CHAPTER 1

Overview

The Cisco 4700 Series Application Control Engine (ACE) appliance performs server load balancing, network traffic control, service redundancy, resource management, encryption and security, and application acceleration and optimization, all in a single network appliance.

This chapter contains a high-level introduction to the following topics:

- ACE Technologies
- Setting Up an ACE Appliance
- Creating Virtual Contexts
- Configuring Access Control Lists
- Configuring Role-Based Access Control
- Configuring a Virtual Server
- Configuring a Load-Balancing Predictor
- Configuring Server Persistence Using Stickiness
- Configuring SSL Security
- Configuring Health Monitoring Using Health Probes

ACE Technologies

Server load balancing helps ensure the availability, scalability, and security of applications and services by distributing the work of a single server across multiple servers.

When you configure server load balancing on your ACE appliance, the ACE decides which server should receive a client request such as a web page or a file. The ACE selects a server that can successfully fulfill the client request most effectively, without overloading the selected server or the overall network.

Table 1-1 shows the ACE technologies that provide network availability, scalability, and security at both the device and network services levels.

Table 1-1 ACE Technologies

Level	Availability	Scalability	Security
Device	Device Setup	Virtual Contexts	Access Control Lists
		Role-Based Access Control	
Network Services	Virtual Servers	Load Balancing Predictors	SSL
	Health Probes	Server Persistence Using Stickiness	Access Control Lists
		Role-Based Access Control	

At the device level, the ACE provides high network availability by supporting:

- Device redundancy—The high availability support of the ACE allows you to set up a peer ACE device to the configuration so that if one ACE becomes inoperative, the other ACE can take its place immediately.
- Scalability—Supports virtualization by partitioning one ACE device into independent virtual devices, each with its own resource allocation.
- Security—Supports access control lists which restrict access from certain clients or to certain network resources.

At the network service level, the ACE provides:

- High services availability—Supports high-performance server load balancing, which distributes client requests among physical servers and server farms, and provides health monitoring at the server and server farm levels through implicit and explicit health probes.
- Scalability—Supports virtualization using advanced load-balancing algorithms (predictors) to distribute client requests among the virtual devices configured in the ACE. Each virtual device includes multiple virtual servers. Each server forwards client requests to one of the server farms. Each server farm can contain multiple physical servers.

  Although the ACE can distribute client requests among hundreds or even thousands of physical servers, it can also maintain server persistence. With some e-commerce applications, all client requests within a session are directed to the same physical server so that all the items in one shopping cart are contained on one server.
- Services-level security—Establishes and maintains a Secure Sockets Layer (SSL) session between the ACE and its peer which provides secure data transactions between clients and servers.

Setting Up an ACE Appliance

To set up an ACE appliance, you first establish a connection to the ACE and perform the initial device setup required to prepare the ACE for providing application networking services. For more information, see Chapter 2, "Setting Up an ACE Appliance."

Creating Virtual Contexts

Next, you partition the ACE device into multiple virtual contexts, each with its own resource allocation. For more information, see Chapter 3, "Creating a Virtual Context."

Configuring Access Control Lists

Then, you control access to your network resources to guarantee that only desired traffic passes through, and that the appropriate users can access the network resources they need.

You use Access Control Lists (ACLs) to secure your network by permitting or denying traffic to or from a specific IP address or an entire network.

You must configure an ACL for each interface on which you want to permit connections. Otherwise, the ACE will deny all traffic on that interface. An ACL consists of a series of ACL permit-or-deny entries, with criteria for the source IP address, destination IP address, protocol, port, or protocol-specific parameters. Each entry permits or denies inbound or outbound network traffic to the parts of your network specified in the entry.

This guide provides an example of ACL configuration at the device level (see Chapter 4, "Configuring Access Control Lists"). To learn how to configure ACL at the network services level, or how to configure more granular access control security, see the *Cisco 4700 Series Application Control Engine Appliance Security Configuration Guide*.

Configuring Role-Based Access Control

You can manage the complexity of large-network security administration by defining the commands and resources available to each user through Role-Based Access Control (RBAC). RBAC supports network security at both the device and network services levels by defining physical or virtual resources in a domain that the user can access.

For more information, see Chapter 5, "Configuring Role-Based Access Control."

Configuring a Virtual Server

You can configure a virtual server to intercept web traffic to a website and allow multiple real servers (physical servers) to appear as a single server for load-balancing purposes.

Table 1-2 illustrates how the ACE supports scalability through virtual contexts, virtual servers, server farms, and real servers.

Table 1-2 ACE Scalability

<table>
<tr><td rowspan="20">ACE</td><td rowspan="10">Virtual Context 1</td><td rowspan="5">Virtual Server A</td><td rowspan="3">Server Farm A</td><td>Real Server A1</td></tr>
<tr><td>Real Server A2</td></tr>
<tr><td>……</td></tr>
<tr><td>Real Server An</td></tr>
<tr><td rowspan="3">Backup Server Farm a</td><td>Real Server a1</td></tr>
<tr><td>Real Server a2</td></tr>
<tr><td>……</td></tr>
<tr><td>Real Server an</td></tr>
<tr><td rowspan="5">Virtual Server B</td><td rowspan="5">Server Farm B</td><td>Real Server B1</td></tr>
<tr><td>Real Server B2</td></tr>
<tr><td>……</td></tr>
<tr><td>Real Server Bn</td></tr>
<tr><td rowspan="9">Virtual Context 2</td><td rowspan="5">Virtual Server C</td><td rowspan="5">Server Farm C</td><td>Real Server C1</td></tr>
<tr><td>Real Server C2</td></tr>
<tr><td>……</td></tr>
<tr><td>Real Server Cn</td></tr>
<tr><td rowspan="5">Virtual Server D</td><td rowspan="5">Server Farm D</td><td>Real Server D1</td></tr>
<tr><td>Real Server D2</td></tr>
<tr><td>……</td></tr>
<tr><td>Real Server Dn</td></tr>
<tr><td>……</td><td>……</td><td>……</td><td>……</td></tr>
</table>

You can partition your ACE into multiple virtual contexts, each of which has its own set of policies, interfaces, and resources. A virtual server is bound to physical resources that run on a real server in a server farm.

Real servers relate to the actual, physical servers on your network. They can be configured to provide client services or as backup servers.

Related real servers are grouped into server farms. Servers in the same server farm often contain identical content (referred to as mirrored content) so that if one server becomes inoperative, another server can take over its functions immediately. Mirrored content also allows several servers to share the load during times of increased demand.

For more information, see Chapter 6, "Configuring Server Load Balancing."

Configuring a Load-Balancing Predictor

To distribute incoming client requests among the servers in a server farm, you define load-balancing rules called predictors using IP address and port information.

When there is a client request for an application service, the ACE performs server load balancing by deciding which server can successfully fulfill the client request in the shortest amount of time without overloading the server or server farm. Some sophisticated predictors take into account factors such as a server's load, response time, or availability, allowing you to adjust load balancing to each application's particular past.

For more information, see Chapter 7, "Configuring a Load-Balancing Predictor."

Configuring Server Persistence Using Stickiness

You can configure the ACE to allow the same client to maintain multiple simultaneous or subsequent TCP or IP connections with the same real server for the duration of a session. A session is defined as a series of interactions between a client and a server over some finite period of time (from several minutes to several hours). Cisco calls this server persistence feature stickiness.

Many network applications require that customer-specific information be stored persistently across multiple server requests. A common example is a shopping cart used on an e-commerce site. With server load balancing in use, it could potentially be a problem if a back-end server needs information generated at a different server during a previous request.

Depending on how you have configured server load balancing, the ACE sticks a client to an appropriate server after it has determined which load-balancing method to use. If the ACE determines that a client is already stuck to a particular server, then the ACE sends subsequent client requests to that server, regardless of the load-balancing criteria. If the ACE determines that the client is not stuck to a particular server, it applies the normal load-balancing rules to the request.

The combination of the predictor and stickiness enables the application to have scalability, availability, and performance even with persistence for transaction processing.

For more information, see Chapter 8, "Configuring Server Persistence Using Stickiness."

Configuring SSL Security

Use the SSL security protocol for authentication, encryption, and data integrity in a Public Key Infrastructure (PKI).

SSL configuration in an ACE establishes and maintains an SSL session between the ACE and its peer, enabling the ACE to perform its load-balancing tasks on the SSL traffic. These SSL functions include server authentication, private-key and public-key generation, certificate management, and data packet encryption and decryption.

For more information, see Chapter 9, "Configuring SSL Security."

Configuring Health Monitoring Using Health Probes

Application services require monitoring to ensure availability and performance. You can configure the ACE to track the health and performance of your servers and server farms by creating health probes. Each health probe that you create can be associated with multiple real servers or server farms.

When you enable ACE health monitoring, the appliance periodically sends messages to the server to determine server status. The ACE verifies the server's response to ensure that a client can access that server. The ACE can use the server's response to place the server in or out of service. In addition, the ACE can use the health of servers in a server farm to make reliable load-balancing decisions.

For more information, see Chapter 10, "Configuring Health Monitoring Using Health Probes."