

Enhancing Availability, Performance, and Security for BEA[®] WebLogic Clusters Using Cisco CSS 11000 Series Content Services Switches¹

Introduction

As enterprises move applications to the Web, they are taking advantage of leading-edge Web application servers like BEA WebLogic Server[™] to shorten development cycles, ease deployment, and enhance application availability. Through advanced clustering capabilities, WebLogic-based e-business applications can be used across multiple servers.² Availability is enhanced by replicating both application components and client session state information. Should a server fail mid-session, the client's session can be restored on another server without losing session data. The WebLogic cluster makes this function transparent to clients that view the cluster as a single "virtual server."

Although software infrastructure has evolved to include specialized applications servers optimized for the delivery of Web content, the network infrastructure must evolve to become "content aware" as well. Cisco Systems is an early pioneer in the development of content switching, a new category of network devices, optimized for the delivery of Web content. These high-performance devices recognize the content being requested, where it is located, and how to route individual content requests to the server best able to manage them. They are used to increase the scalability, performance, availability, and security of high-end Web sites and Web-based applications. This paper describes how the Cisco CSS 11000 Series content services switches can be used to enhance the capabilities of BEA WebLogic clusters.

1. Disclaimer: WebLogic and all its associated capabilities are the property of BEA Systems, Inc. Although we believe the following information to be accurate at the time of writing, Cisco Systems, Inc. makes no warranty as to the accuracy of these statements with respect to WebLogic operation. The reader is referred to BEA Systems for up to date information on WebLogic capabilities.

2. WebLogic supports multiple types of clustering only one of which is relevant here—what is referred to as "Web Clustering". In Web Clustering we are dealing with the clustering of the HTTP or presentation layer of the Web application. It should be assumed that this is what is referred to by the terms "cluster" or "clustering" in this paper. BEA Web Logic Server (WLS) also supports other types of clustering (such as EJB or JMS clustering) that deal with distributing RMI request traffic. These are handled by WLS internally.



WebLogic Clustering in Action

A complete discussion of WebLogic's clustering capabilities is beyond the scope of this paper, however, certain aspects of cluster operation must be understood before proceeding. Particularly germane to our discussion are the mechanisms available for replicating client-session state and for recovering this session state in the event of server failure.

Note: For a more complete discussion of clustering from an applications perspective, see BEA's white paper at: http://www.bea.com/products/weblogic/server/paper_wls_clustering.pdf.

A single client transaction may require multiple HTTP operations to complete. An example of this is the presentation interface for a Web-based stock trading portal. A client might go through several steps in executing a trade such as getting quotes, doing research, placing the order, receiving a confirmation, etc. The application keeps track of the state of this transaction through each of its stages. To provide protection from server failure, the session state is replicated to another server. Should the primary server fail, the session can be reinitiated to a new server (either the server that hosted the state replica or a third server that recovers the session state from it.) WebLogic supports three different methods for replicating and recovering a session state: database replication via Java Database Connectivity (JDBC), file-based replication, and in-memory replication.

File and database replication are similar in operation. Each server in the cluster maintains connections to a shared file server or database server. State information is written to a file or to a database record as it is created or changed. When a server in the cluster fails, subsequent client requests are routed to another available member of the cluster. The new server reads the session ID (established at the initiation of the session and stored in a cookie or encoded in the URL of the request) and fetches the associated state from the database or file system. The new server can now continue processing the client's transaction.

In-memory replication sends state information from a primary server to a designated backup where it is stored in memory. If the primary server fails, the client is routed to a new server. This new server will (1) have the state replica, in which case it designates a new server as the backup, creates a backup copy of the session state, and assumes the primary role, or (2) it will fetch the state from the backup, in which case it assumes the primary role, removes the session state from the backup, elects a new backup server, and creates a new state replica on that new backup. Like the session ID, the backup server ID is encoded in the cookie or URL, allowing the new server to recover the session state from the backup. The new server can now continue processing the client's transaction.



Request Routing

Because multiple servers in a cluster are capable of servicing a particular set of client requests, some mechanism must be used to route client requests made to the virtual server to one of the many servers in the cluster. The first and simplest goal of a request-routing mechanism is to:

1. Balance load across the available servers in the cluster.

However, for transactions that span multiple HTTP operations (and possibly multiple TCP connections), after a client session has been established with a particular cluster member, subsequent operations should be directed to the same member until the session completes. This will reduce the overhead and latency associated with fetching the session state from another server for each successive operation. This improves user response time and overall utilization of the cluster. Therefore, a second goal of request routing for WebLogic clusters is to:

2. Maintain “persistent sessions” between clients and cluster members that are cooperating to perform complex transaction logic.

DNS-based load balancing available in public domain DNS servers such as BIND³ allows the operator to configure several host addresses in association with a particular domain name (for example, www.mystore.com). Clients attempting to access this site will be resolved to each of the configured addresses in round robin fashion. Clients will cache the returned name-to-address mapping for the time-to-live (TTL) configured in the DNS server by the domain name owner.⁴ Subsequent requests to the same host name will be directed to the same server using the cached mapping, thereby creating a crude type of session persistence. A major downfall of DNS-based schemes, however, is that they are incapable of routing around a server failure. When a client is “bound” to a server, that binding remains in place until the TTL for the cached name-to-address mapping expires. Therefore, clients can continue to attempt to contact a host even though that server has failed or been taken out of service. Something more than simple round robin DNS is needed for a robust local request-routing mechanism,⁵ so it is appropriate to expand the list of request-routing goals to include:

3. Rapidly detect and route around server and process failures.

Proxy Web-server-based request routing places an additional server at the front end of the cluster that accepts incoming client requests and directs them to the appropriate server. BEA’s WebLogic Server acting as a proxy or BEA’s plug-in application for third-party Web servers are examples of this approach. By participating in the clustering protocol, the proxy is able to detect failure of servers quickly and route around them. Session persistence is provided via examination of the session cookie. Because it is a full member of the cluster, the proxy understands the session cookie format and the associations between sessions and servers. Because they are general purpose servers, however, these proxies are not optimized for request routing, and their performance is lower than specialized load- balancing appliances.

3. For more information on BIND see <http://www.isc.org/products/BIND/>

4. It should be noted that, although this is the specified behavior for DNS clients, actual behavior varies considerably by operating system and by browser.

5. DNS based techniques, when deployed in conjunction with more robust local request routing schemes are in fact very useful for disaster recovery and for balancing load between multiple geographically separate locations.



Load-balancing appliances are general purpose PCs with software tuned for load balancing applications. Similar to the first generation of routers, these general purpose architectures use a single, centralized CPU for all load-balancing decisions and for packet forwarding. Additional functions are added by taking off-the-shelf PC boards such as Secure Sockets Layer (SSL) acceleration cards and writing drivers to integrate them with the rest of the system. By focusing on a specific task, these appliances provide better performance, redundancy, and features than their proxy-server-based counterparts. Ultimately, however, the hardware architecture of appliances becomes the limiting factor in the performance and scalability of the cluster. As one of the goals for request routing for enterprise class applications, we must include performance to:

4. Scale in accordance with the anticipated client request volume and data transfer rates.

Similar to the way that general purpose computing architectures became a limiting factor to scaling for the first generation of routers, general purpose PC architectures have become a limiting factor for request routing. Recognizing this, Cisco pioneered the development of technology known as content switching.

The core intellectual property associated with content switches is a switching architecture that separates the processing of the control functions associated with request routing (server selection, session establishment, and health checking, etc.) from the processing associated with the packet forwarding functions of request routing (network address translation (NAT), TTL decrementing, Media Access Control (MAC) address replacement, etc.). This fundamental separation of functions allows Cisco content switches to provide superior features, functions, and performance to their PC-based load-balancing counterparts.

Cisco is now providing its third generation of content switches and has invested significantly in a level of integration between the network and cluster that is a leader in the industry. In the balance of this paper we will discuss how to design an infrastructure using Cisco CSS 11000 Series content switches that delivers reliability, availability, scalability, performance, and security for BEA WebLogic clusters.

Content Switching Deployment

The Cisco CSS 11000 Series is a scalable set of content switches that range from low-end, fixed configurations to two dedicated mid-range chassis configurations with integrated SSL termination. Specifically, the product line includes the Cisco CSS 11000 Content Services Switch in both fixed (Cisco CSS 11150 and 11050) and chassis (Cisco CSS 11503 and Cisco CSS 11506).

Content switches perform Layer 4 to Layer 7 request routing and load balancing within a data center to provide the best possible end user experience, make best use of available resources such as servers, firewalls, caching systems, and staff, and provide solid security of a customer's Web infrastructure. With patented content switching technology, the Cisco CSS 11000 Series provides several advantages to the enterprise seeking integrity and consistent performance of their e-business infrastructure. In many customer sites, a Cisco content switch is combined with Cisco Catalyst® LAN switches, Cisco PIX® firewalls, Cisco routers, and other Cisco products to provide a complete data center networking solution offering high capacity, security, availability, and network intelligence.



Cisco CSS 11000 Series (left) and Cisco CSS 11500 Series (right) Content Services Switches



Logically, content switches are deployed in front of the WebLogic cluster. Requests to the cluster are directed to a virtual IP address (VIP) that represents the cluster to the public. The content switch receives connections and HTTP requests from outside the infrastructure and routes them to the appropriate member of the cluster based on configured policies. In the case of BEA WebLogic clusters, these policies include the failover and persistence mechanisms discussed previously.

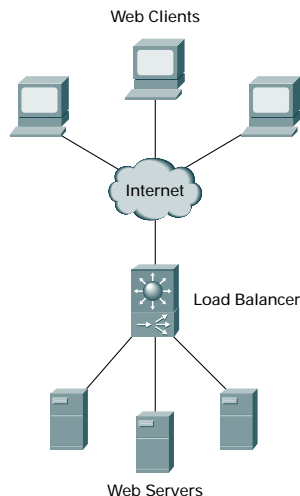
Physically, the network topology can take many forms. It is important to recognize that there are several factors influencing data center network design. Examples include:

- Network location—whether it is on the customer premise or co-located at a service provider
- Firewall deployment—whether there is one layer or many layers of firewall protection
- Intrusion detection deployment
- Topology of the management infrastructure
- Other devices that must be connected in the same facility

The importance of these factors in designing the overall network and in selecting individual components of the design cannot be overstated.

Although a complete discussion of these factors is beyond the scope of this paper, the following point needs to be made. When discussing the merits of various load-balancing and request-routing products, marketing presentations focus on simple topologies such as the one depicted in Figure 1.

Figure 1. Simple Load-Balancing Scenario





Although a simple presentation makes it easy to focus on specific load balancing features, it is flawed because networks are seldom built this way. Networks constructed to support Web applications as indicated above are much more complex. More seemingly mundane issues such as firewall placement, virtual LAN (VLAN) layout, routing design, spanning tree design, addressing, and redundancy are significant components of robust system design.

Specific features designed to enable or simplify the deployment of complex infrastructure are as important to consider because they are typically associated with load balancing. A useful discussion of request routing, therefore, needs to take place within the context of a typical set of requirements for Web application deployment. A typical set of requirements might include the following:

- The application requires no single point of failure—including redundant connectivity to the Internet
- A Layer 2 to Layer 3 infrastructure is required to provide connectivity for the data center
- The WebLogic cluster uses in-memory replication for application-state maintenance and failover, and has connections to a back-end database for application support. Firewalls are required between the Internet and the cluster and between the cluster and the backend database
- Physical isolation is required between the backend database(s) and the rest of the network for security reasons (i.e. So that neither accidental nor deliberate reconfiguration of the network can result in traffic bypassing this “interior” firewall layer)
- The network must provide graceful failover and reroute quickly around failed network or server components

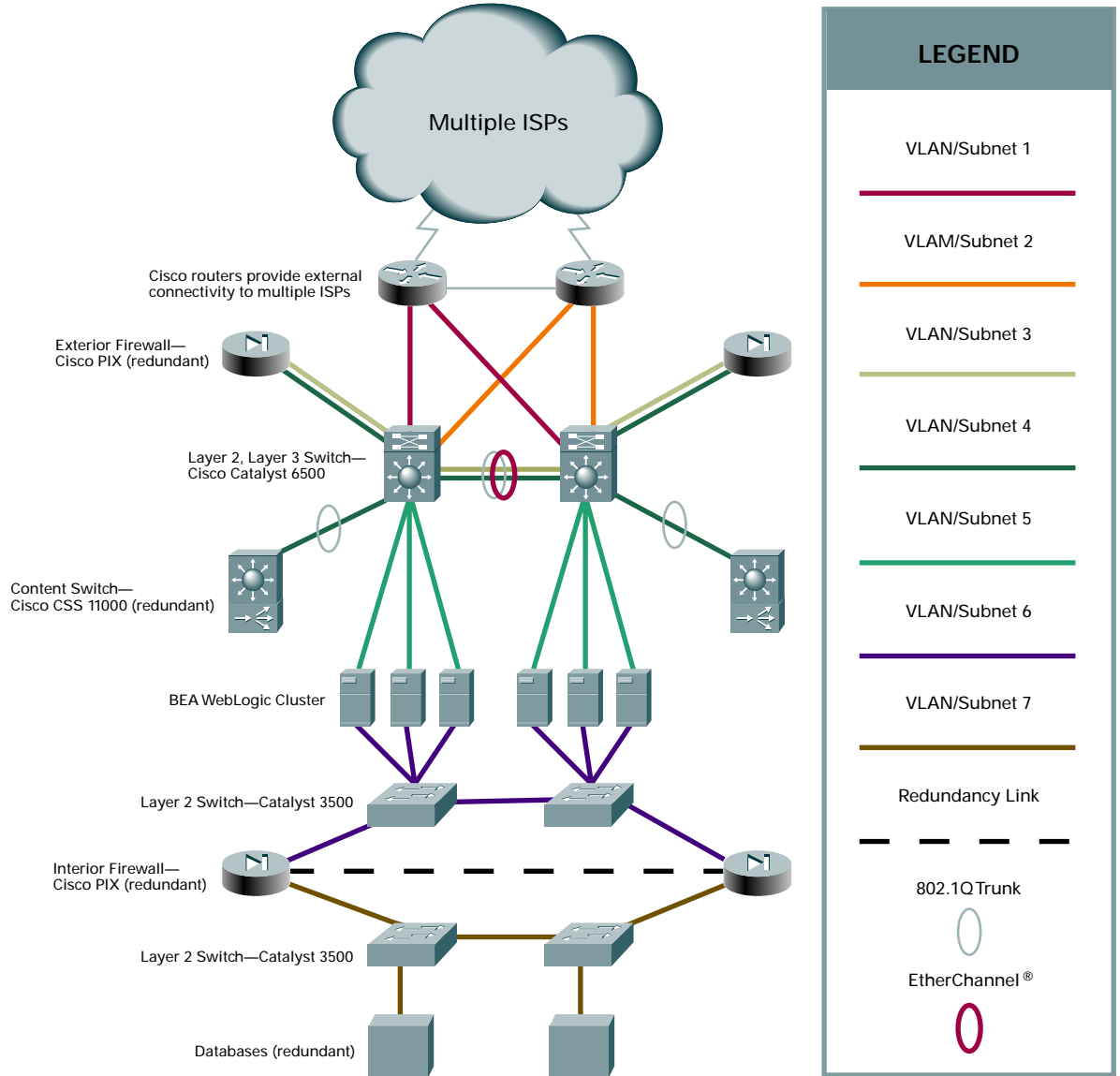
Network designs that meet these requirements are far more complex than the simple topology shown in Figure 1. Figure 2 shows a network design that meets these requirements using a combination of Cisco Catalyst switches operating at Layer 2 and Layer 3, Cisco PIX firewalls for security, and standalone Cisco CSS 11000 Series content switches to provide request routing and failover for the WebLogic cluster.⁶

We now turn to the aspects of these designs that enhance the availability, performance, and security of the Web application.

6. It should be noted that there are many permutations of this network design, which could be discussed in order to meet these and other requirements. In the interest of brevity they are not covered in this paper whose primary purposes is to discuss the features of Cisco Content Switching which enhance BEA WebLogic clustering.



Figure 2. Network Design Using Standalone Cisco CSS 11000 Series Switches





Robust Layer 2 and Layer 3 Features Provide High Network Availability

Server failure is only one of many events that can affect overall system availability. In fact, failure of any single network component, firewall, or server is a potential trouble spot. For this reason, networks are typically deployed with complete redundancy in all aspects—from network connectivity to firewalls, servers, content switches, and to ISPs.

Features such as port fast, root fast, uplink fast, and root guard provide spanning tree robustness and rapid spanning tree convergence around failures. Hot Standby Router Protocol (HSRP) and Virtual Router Redundancy Protocol (VRRP) provide next hop redundancy for servers. The industry's most heavily used IP routing stack provides unmatched support for both interior and exterior routing protocols such as Open Shortest Path First (OSPF) Protocol and Border Gateway Protocol (BGP). EtherChannel technology provides simplicity for scalable network bandwidth and 802.1Q provides simplicity for VLAN deployment. Operating together in complex deployments such as those depicted, these protocols and capabilities combine to provide a high level of network availability.

Rich Content Switching Capabilities Provide High Availability of Server Farms

Cisco content switches provide the most complete set of features for maintaining a high-availability WebLogic cluster. These features include switch redundancy, session state failover, and advanced health checking. Each of these is described briefly as follows.

Switch Redundancy

Cisco CSS 11000 Series content switches may be deployed in redundant pairs. If the primary switch fails, the secondary switch takes control. Depending on the configuration of session state redundancy, this failover may take place without disrupting the client-to-server connection. Certain configurations may use both switches as simultaneously active and backup for certain groups of servers or content types. This configuration is known as “active-active” redundancy. It is critical to note that convergence needs to occur at the network as well as at the content switch level, making content switch features such as VRRP or HSRP on both uplinks and server links required.

Session State Redundancy

For some applications it is critical for content switch failover to occur without disrupting existing client-to-server connections. For these applications, the content switches may be configured to maintain session state information on both primary and secondary switches. In the event of a failure, the redundant switch will forward packets associated with the existing connection between client and server as soon as the underlying network reconverges.

Advanced Health Checking

Cisco content switches use both active and passive techniques to monitor server health. By periodically probing servers, the content switch will rapidly detect server failures and quickly re-route connections to available servers. A variety of health checking features are supported, including the ability to verify Web servers, SSL servers, application servers, databases, File Transfer Protocol (FTP) servers, streaming media servers, and a host of others.



Cisco Cookie Switching Provides Most Flexible Set of Session Persistence Options

As discussed previously, after the initial selection of a server within the cluster is made, for performance reasons it is important to keep connections from the same client routed to the same server until the transaction is completed. This is called “session persistence.” It is critical that the content switch leave the session cookies sent from server to client and from client to server intact. Cisco CSS 11000 Series content switches support two different methods for achieving session state persistence for WebLogic clusters. Which one to use will be the choice of the application and network designers.

Active Cookie Insert

When using this method the content switch is configured to insert a unique cookie for each server when a new session is detected. This cookie is then sent back by the client to the server and used by the content switch to keep the session routed to the same server.

Cookie Matching

In this mode of operation, the content switch is configured to look for a specific string in the cookie that indicates the server that initially set it. This may either be a string that is uniquely identified in the session cookie for this purpose or a different cookie set by the WebLogic programmer. Although this method is potentially more complex from a programming perspective, it gives the Web application designer the most flexibility in controlling application behavior.⁷

Content Switching Technology Enhances Site Security

Cisco content switches help to protect the WebLogic cluster in four different ways:

Access Control Lists

By constructing access control lists on content switches (or on the Catalyst switch in the case of the content switching module), the operator can control who has access to the real IP addresses of the cluster members and who has access to the switches.

Network Address Translation

Cisco content switches perform NAT from the VIP (which represents the cluster to those outside the cluster) to the real IP addresses of the cluster members. This allows the cluster to be numbered using private IP addresses. More importantly, it hides the details of the cluster configuration.

Denial of Service Protection

Because content switches participate in both TCP and HTTP, they are in an ideal position to detect and stop TCP—and HTTP-based denial of service attacks before they affect a server.

⁷ Cookie matching can be used for numerous purposes including dynamic load shedding and third party routing. A discussion of these techniques is beyond the scope of this paper.



Firewall Load Balancing

If the application generates enough traffic to warrant additional firewalls, Cisco content switches can be used to load balance several firewalls. This is rarely needed, though, because Cisco PIX firewalls can support traffic at 1 GB per second.

SSL Termination Improves Performance and Enables Persistence for Secure Sites

Running SSL on the WebLogic servers is a tremendous drain on server resources. By offloading SSL processing, those resources can be applied to traditional WebLogic functions. In addition, because persistence information used by the content switches is inside the HTTP header, this information is no longer visible when carried inside SSL sessions. By terminating these sessions before applying content switching decisions, all the persistence options previously discussed become available for secure sites.

The Cisco CSS 11503 and Cisco CSS 11503 chassis can support optional integrated SSL termination modules. The Cisco 11150 and 11150 standalone platforms can be deployed with Cisco standalone Cisco SCA 11000 Series secure content accelerators to achieve the same level of functionality.

Cisco Content Switches Provide Extensive Scalability and Performance Options

With the industry's broadest content switching portfolio, Cisco offers customers the maximum flexibility when designing their sites. Products range from the entry level Cisco CSS 11050, which supports 1 GB per second of uplink bandwidth, to the Cisco Catalyst switch with content switching modules that can scale to 32 GBs per second and 1,000,000 HTTP requests per second.⁸ Note that a single Cisco Catalyst 6500 Series switch can support up to 10 content switching modules.

In addition to the switching performance of the platforms themselves, Cisco content switches provide the following capabilities, which can be used to scale the performance of the servers as well as the performance of the client-to-server connection.

Server Farm Partitioning

Content switches can be used to partition components of a single Web application across several cluster members. For example the two URLs www.mycompany.com/quotes/getquote.jsp and www.mycompany.com/trades/order.jsp could be located on two different servers even though the domain name is the same. This allows the application developer to easily scale the application to several servers without numerous code modifications. Furthermore, it maximizes the cache coherency of the servers by keeping requests for the same pages on the same servers.

Additionally, content switches may be used to push requests for cacheable content such as image files to a set of caches that can serve them more cost effectively than the application servers.

HTTP 1.1 Connection Re-Mapping

When using HTTP 1.1, clients may request several URLs over the same TCP connection. This allows the client and server to run more efficiently by reducing the overhead of TCP connection maintenance. When content is partitioned across multiple servers or caches as described in the preceding section, it is important to be able to send multiple

8. Up to 10 CSM's can be supported in a single Catalyst 6500

HTTP “gets” for different pieces of content that might arrive over the same HTTP 1.1 connection to different servers. Effectively, the client side HTTP 1.1 connection must be “re-mapped” from one server to another. Cisco content switches perform this function to maximize connection efficiency.

Summary and Conclusion

In this paper we have briefly discussed how Cisco CSS 11000 Series content switches can be used to enhance the availability, security, and performance of BEA WebLogic clusters. With the broadest array of platforms and a superior set of features for application server clustering, Cisco is ideally positioned to help customers design, implement, and scale even the most demanding Web-based applications. Taking advantage of pioneering technology, Cisco content switches deliver higher performance for these applications than any other products in the marketplace.

This superior request-routing performance notwithstanding, it is important to recognize that server and process failures are not the only events that can affect cluster performance and availability. The system cannot be more reliable than its weakest link. If the network infrastructure is not reliable and available, sophisticated server load-balancing algorithms are of little value. If the request-routing function does not incorporate information about availability and performance of the network, extraordinary measures taken to ensure network redundancy and availability are also of little value. Cisco developed content switching for easy integration into the rest of the network, greatly enhancing overall system performance. Finally, Cisco has developed a set of features that allow the site designer to improve the performance of servers in the cluster and the performance of clients to optimize the system for a superior user experience.

About BEA

BEA Systems, Inc. (Nasdaq: BEAS) is the world’s leading application infrastructure software company, providing the enterprise software foundation for 13,000 customers around the world, including the majority of the Fortune Global 500. BEA and its WebLogic® brand are among the most trusted names in business.

Headquartered in San Jose, Calif., BEA has 91 offices in 33 countries and is on the Web at <http://www.bea.com>.



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 Europe
11 Rue Camille Desmoulins
92782 Issy-les-Moulineaux
Cedex 9
France
www-europe.cisco.com
Tel: 33 1 58 04 60 00
Fax: 33 1 58 04 61 00

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: +65 317 7777
Fax: +65 317 7799

Cisco Systems has more than 200 offices in the following countries and regions. Addresses, phone numbers, and fax numbers are listed on the
Cisco Web site at www.cisco.com/go/offices

Argentina • Australia • Austria • Belgium • Brazil • Bulgaria • Canada • Chile • China PRC • Colombia • Costa Rica • Croatia
Czech Republic • Denmark • Dubai, UAE • Finland • France • Germany • Greece • Hong Kong SAR • Hungary • India • Indonesia • Ireland
Israel • Italy • Japan • Korea • Luxembourg • Malaysia • Mexico • The Netherlands • New Zealand • Norway • Peru • Philippines • Poland
Portugal • Puerto Rico • Romania • Russia • Saudi Arabia • Scotland • Singapore • Slovakia • Slovenia • South Africa • Spain • Sweden
Switzerland • Taiwan • Thailand • Turkey • Ukraine • United Kingdom • United States • Venezuela • Vietnam • Zimbabwe