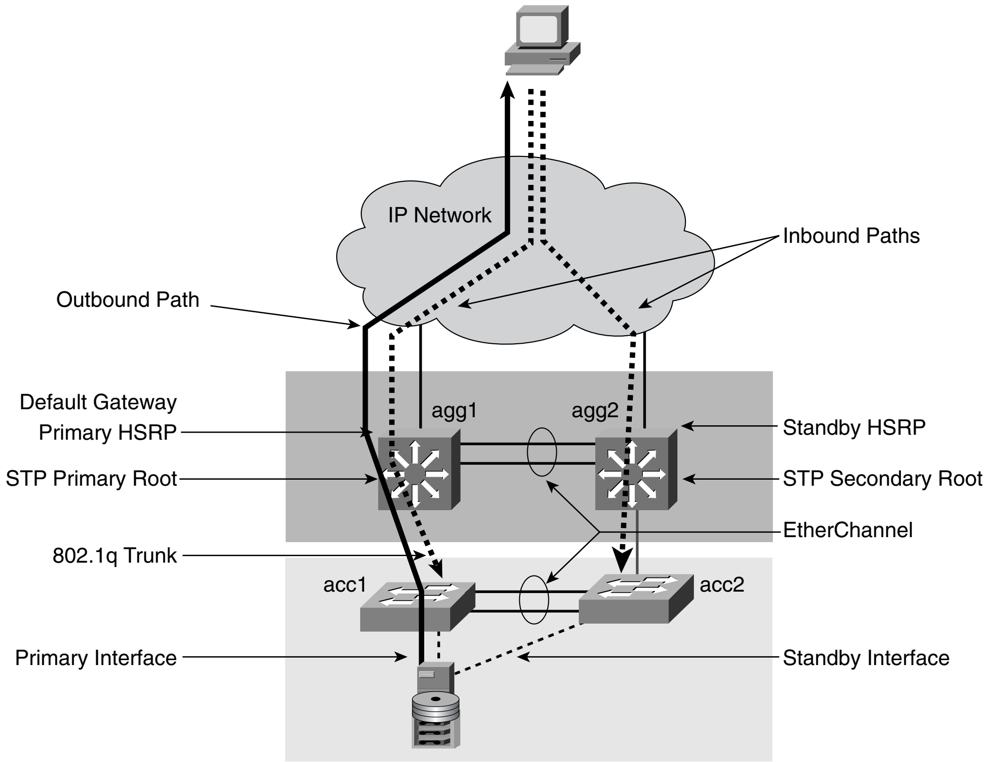


Figure 4-8 presents a topology in which the network purposely is designed not to have loops. Although STP is running, its limitations do not present a problem. This loopless topology is accomplished by removing or not allowing the VLAN(s), used at the access-layer switches, through the trunk between the two aggregation switches. This basically prevents a loop in the topology while it supports the requirements behind the need for Layer 2.

In this topology, the servers are configured to use the agg1 switch as the primary default gateway. This means that outbound traffic from the servers connected to acc2 traverses the link between the two access switches. Inbound traffic can use either aggregation switch because both have active (nonblocking) paths to the access switches. The inbound paths are represented by the dotted arrows, and the outbound path is represented by the solid arrows.

This topology is not without its own challenges. These challenges are discussed later in the chapter after other information related to the deployment of services becomes available.

Multiple-Tier Designs

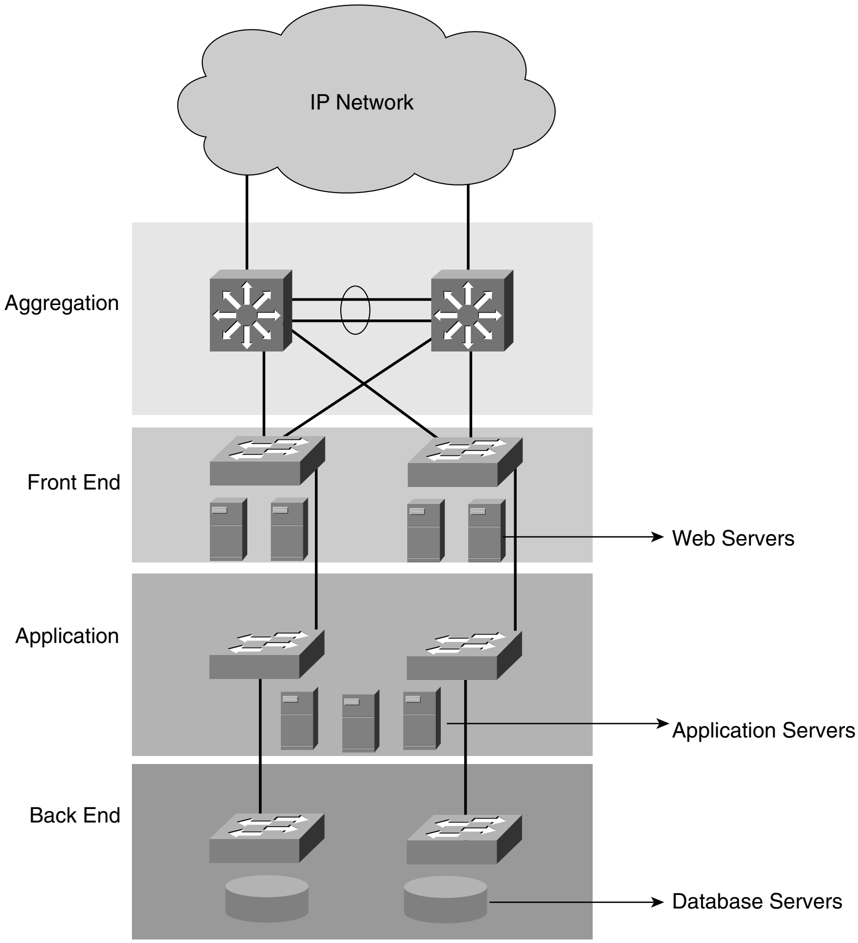
Most applications conform to either the client/server model or the n-tier model, which implies most networks, and server farms support these application environments. The tiers supported by the Data Center infrastructure are driven by the specific applications and could be any combination in the spectrum of applications from the client/server to the client/web server/application server/database server. When you identify the communication requirements between tiers, you can determine the needed specific network services. The communication requirements between tiers are typically higher scalability, performance, and security. These could translate to load balancing between tiers for scalability and performance, or SSL between tiers for encrypted transactions, or simply firewalling and intrusion detection between the web and application tier for more security.

Figure 4-9 introduces a topology that helps illustrate the previous discussion.

Notice that Figure 4-9 is a logical diagram that depicts layer-to-layer connectivity through the network infrastructure. This implies that the actual physical topology might be different. The separation between layers simply shows that the different server functions could be physically separated. The physical separation could be a design preference or the result of specific requirements that address communication between tiers.

For example, when dealing with web servers, the most common problem is scaling the web tier to serve many concurrent users. This translates into deploying more web servers that have similar characteristics and the same content so that user requests can be equally fulfilled by any of them. This, in turn, requires the use of a load balancer in front of the server farm that hides the number of servers and virtualizes their services. To the users, the specific service is still supported on a single server, yet the load balancer dynamically picks a server to fulfill the request.

Figure 4-9 *Multiple-Tier Application Environments*



Suppose that you have multiple types of web servers supporting different applications, and some of these applications follow the n-tier model. The server farm could be partitioned along the lines of applications or functions. All web servers, regardless of the application(s) they support, could be part of the same server farm on the same subnet, and the application servers could be part of a separate server farm on a different subnet and different VLAN.

Following the same logic used to scale the web tier, a load balancer logically could be placed between the web tier and the application tier to scale the application tier from the web tier perspective. A single web server now has multiple application servers to access.

The same set of arguments holds true for the need for security at the web tier and a separate set of security considerations at the application tier. This implies that firewall and intrusion-detection capabilities are distinct at each layer and, therefore, are customized for the requirements of the application and the database tiers. SSL offloading is another example of a function that the server farm infrastructure might support and can be deployed at the web tier, the application tier, and the database tier. However, its use depends upon the application environment using SSL to encrypt client-to-server and server-to-server traffic.

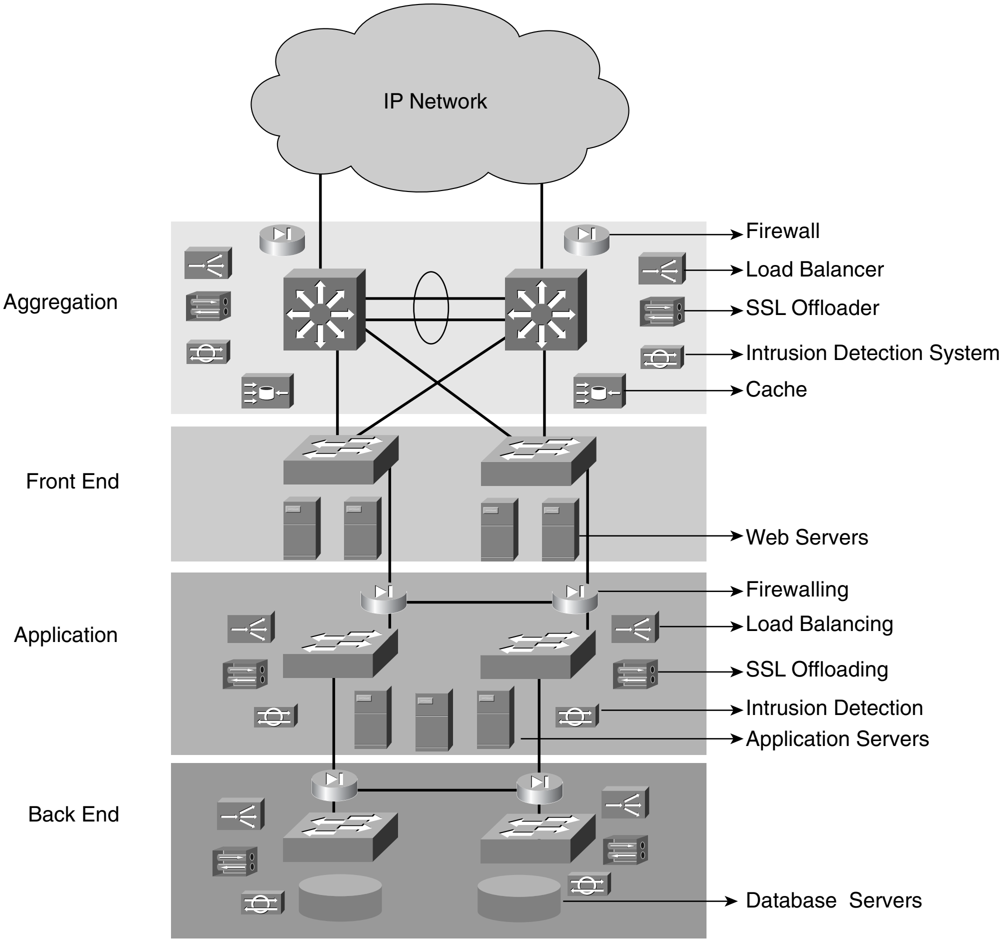
Expanded Multitier Design

The previous discussion leads to the concept of deploying multiple network-based services in the architecture. These services are introduced in Figure 4-10 through the use of icons that depict the function or service performed by the network device.

NOTE Figure 4-10 introduces the icons used through this chapter to depict the services provided by network devices in the Data Center.

The different icons are placed in front of the servers for which they perform the functions. At the aggregation layer, you find the load balancer, firewall, SSL offloader, intrusion-detection system, and cache. These services are available through service modules (line cards that could be inserted into the aggregation switch) or appliances. An important point to consider when dealing with service devices is that they provide scalability and high availability beyond the capacity of the server farm, and that to maintain the basic premise of "no single point of failure," at least two must be deployed. If you have more than one (and considering you are dealing with redundancy of application environments), the failover and fallback processes require special mechanisms to recover the connection context, in addition to the Layer 2 and Layer 3 paths. This simple concept of redundancy at the application layer has profound implications in the network design.

Figure 4-10 *Network Service Icons*



A number of these network service devices are replicated in front of the application layer to provide services to the application servers. Notice in Figure 4-10 that there is physical separation between the tiers of servers. This separation is one alternative to the server farm design. Physical separation is used to achieve greater control over the deployment and scalability of services. The expanded design is more costly because it uses more devices, yet it allows for more control and better scalability because the devices in the path handle only a portion of the traffic. For example, placing a firewall between tiers is regarded as a more secure approach because of the physical separation between the Layer 2 switches.

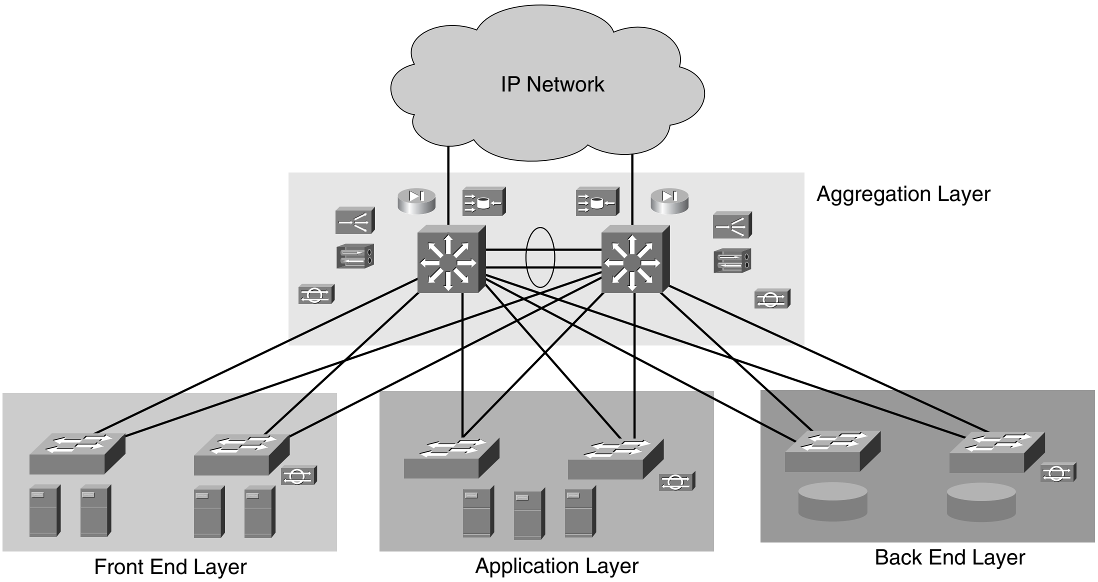
This argument is correct, yet it is likely to be much more related to an existing security policy than a real threat. Having logical instead of physical separation simply requires a consistent application of security policies to ensure that the expanded security zone is as secure logically as it is physically.

This brings the discussion to another alternative of designing the multitier server farm, an alternative in which there is no physical separation, but rather a logical separation between tiers, as presented in the next section.

Collapsed Multitier Design

A collapsed multitier design is one in which all the server farms are directly connected at the access layer to the aggregation switches, and there is no physical separation between the Layer 2 switches that support the different tiers. Figure 4-11 presents the collapsed design.

Figure 4-11 *Collapsed Multiple-Tier Design*



Notice that in this design, the services again are concentrated at the aggregation layer, and the service devices now are used by the front-end tier and between tiers. Using a collapsed model, there is no need to have a set of load balancers or SSL offloaders dedicated to a particular tier. This reduces cost, yet the management of devices is more challenging and the performance demands are higher. The service devices, such as the firewalls, protect all server tiers from outside the Data Center, but also from each other. The load balancer also can be used concurrently to load-balance traffic from client to web servers, and traffic from web servers to application servers.

Notice that the design in Figure 4-11 shows each type of server farm on a different set of switches. Other collapsed designs might combine the same physical Layer 2 switches to house web applications and database servers concurrently. This implies merely that the servers logically are located on different IP subnets and VLANs, yet the service devices still are used concurrently for the front end and between tiers. Notice that the service devices are always in pairs. Pairing avoids the single point of failure throughout the architecture. However, both service devices in the pair communicate with each other, which falls into the discussion of whether you need Layer 2 or Layer 3 at the access layer.

The Need for Layer 2 at the Access Layer

Each pair of service devices must maintain state information about the connections the pair is handling. This requires a mechanism to determine the active device (master) and another mechanism to exchange connection state information on a regular basis. The goal of the dual–service device configuration is to ensure that, upon failure, the redundant device not only can continue service without interruption, but also seamlessly can failover without disrupting the current established connections.

In addition to the requirements brought up earlier about the need for Layer 2, this section discusses in depth the set of requirements related to the service devices:

- Service devices and the server farms that they serve are typically Layer 2–adjacent. This means that the service device has a leg sitting on the same subnet and VLAN used by the servers, which is used to communicate directly with them. Often, in fact, the service devices themselves provide default gateway support for the server farm.
- Service devices must exchange heartbeats as part of their redundancy protocol. The heartbeat packets might or might not be routable; if they are routable, you might not want the exchange to go through unnecessary Layer 3 hops.
- Service devices operating in stateful failover need to exchange connection and session state information. For the most part, this exchange is done over a VLAN common to the two devices. Much like the heartbeat packets, they might or might not be routable.
- If the service devices provide default gateway support for the server farm, they must be adjacent to the servers.

After considering all the requirements for Layer 2 at the access layer, it is important to note that although it is possible to have topologies such as the one presented in Figure 4-8, which supports Layer 2 in the access layer, the topology depicted in Figure 4-7 is preferred. Topologies with loops are also supportable if they take advantages of protocols such as 802.1w and features such as Loopguard.

NOTE To date, most common implementations use Layer 2 at the access layer and rely on the Spanning Tree Protocols and Cisco enhancements to lower convergence times and achieve stability, as depicted in Figure 4-7. Few use the loopless topology. The main reasons relate to whether it is possible to have a loopless topology, given the restrictions imposed by the requirements, and, if possible, whether the setup is simple enough for support, maintenance, and management reasons. Dual-homing requires Layer 2 adjacency between access switches to carry the same VLANs, and redundant stateful service devices need Layer 2 adjacency to work properly. Therefore, it is important to carefully consider the requirements when designing the server farm network infrastructure.

The following section discusses topics related to the topology of the server farms.

Fully Redundant Layer 2 and Layer 3 Designs

Up to this point, all the topologies that have been presented are fully redundant. This section explains the various aspects of a redundant and scalable Data Center design by presenting multiple possible design alternatives, highlighting sound practices, and pointing out practices to be avoided.

The Need for Redundancy

Figure 4-12 explains the steps in building a redundant topology.

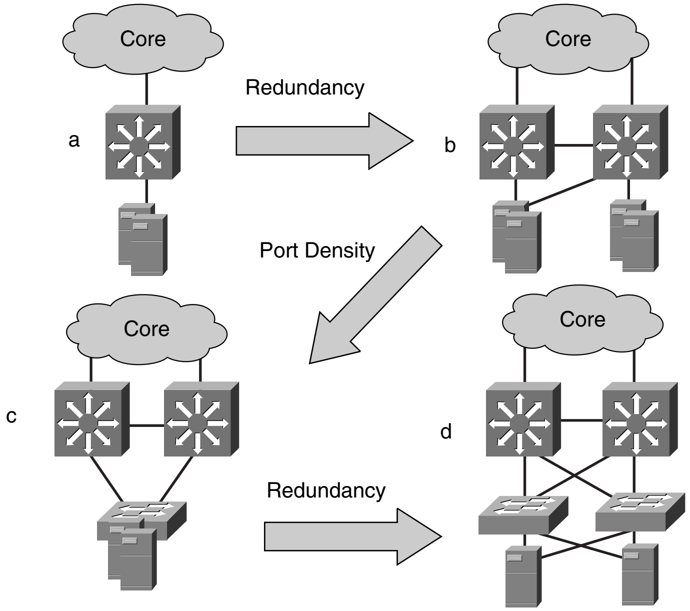
Figure 4-12 depicts the logical steps in designing the server farm infrastructure. The process starts with a Layer 3 switch that provides ports for direct server connectivity and routing to the core. A Layer 2 switch could be used, but the Layer 3 switch limits the broadcasts and flooding to and from the server farms. This is option **a** in Figure 4-12. The main problem with the design labeled **a** is that there are multiple single point of failure problems: There is a single NIC and a single switch, and if the NIC or switch fails, the server and applications become unavailable.

The solution is twofold:

- Make the components of the single switch redundant, such as dual power supplies and dual supervisors.
- Add a second switch.

Redundant components make the single switch more tolerant, yet if the switch fails, the server farm is unavailable. Option **b** shows the next step, in which a redundant Layer 3 switch is added.

Figure 4-12 *Multilayer Redundant Design*



By having two Layer 3 switches and spreading servers on both of them, you achieve a higher level of redundancy in which the failure of one Layer 3 switch does not completely compromise the application environment. The environment is not completely compromised when the servers are dual-homed, so if one of the Layer 3 switches fails, the servers still can recover by using the connection to the second switch.

In options **a** and **b**, the port density is limited to the capacity of the two switches. As the demands for more ports increase for the server and other service devices, and when the maximum capacity has been reached, adding new ports becomes cumbersome, particularly when trying to maintain Layer 2 adjacency between servers.

The mechanism used to grow the server farm is presented in option **c**. You add Layer 2 access switches to the topology to provide direct server connectivity. Figure 4-12 depicts the Layer 2 switches connected to both Layer 3 aggregation switches. The two uplinks, one to each aggregation switch, provide redundancy from the access to the aggregation switches, giving the server farm an alternate path to reach the Layer 3 switches.

The design described in option **c** still has a problem. If the Layer 2 switch fails, the servers lose their only means of communication. The solution is to dual-home servers to two different Layer 2 switches, as depicted in option **d** of Figure 4-12.

NOTE Throughout this book, the terms *access layer* and *access switches* refer to the switches used to provide port density. The terms *aggregation layer* and *aggregation switches* refer to the switches used both to aggregate the traffic to and from the access switches and to connect service devices (load balancers, SSL offloaders, firewalls, caches, and so on).

The *aggregation switches* are Layer 3 switches, which means that they have a built-in router that can forward traffic at wire speed.

The *access switches* are predominantly Layer 2 switches, yet they could be Layer 3 switches merely operating in Layer 2 mode for the server farms.

Layer 2 and Layer 3 in Access Layer

Option **d** in Figure 4-12 is detailed in option **a** of Figure 4-13.

Figure 4-13 *Layer 3 and Layer 2 in the Data Center*

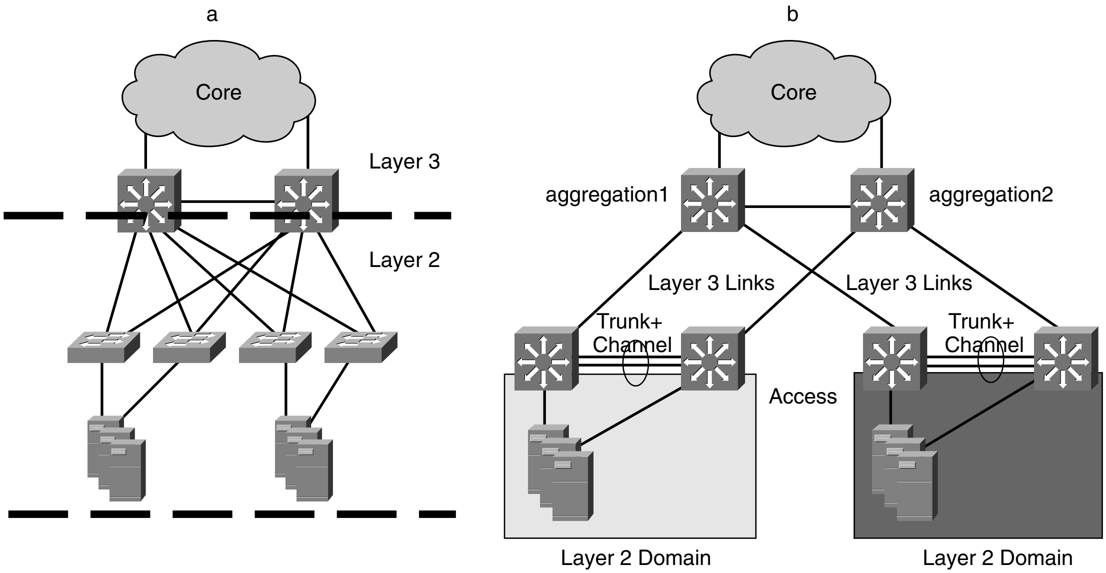


Figure 4-13 presents the scope of the Layer 2 domain(s) from the servers to the aggregation switches. Redundancy in the Layer 2 domain is achieved mainly by using spanning tree, whereas in Layer 3, redundancy is achieved through the use of routing protocols.

Historically, routing protocols have proven more stable than spanning tree, which makes one question the wisdom of using Layer 2 instead of Layer 3 at the access layer. This topic was discussed previously in the "Need for Layer 2 at the Access Layer" section. As shown in option **b** in Figure 4-13, using Layer 2 at the access layer does not prevent the building of pure Layer 3 designs because of the routing between the access and distribution layer or the supporting Layer 2 between access switches.

The design depicted in option **a** of Figure 4-13 is the most generic design that provides redundancy, scalability, and flexibility. Flexibility relates to the fact that the design makes it easy to add service appliances at the aggregation layer with minimal changes to the rest of the design. A simpler design such as that depicted in option **b** of Figure 4-13 might better suit the requirements of a small server farm.

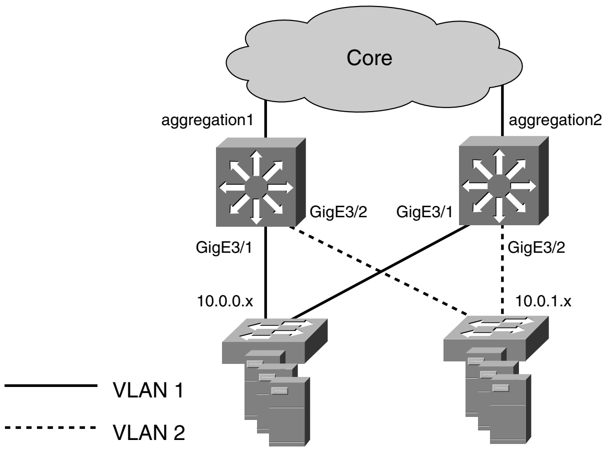
Layer 2, Loops, and Spanning Tree

The Layer 2 domains should make you think immediately of loops. Every network designer has experienced Layer 2 loops in the network. When Layer 2 loops occur, packets are replicated an infinite number of times, bringing down the network. Under normal conditions, the Spanning Tree Protocol keeps the logical topology free of loops. Unfortunately, physical failures such as unidirectional links, incorrect wiring, rogue bridging devices, or bugs can cause loops to occur.

Fortunately, the introduction of 802.1w has addressed many of the limitations of the original spanning tree algorithm, and features such as Loopguard fix the issue of malfunctioning transceivers or bugs.

Still, the experience of deploying legacy spanning tree drives network designers to try to design the Layer 2 topology free of loops. In the Data Center, this is sometimes possible. An example of this type of design is depicted in Figure 4-14. As you can see, the Layer 2 domain (VLAN) that hosts the subnet 10.0.0.*x* is not trunked between the two aggregation switches, and neither is 10.0.1.*x*. Notice that GigE3/1 and GigE3/2 are not bridged together.

Figure 4-14 *Loop-Free Layer 2 Design*



TIP It is possible to build a loop-free access layer if you manage to keep subnets specific to a single access switch. If subnets must span multiple access switches, you should have a "looped" topology. This is the case when you have dual-attached servers because NIC cards configured for "teaming" typically use a floating IP and MAC address, which means that both interfaces belong to the same subnet.

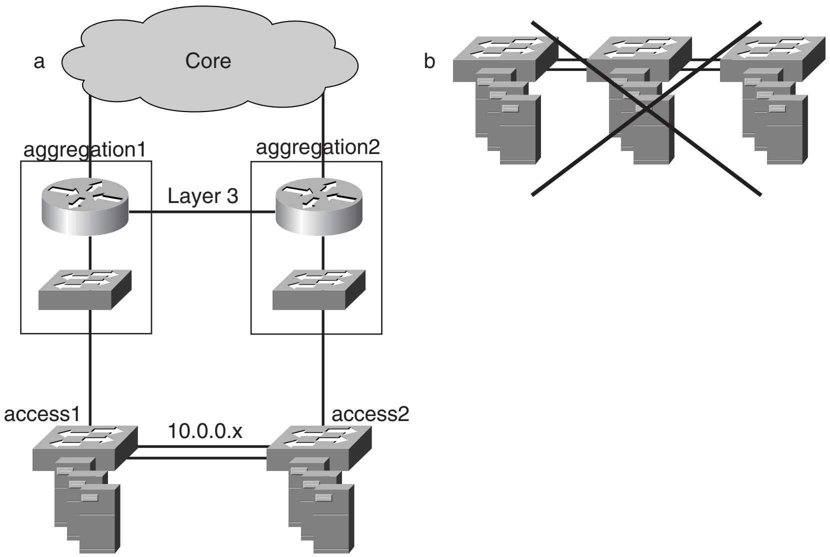
Keep in mind that a "loop-free" topology is not necessarily better. Specific requirements such as those mandated by content switches actually might require the additional path provided by a "looped" topology.

Also notice that a "looped" topology simply means that any Layer 2 device can reach any other Layer 2 device from at least two different physical paths. This does not mean that you have a "forwarding loop," in which packets are replicated infinite times: Spanning tree prevents this from happening.

In a "looped" topology, malfunctioning switches can cause Layer 2 loops. In a loop-free topology, there is no chance for a Layer 2 loop because there are no redundant Layer 2 paths.

If the number of ports must increase for any reason (dual-attached servers, more servers, and so forth), you could follow the approach of daisy-chaining Layer 2 switches, as shown in Figure 4-15.

Figure 4-15 *Alternate Loop-Free Layer 2 Design*



To help you visualize a Layer 2 loop-free topology, Figure 4-15 shows each aggregation switch broken up as a router and a Layer 2 switch.

The problem with topology **a** is that breaking the links between the two access switches would create a discontinuous subnet—this problem can be fixed with an EtherChannel between the access switches.

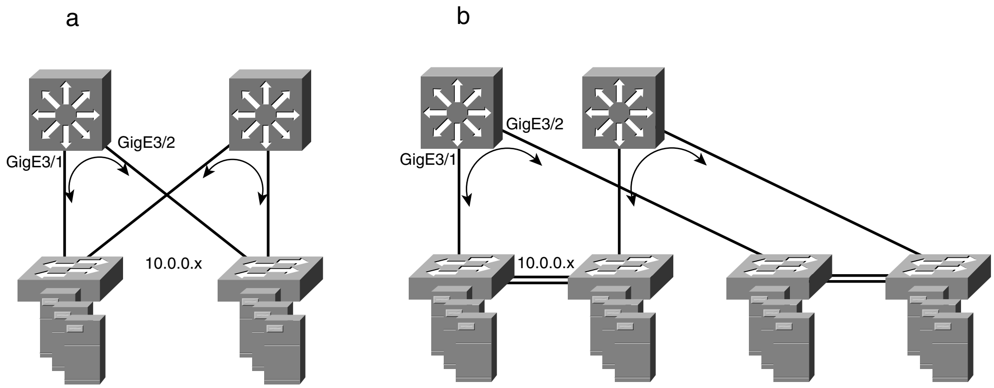
The other problem occurs when there are not enough ports for servers. If a number of servers need to be inserted into the same subnet 10.0.0.*x*, you cannot add a switch between the two existing servers, as presented in option **b** of Figure 4-15. This is because there is no workaround to the failure of the middle switch, which would create a split subnet. This design is not intrinsically wrong, but it is not optimal.

Both the topologies depicted in Figures 4-14 and 4-15 should migrate to a looped topology as soon as you have any of the following requirements:

- An increase in the number of servers on a given subnet
- Dual-attached NIC cards
- The spread of existing servers for a given subnet on a number of different access switches
- The insertion of stateful network service devices (such as load balancers) that operate in active/standby mode

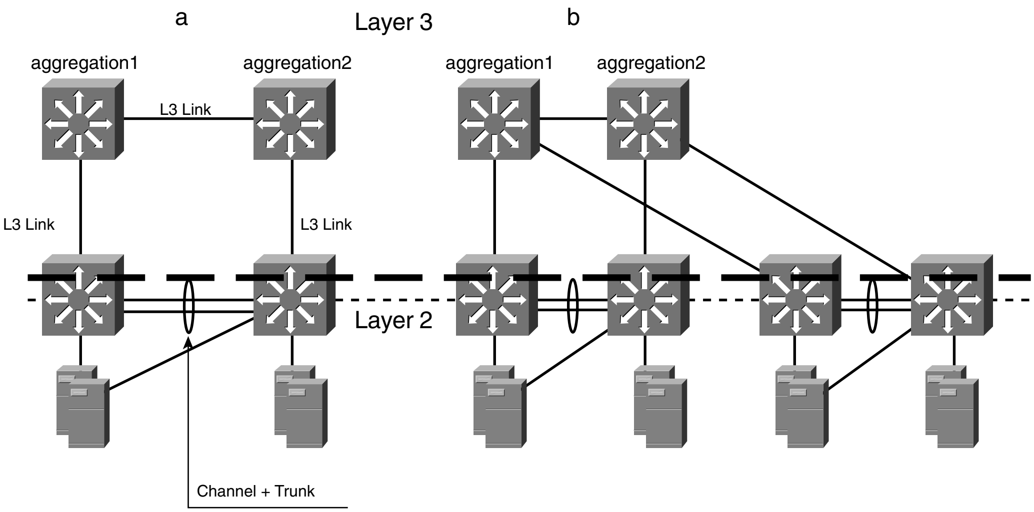
Options **a** and **b** in Figure 4-16 show how introducing additional access switches on the existing subnet creates "looped topologies." In both **a** and **b**, GigE3/1 and GigE3/2 are bridged together.

Figure 4-16 *Redundant Topologies with Physical Layer 2 Loops*



If the requirement is to implement a topology that brings Layer 3 to the access layer, the topology that addresses the requirements of dual-attached servers is pictured in Figure 4-17.

Figure 4-17 *Redundant Topology with Layer 3 to the Access Switches*



Notice in option **a** of Figure 4-17, almost all the links are Layer 3 links, whereas the access switches have a trunk (on a channel) to provide the same subnet on two different switches. This trunk also carries a Layer 3 VLAN, which basically is used merely to make the two switches neighbors from a routing point of view. The dashed line in Figure 4-17 shows the scope of the Layer 2 domain.

Option **b** in Figure 4-17 shows how to grow the size of the server farm with this type of design. Notice that when deploying pairs of access switches, each pair has a set of subnets disjointed from the subnets of any other pair. For example, one pair of access switches hosts subnets 10.0.1.*x* and 10.0.2.*x*; the other pair cannot host the same subnets simply because it connects to the aggregation layer with Layer 3 links.

NOTE If you compare the design in Figure 4-17 with option **b** in Figure 4-12, the natural questions are these: Why is there an aggregation layer, and are the access switches not directly connected to the core? These are valid points, and the answer actually depends on the size of the Data Center. Remember that the access layer is added for reasons of port density, whereas the aggregation layer is used mainly to attach appliances, such as load-balancing devices, firewalls, caches, and so on.

So far, the discussions have centered on redundant Layer 2 and Layer 3 designs. The Layer 3 switch provides the default gateway for the server farms in all the topologies introduced thus far. Default gateway support, however, could also be provided by other service devices, such as load balancers and firewalls. The next section explores the alternatives.

Fully Redundant Layer 2 and Layer 3 Designs with Services

After discussing the build-out of a fully redundant Layer 2 and Layer 3 topology and considering the foundation of the Data Center, the focus becomes the design issues related to other Data Center services. These services are aimed at improving security and scaling the performance of application services by offloading processing away from the server farm to the network. These services include security, load balancing, SSL offloading, and caching; they are supported by a number of networking devices that must be integrated into the infrastructure following the design requirements.

Additionally, this section discusses application environment trends brought about by technology advancements in either applications, the application infrastructure, or the network infrastructure.

Additional Services

At the aggregation layer, in addition to Layer 2 and Layer 3, the Data Center might need to support the following devices:

- Firewalls
- Intrusion Detection Systems (IDSs)
- Load balancers
- SSL offloaders
- Caches

It is important to discuss design issues when supporting some of these devices.

Service devices bring their own requirements that could change certain aspects of the design—for instance, the exchange state or status information, the NAT function that they perform on the source or destination IP addresses that forces them to be in the inbound and outbound path, and so on.

Service devices can be deployed using service modules integrated in the aggregation switches or as appliances connected to the aggregation switches. Both deployments require network connectivity and forethought about the actual traffic path.

Firewalls and load balancers may support the default gateway function on behalf of the server farms. Default gateway support traditionally has been provided by the router, so with two additional alternatives, you need to decide which is the default gateway and in which order traffic is processed through the multiple devices. Firewalls and load balancers are capable of providing stateful failover, which is supported by specific redundancy protocols. The protocols, which are specific to the firewalls or load balancers, must be supported by the design. SSL offloaders are typically used with load balancers and require the same considerations, with one exception: They do not support default gateway services.

IDSs are transparent to the design, which means that they integrate well with any existing design. The main consideration with regard to IDSs is their placement, which depends on selecting the location to analyze traffic and the traffic types to be monitored.

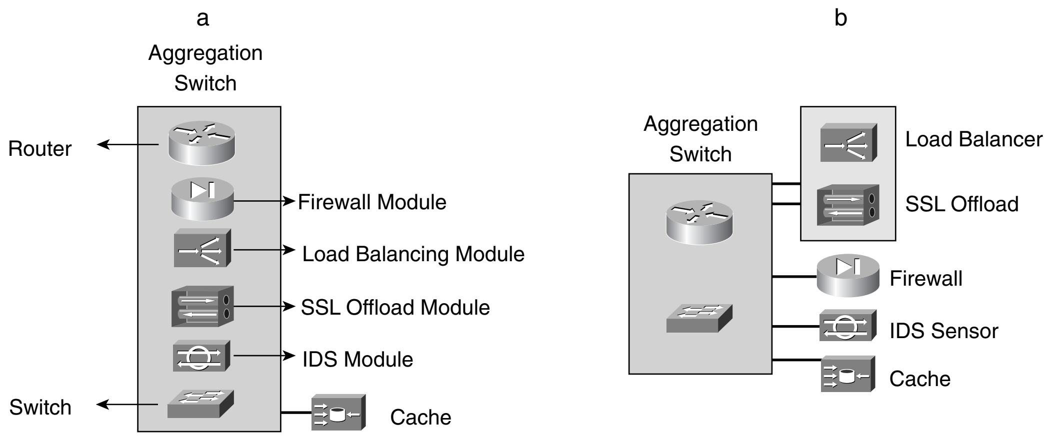
Caches, on the other hand, are deployed in reverse proxy cache mode. The placement of the caches and the mechanism for directing traffic to them impact the Data Center design. The options for traffic redirection are the Web Cache Communication Protocol (WCCP) on the Layer 2 or Layer 3 switches, and load balancers to distribute the load among the cache cluster. In either case, the cache or cache cluster changes the basic traffic path to the server farm when in use.

The following section presents the multiple deployment options.

Service Deployment Options

Two options exist when deploying Data Center services: using service modules integrated into the aggregation switch and using appliances connected to the aggregation switch. Figure 4-18 shows the two options.

Figure 4-18 *Service Deployment Options*



Option **a** shows the integrated design. The aggregation switch is represented by a router (Layer 3) and a switch (Layer 2) as the key components of the foundation (shown to the left) and by a firewall, load balancer, SSL module, and IDS module (shown to the right as add-on services). The service modules communicate with the routing and switching components in the chassis through the backplane.

Option **b** shows the appliance-based design. The aggregation switch provides the routing and switching functions. Other services are provided by appliances that are connected directly to the aggregation switches.

NOTE Designs that use both modules and appliances are also possible. The most common case is when using caches, which are appliances, in both design options. Current trends on Data Center services lean toward integrated services. Evidence of this integration trend is the proliferation of services modules in the Catalyst 6500 family and the use of blade servers and blade chassis to collapse multiple services in one device.

A thoughtful approach to the design issues in selecting the traffic flow across different devices is required whether you are considering option **a**, option **b**, or any combination of the options in Figure 4-18. This means that you should explicitly select the default gateway and the order in which the packets from the client to the server are processed. The designs that use appliances require more care because you must be concerned with physical connectivity issues, interoperability, and the compatibility of protocols.

Design Considerations with Service Devices

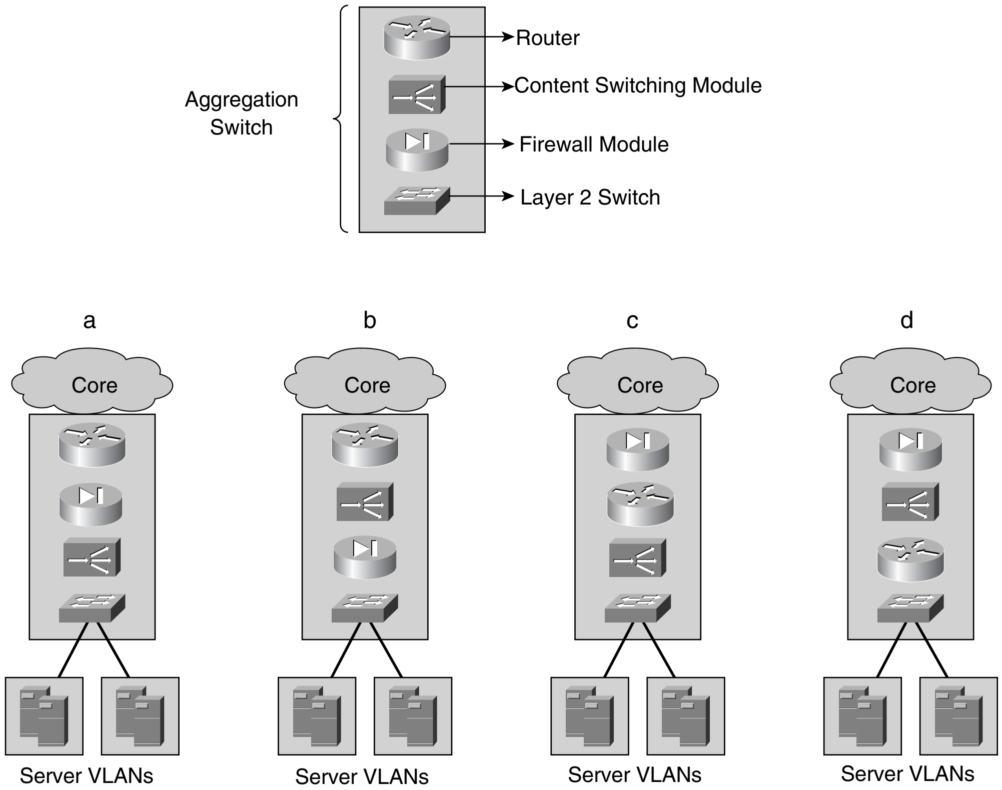
Up to this point, several issues related to integrating service devices in the Data Center design have been mentioned. They are related to whether you run Layer 2 or Layer 3 at the access layer, whether you use appliance or modules, whether they are stateful or stateless, and whether they require you to change the default gateway location away from the router. Changing the default gateway location forces you to determine the order in which the packet needs to be processed through the aggregation switch and service devices.

Figure 4-19 presents the possible alternatives for default gateway support using service modules. The design implications of each alternative are discussed next.

Figure 4-19 shows the aggregation switch, a Catalyst 6500 using a firewall service module, and a content-switching module, in addition to the routing and switching functions provided by the Multilayer Switch Feature Card (MSFC) and the Supervisor Module.

The one constant factor in the design is the location of the switch providing server connectivity; it is adjacent to the server farm.

Figure 4-19 *Service Module Interoperability Alternatives*



Option **a** presents the router facing the core IP network, the content-switching module facing the server farm, and the firewall module between them firewalling all server farms. If the content switch operates as a router (route mode), it becomes the default gateway for the server farm. However, if it operates as a bridge (bridge mode), the default gateway would be the firewall. This configuration facilitates the creation of multiple instances of the firewall and content switch combination for the segregation and load balancing of each server farm independently.

Option **b** has the firewall facing the server farm and the content switch between the router and the firewall. Whether operating in router mode or bridge mode, the firewall configuration must enable server health-management (health probes) traffic from the content-switching module to the server farm; this adds management and configuration tasks to the design. Note that, in this design, the firewall provides the default gateway support for the server farm.

Option **c** shows the firewall facing the core IP network, the content switch facing the server farm, and the firewall module between the router and the content-switching module. Placing a firewall at the edge of the intranet server farms requires the firewall to have "router-like" routing capabilities, to ease the integration with the routed network while segregating all the server farms concurrently. This makes the capability to secure each server farm independently more difficult because the content switch and the router could route packets between the server farm without going through the firewall. Depending on whether the content-switching module operates in router or bridge mode, the default gateway could be the content switch or the router, respectively.

Option **d** displays the firewall module facing the core IP network, the router facing the server farm, and the content-switching module in between. This option presents some of the same challenges as option **c** in terms of the firewall supporting IGPs and the inability to segregate each server farm independently. The design, however, has one key advantage: The router is the default gateway for the server farm. Using the router as the default gateway allows the server farms to take advantage of some key protocols, such as HSRP, and features, such as HSRP tracking, QoS, the DHCP relay function, and so on, that are only available on routers.

All the previous design options are possible—some are more flexible, some are more secure, and some are more complex. The choice should be based on knowing the requirements as well as the advantages and restrictions of each. The different design issues associated with the viable options are discussed in the different chapters in Part V. Chapter 21, "Integrating Security into the Infrastructure," addresses the network design in the context of firewalls.

Application Environment Trends

Undoubtedly, the most critical trends are those related to how applications are being developed and are expected to work on the network. These trends can be classified arbitrarily into two major areas:

- Application architectures
- Network infrastructure

Application Architecture Trends

Application architecture trends include the evolution of the classic client/server model to the more specialized n-tier model, web services, specific application architectures, the server and client software (operating systems), application clients, the server and client hardware, and middleware used to integrate distributed applications in heterogeneous environments.

The more visible trends of application architectures are the wide adoption of web technology in conjunction with the use of the n-tier model to functionally segment distinct server types. Currently, web, application, and database servers are the basic types, yet they are combined in many ways (depending on the vendor of the application and how the buyer wants to implement it).

This functional partitioning demands that the network be smarter about securing and scaling the tiers independently. For instance, the n-tier model's web tier layer created the need for smaller and faster servers used to scale up the front-end function. This resulted in 1RU (rack unit) servers, which offer adequate performance for web servers at a low cost and minimal infrastructure requirements (power and rack space).

Web services are bringing a service-oriented approach to the use of different and distinct distributed applications that are accessible using standard messages over Internet protocols. Web services rely initially on the transport functions of the network and eventually on using the network as an extension to provide computing capacity to the distributed application environments by offloading tasks to network hardware.

NOTE The World Wide Web consortium (W3C) defines a web service as "a software application identified by a URI, whose interfaces and binding are capable of being defined, described, and discovered by XML artifacts and supports direct interactions with other software applications using XML-based messages via Internet-based protocols." For more information on web services and its architecture, consult the W3C at www.w3.org.

Grid computing is another trend that actually brings the applications and the network closer together by treating the servers as a network of CPUs in which the applications use the most available CPU on the network. Other trends related to grid computing include blade servers as an alternative to 1RU servers, to provide higher CPU density per RU, lower power consumption per server, and an additional benefit of lower cabling requirements. Blade servers are servers on blades (or modules) that are inserted into a chassis, much like network modules or line cards are inserted on a switch chassis. Using blade servers in blade chassis enables you to centralize the server-management functions (one chassis instead of however many servers are in the chassis), requires less cables (one set per chassis instead of one set per server), and provides higher computing and memory capacity per rack unit.

However, the blade server technology is still young, which explains the variety of flavors, architectures, connectivity options, and features.

An instance of middleware is the software used in the management and control of distributed CPUs in a grid of computers that can be 1RU or blade servers. This specific middleware virtualizes the use of CPUs so that the applications are given a CPU cycle from CPUs on the network instead of through the traditional manner.

Network Infrastructure Trends

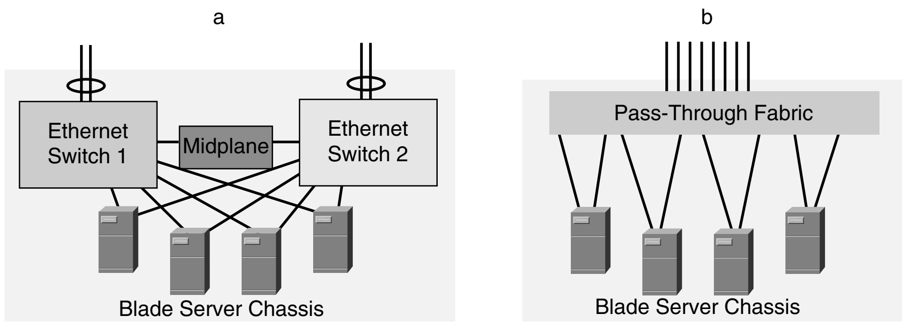
The network infrastructure is growing smarter and more application-aware, and it thereby supports application environments both by offloading some computationally intense tasks to the network (typically hardware-based) and by replacing some functions performed by servers that could be better handled by networking devices.

Load balancing is a good example of a function performed by the network that replaces clustering protocols used by servers for high availability. Clustering protocols tend to be software-based, hard to manage, and not very scalable in providing a function that the network performs well using hardware.

Trends such as blade servers bring new design considerations. Most blade server chassis (blade chassis, for short) in the market support both an option to provide redundant Ethernet switches inside the chassis and as an option to connect the blade servers to the network using pass-through links, with the chassis simply providing at least twice as many uplinks as servers in the chassis, to allow dual-homing.

Figure 4-20 presents both connectivity alternatives for a blade chassis.

Figure 4-20 *Blade Server Chassis Server Connectivity*



Option **a** in Figure 4-20 shows a blade server chassis in which each blade server is connected to each of the blade chassis's redundant Layer 2 Ethernet switches. Each blade chassis's Ethernet switch provides a number of uplinks that can be channeled to the IP network. The number of uplinks is typically smaller than the combined number of links per server, which requires planning for oversubscription, particularly if the servers are Gigabit Ethernet–attached. The midplane is the fabric used for management tasks, that is, control plane traffic such as switch status.

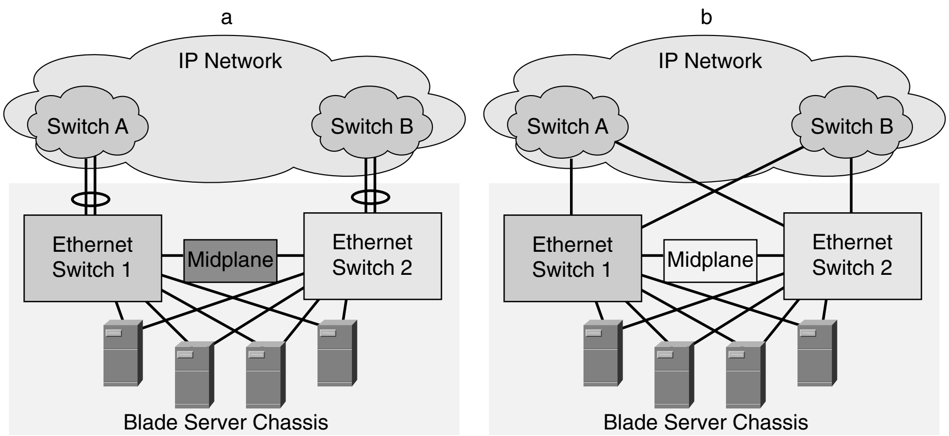
Option **b** in Figure 4-20 presents the pass-through option in which the servers are dual-homed and preconnected to a pass-through fabric that provides the connectivity to the IP network. This option does not use Ethernet switches inside the chassis. The pass-through fabric is as simple as a patch panel that conserves the properties of the server NICs, but it also could become a more intelligent fabric, adding new features and allowing blade server vendors to differentiate their products. Either approach you take to connect blade servers to your network requires careful consideration on short- and long-term design implications.

For instance, if the choice is to utilize the redundant Ethernet switches in the blade chassis, you have the following design alternatives to consider:

- How to use the redundant Ethernet switches' uplinks for connectivity
- Whether to connect the blade chassis to the access or aggregation switches
- What level of oversubscription is tolerable

Figure 4-21 displays two connectivity choices utilizing the uplinks on the redundant Ethernet switches. For redundancy, two switches are used to connect the uplinks from the blade chassis. Switches A and B, the small clouds in the IP network cloud, provide a redundant network fabric to the blade chassis to avoid single point of failure issues.

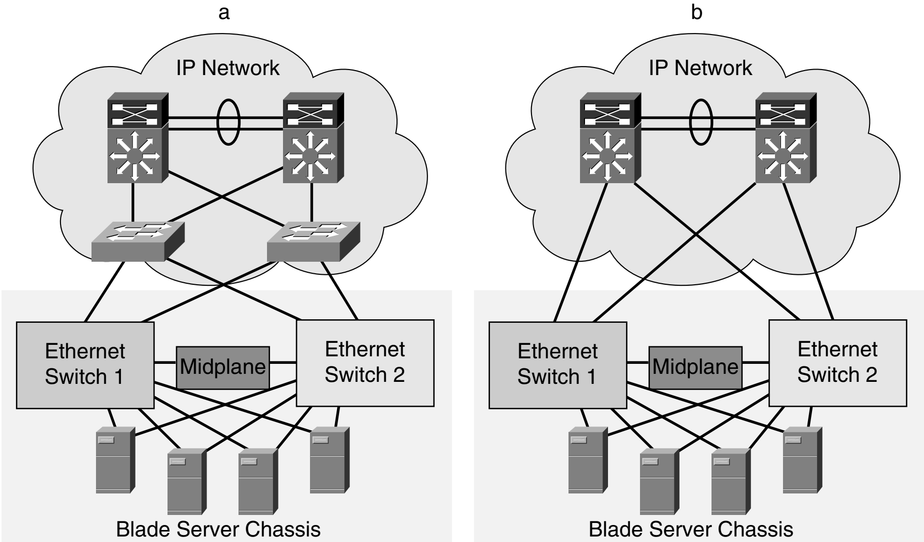
Figure 4-21 *Blade Chassis Uplink Connectivity*



Option **a** in Figure 4-21 shows all the uplinks from both blade chassis' Ethernet switches connected to a single switch in the IP network. This allows the uplinks to be channeled. In contrast, option **b** in Figure 4-21 shows each blade chassis Ethernet switch connected to each IP network switch, also avoiding a single point of failure. This presents the advantage of having a direct link to either switch A or switch B, thus avoiding unnecessary hops. Additionally, if each blade chassis Ethernet switch supports more than two uplinks, they can also be channeled to switches A and B for greater redundancy and higher bandwidth.

The next step is to determine whether to connect the blade chassis to the access-layer switches, as is traditionally done with servers, or to the aggregation layer switches. Figure 4-22 displays the connectivity options for the next-hop switches from the blade chassis.

Figure 4-22 *Blade Chassis Next-Hop Switch*

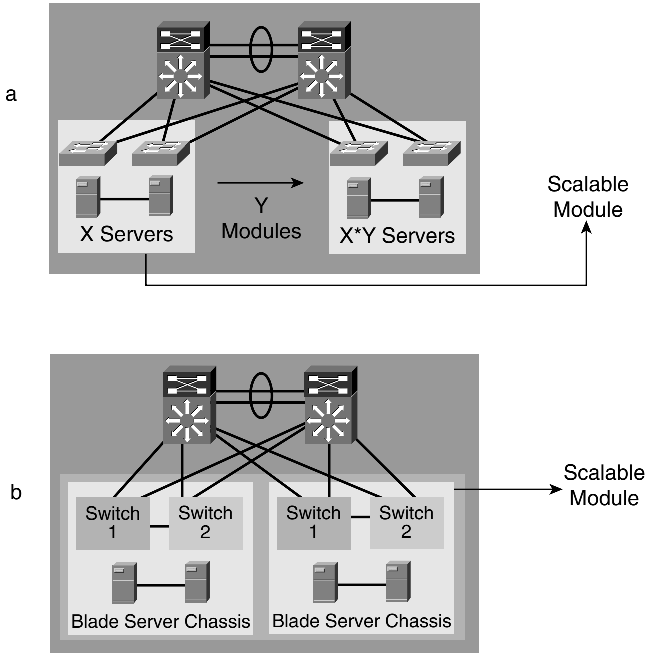


Option **a** in Figure 4-22 shows the blade chassis connected to the access-layer switches. This particular design choice is equivalent to connecting Layer 2 access switches to Layer 2 access switches. The design must take into account spanning tree recommendations, which, based on the topology of option **a** in Figure 4-22, are aimed at determining a loop-free topology given the number of Layer 2 switches and the amount of available paths to the STP root and the secondary root from each leaf node. If the blade chassis Ethernet switches support 802.1w, the convergence time stays within two to three seconds; however, if the support is strictly 802.1d, the convergence time goes back to the typical range of 30 to 50 seconds. Other design considerations have to do with whether the midplane is used for more than management and switch-to-switch control traffic communication functions. If for some reason the midplane also is used to bridge VLANs (forward Bridge Protocol Data Units, or BPDUs) the STP topology needs to be considered carefully. The design goals remain making the topology predictable and deterministic. This implies that you need to explicitly set up root and bridge priorities and analyze the possible failure scenarios to make sure they support the requirements of the applications.

Option **b** in Figure 4-22 shows the blade chassis Ethernet switches directly connected to the aggregation switches. This is the preferred alternative because it lends itself to being more deterministic and supporting lower convergence times. Much like in the previous option, if the blade chassis Ethernet switches do not support 802.1w or some of the STP enhancements such as Uplinkfast and Loopguard, the convergence time would be in the range of 30 to 50 seconds. The topology still needs to be made deterministic and predictable by explicitly setting up root and bridge priorities and testing the failures scenarios.

How to scale the blade server farm is another consideration. Scalability on server environments is done simply by adding pairs of access switches for redundancy and connecting them to the aggregation switches, as shown in option **a** in Figure 4-23.

Figure 4-23 *Server Farm Scalability*



If a single scalable server module supports X servers (limited by port density), higher scalability is achieved by replicating the scalable module Y times (limited by slot density in the aggregation switch). The total number of servers could be X * Y. Depending on the access switch port density and the aggregation switch slot density, this could grow to thousands of servers. Scaling the number of blade servers might require a slightly different strategy. Because blade chassis with Ethernet switches are the access layer, the amount of blade server is limited to the number of slots and ports per slot at the aggregation switches. Option **b** in Figure 4-23 shows this alternative.

Notice that the scalable module is now the aggregation switch along with a set number of blade chassis. This is because the aggregation switch has a limit to the number of slots that can be used for blade chassis. In addition, line cards used to support blade server uplinks now receive aggregate server traffic, thus requiring less oversubscription. This leads to fewer ports used per line card. So, the total number of blade servers is limited somewhat by the slot and port density. Even though this design alternative is likely to support hundreds of blade servers and satisfy the requirements for a fast-growing server farm environment, you must have a plan for what to do if you need to increase your server farm beyond what the current design supports. Figure 4-24 shows this alternative.

Figure 4-24 *Core Layer Within the Data Center*

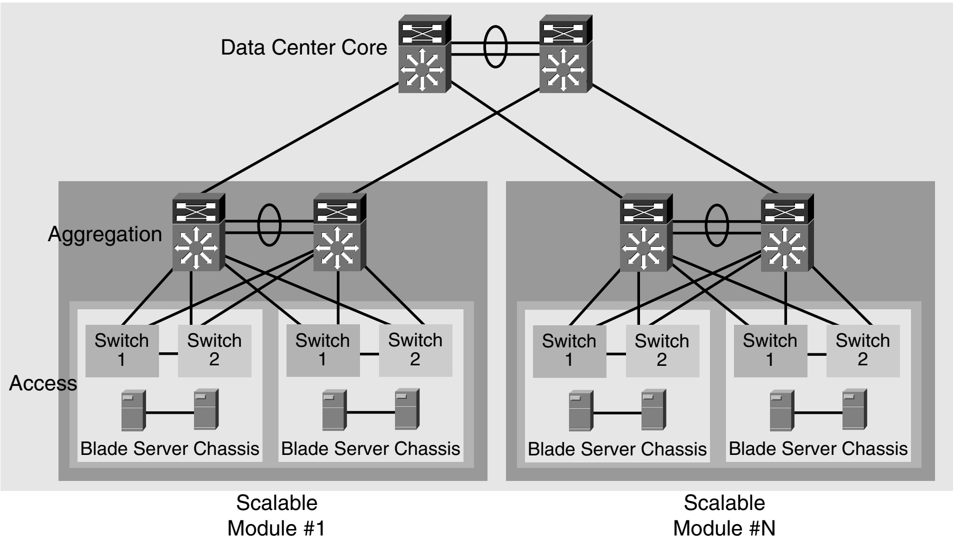


Figure 4-24 introduces a new layer in the Data Center: the core layer. The core layer is used to aggregate as many server blade modules as needed, but the number is limited to the port and slot capacity to the aggregation switches. The pass-through option might not require as much planning because the blade chassis do not have redundant Ethernet switches. The uplinks are connected to the access layer, which is equivalent to current designs in which servers are dual-homed to a redundant set of access switches.

Setting aside the connectivity, port density, slot density, and scalability considerations, other areas, such as oversubscription, uplink capacity, and service deployment options, might require design and testing before the Data Center architecture is established.

Additional trends include the dual-homing of servers, the migration from Fast Ethernet to Gigabit Ethernet, application firewalls, and the use of transparent network service devices. Application firewalls are firewalls that are more in tune with application behavior than ordinary firewalls, thus making the firewalling process more granular to application information in addition to just network or transport layer information. For instance, an application firewall might be capable of identifying not only that a packet is TCP and that the information in the TCP payload is HTTP, but also that the request comes from a specific high-priority user and is a SQL request for sensitive payroll information, which requires a higher security service level.

Transparent network services include firewalling, load balancing, SSL offloading, and so on. These services are provided by network devices with minimal interoperability issues that leave the existing designs unchanged. These transparent services could apply to traditional network services such as load balancing and firewalling, yet they are implemented to minimize disruption and changes in the application environment. This approach might include using physical devices as if they were distinct logical entities providing services to different server farms concurrently. This implies that the administration of those services, such as configuration changes or troubleshooting efforts, is isolated to the specific logical service. Think of it as a single physical firewall that is deployed to support many server farms concurrently where access to the CLI and configuration commands is available only to users who have been granted access to the specific server farm firewall service. This would appear to the user as a completely separate firewall.

Some of these trends are ongoing, and some are barely starting. Some will require special design and architectural considerations, and some will be adopted seamlessly. Others will not exist long enough for concern.

Summary

Data Centers are very dynamic environments hosting multiple types of server farms that all support key business applications. The design of the Data Center involves a variety of aspects related to how applications are architected, how they are deployed, and their network infrastructure.

A sound approach to design involves using a combination of architectural principles, such as scalability, flexibility, and high availability, as well as applying those principles to the requirements of the application environment. The result should be an architecture that meets the current needs but that is flexible enough to evolve to meet the needs of short- and long-term trends.

A solid foundation for Data Center design is based on a redundant, scalable, and flexible Layer 2 and Layer 3 infrastructure in which the behavior is both predictable and deterministic. The infrastructure also should accommodate service devices that perform key functions aimed at scaling or securing application environments. The deployment of service devices such as firewalls, load balancers, SSL offloaders, and caches requires careful planning.

The planning efforts must ensure that the desired behavior is achieved in the following areas: redundancy protocols between service devices, the exchange of connection and session information between stateful devices, the location of default gateway services, and the traffic path through the Data Center infrastructure from device to device.

Additional considerations require an architectural approach to deal with the application environment trends and the requirements that are imposed on the network infrastructure. Subsequent chapters in the book dig deeper into the specifics of Data Center and server farm designs.

INDEX

Numerics

A

B

C

D

E

F

G

H

I

J

K–L

M

N

O

P

Q

R

S

T

U

V

W

X–Z