CHAPTER 3

The Generic Provisioning Problem

Chapter 2, "IP Architecture Overview," looked at some of the technologies and standards that are being developed within the Internet community. However, the deployment of these technologies is usually a complex and difficult undertaking. The rollout of the technology within a network requires configuring the different devices within the network that support those technologies. Policy-based techniques can help alleviate the management issues associated with the rollout of these technologies.

The configuration and provisioning of new technologies is complex for a variety of reasons. One of the challenges is that deployment requires configuring a large number of devices in a typical-sized network. Another challenge is that the technologies are often developed without adequate attention to the issues of management and deployment. As a result, there is often a mismatch between the technology specifications and the reason a network operator might want to deploy a given technology within the network.

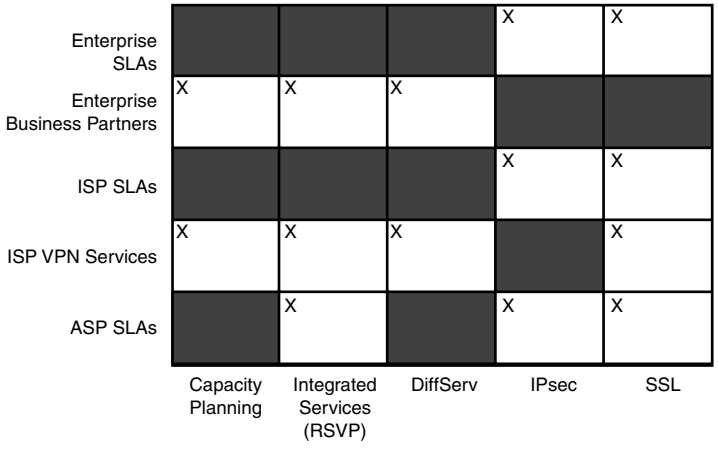
The administrators and operators of most networks do not want to deploy technology for the sake of the technology. Very few enterprise CIOs would choose to deploy Differentiated Services within their network just because it has been standardized or because it is readily available from the different router/server vendors. The reason for deploying any technology is to satisfy a business need. For example, if a CIO determines that it needs to satisfy a network performance SLA put in place for an important customer, and Differentiated Services is the best approach to satisfy that SLA, the CIO might choose to deploy the relevant technology within the enterprise.

As described in Chapter 1, "Policy-Enabled Networking Architecture," there are two types of policies: high-level (operator view) and low-level (device view). The business needs of the deployers drive the high-level policy definitions, and the details of the technology drive the low-level policy definition. A policy management tool bridges the gap between the two views and simplifies the job of technology deployment. This chapter takes a closer look at the high-level and low-level policies that can be used in different types of network environments.

The organization of this chapter can be explained through the policy application matrix shown in Figure 3.1. The vertical axis of the matrix shows the different business needs that can arise in the various types of network environments. The horizontal axis shows the different technologies that can be used to satisfy these business needs. A shaded box indicates that the corresponding technology can be used to satisfy the matching business needs within a specific environment. Not all technologies are appropriate for each business need, and the ones that are not suitably matched have an X. An overview of the technologies shown on the horizontal axis is provided in Chapter 2.

The reader must keep in mind that the entries in Figure 3.1 are shown mainly for the purpose of illustration. In any organization, one may find business needs that are not captured by the items shown on the vertical axis. Similarly, in certain cases, some of the business needs may be better satisfied by mapping to a technology that is different from the one shown in Figure 3.1. As an example, an *Application Service Provider* (ASP) may want to have a business SLA covering privacy and security needs of its customers. In this case, SSL would be useful for satisfying these SLAs, even though it is not shown as an appropriate technology in the figure.

Figure 3.1 Policy Application Matrix



The next section of this chapter takes a closer look at three different types of network environments that are common in current IP-based networks. Then I will describe the two axes of the policy application matrix. Specifically, I will discuss the high-level policies that are most appropriate for the business needs of the different network environments. After that, I will describe the low-level policies that are appropriate for each of the technologies that can be used to satisfy those business needs. Finally, I will describe the structure of a generic policy management tool that can be used to map the policy representation along the vertical axis into the policy representation along the horizontal axis.

This chapter focuses on the generic principles that can be exploited to build a policy management tool. The details of how a technology can be used to satisfy the various business needs are described in Chapter 4, "Technology Support for Business Needs."

Business Environments

The high-level policies within the network reflect the business needs that motivate the deployment of a specific technology. An organization's business needs include items such as the services it provides to its customers, its internal requirements for smooth operation, or compliance with any regulatory or legal statutes that might apply to that organization.

The business needs of each organization depend on the organization's nature and characteristics. In order to study these business needs, we will look at three types of organizations, each with a different set of networking needs: an enterprise, a networking services provider, and an application hosting services provider. The enterprise environment is a corporation with its own network and computational infrastructure. The networking services provider, an *Internet service provider* (ISP), offers network connectivity services to its customers. The application hosting services provider hosts applications, such as Web sites or mail servers for its customers.

This section describes the different business environments in more detail. The following section examines their business needs.

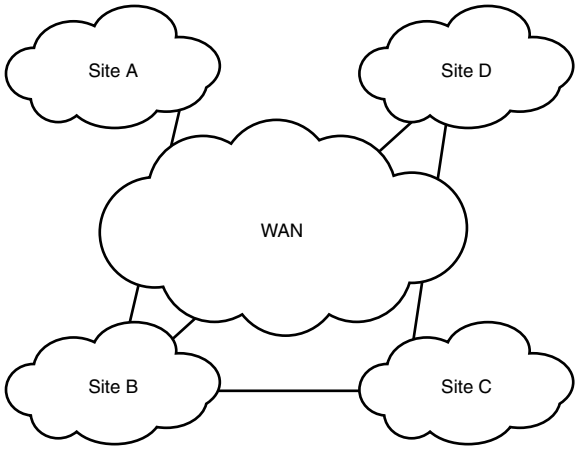
The Enterprise Environment

The enterprise environment represents the typical network of a large corporation. Such a corporation typically has many branches that could be geographically separated by large distances. Depending on the size of the enterprise, the branches might be a few miles away, or they could be distributed across the globe.

Such an enterprise is typically managed by an IT department, which is responsible for the operation of the networking and computing infrastructure. The IT department needs to operate the network at required performance levels and enforce adequate security in networked communication. The management and operation of such a network can benefit tremendously from a policy-based approach.

A typical enterprise environment consists of several local campus networks that are connected by a WAN. Campus networks typically are based on LAN technologies, such as Ethernet, Gigabit Ethernet, FDDI, and token ring. Each LAN connects to the WAN by means of one or more access routers. The WAN typically consists of leased lines obtained by the enterprise from a telecommunications company or an ISP. Such an environment is shown in Figure 3.2.

Figure 3.2 Enterprise Environment



In most cases, the enterprise has adequate bandwidth on the LANs for its purposes. Current LAN technologies are cheap enough to offer gigabits of local bandwidth at relatively low costs. Even at the relatively old technologies of ethernets operating at 10Mbps and token rings operating at 16Mbps, LAN bandwidth is an order of magnitude faster than access links to WAN, which are predominantly T1 (1.5Mbps) links. New technologies of Fast Ethernet and Gigabit Ethernet are significantly faster than T3 access links (45Mbps), which are becoming more widespread.

The reasons that deployed LANs are significantly faster than wide-area access links are most likely business-related rather than technical. There is no technical reason why a wide-area link can't be as fast as or faster than a LAN. Speeds in excess of 1Gbps can be achieved quite readily over optical fiber. However, the cost of operating a LAN is significantly different from the cost of operating a wide-area link.

The installation of a LAN is dominated by a one-time lump-sum cost associated with the purchase of equipment and any building infrastructure updates that might be needed. Although this cost might be a substantial one-time expense, relatively few recurring monthly costs are associated with LAN bandwidth. Amortized over the typical life span of a building LAN (which is usually a few years), the cost per unit of bandwidth per month of LAN is fairly small. However, wide-area access links are available only through the telecommunication companies and are priced on a monthly basis, which easily dwarfs the expenses associated with the one-time equipment installation costs over one year. Unless there are fundamental shifts in the way wide-area links are priced, it is safe to assume that the bottlenecks associated with network communications are more likely to be in the WAN rather than the LAN.

As far as security, different corporations vary widely in the trust model that exists among campuses in the enterprise network. Most companies tend to have an open internal access approach, whereby almost anyone can have network connectivity (such as being able to ping the computer) to any computer on the internal corporation network. This is not to say that access controls at the application layer are not needed. In most companies, files and documents should be given only to those employees who really need them. However, no firewalls or other network-layer security devices are commonly deployed between the different departments.

Enterprise networks do have to contend with a different type of security issue. There is an increasing need for employees to gain remote access to the company's network from external locations. In many cases, the external location is the residence of an employee working from home. However, there might be employees who are accessing the network from their hotel rooms or over the Internet. Their access needs to be supported as well.

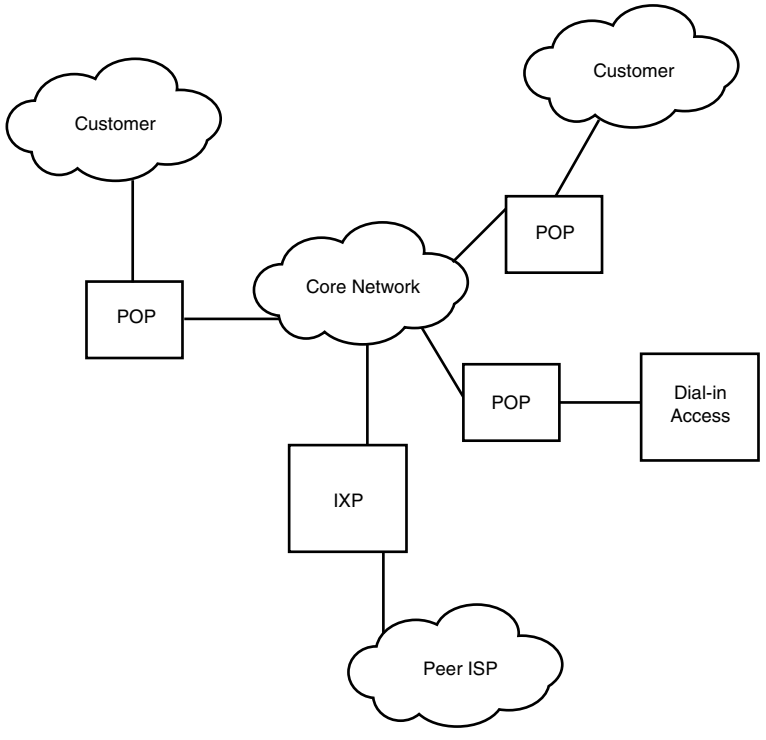
It is in the backdrop of such an enterprise that we will look at the requirement for the high-level and low-level policies necessary to manage the performance and security of an enterprise network. The high-level policies need to be defined in terms of functions that are used in daily operations of the companies, and the low-level policies need to be defined in terms of the technology deployed within the network (such as configuration of access server and routers).

The Network Services Provider (ISP) Environment

The network services provider is an organization that operates a network on behalf of its customers. I refer to such an organization as an ISP because many Internet service providers essentially offer such a service to their customers. Examples of some companies that provide network connectivity are UUNET, Sprint, and AT&T Global Networks. These companies offer WAN connectivity to their customers.

A simplified view of the computer network environment for a typical network provider is shown in Figure 3.3.

Figure 3.3 A Typical ISP Network



The oval shown as the core network is the domain of a single network operator or ISP. An ISP would have multiple *points of presence* (POPs) at various cities. The POPs are sites that could be used to access the ISP network. Customers may connect to POPs using leased lines or dial-up lines. Dial-up access requires modem banks that terminate in a local office of the ISP and are connected to the POP using high-speed links (typically T1). For access to other customers, the ISP may place a router on the customer's premises (CSR: Customer Site Router) and connect it to the POP using a metropolitan-area network or a leased line. In addition to the POPs, the ISP needs to partner with other peer ISPs in order to connect to the Internet. These peering points are known as *Internet Exchange Points* (IXPs). An IXP can connect a regional ISP to a national ISP or act as conduit among several ISP networks. Different ISPs have peering agreements among themselves as to which traffic they will accept from other ISPs at an IXP. Very large service providers also have private peering arrangements with each other.

Note

An exchange point is also known by several names other than IXP. Common equivalent terms include NAP (Network Access Point), MAE (Metropolitan Area Exchange), and FIX (Federal Internet Exchange).

The POPs and IXPs supported by the ISP are interconnected by its core network. The ISP's core network consists of several routers connected by means of high-bandwidth circuits that may be owned by the ISP or leased from other bandwidth vendors.

Note

The public Internet consists of all the ISP networks and the different servers provided by the ISPs or their customers. The IXPs provide the gateways by which a user on an ISP network can access servers provided by a customer of a different ISP.

The Application Hosting Provider Environment

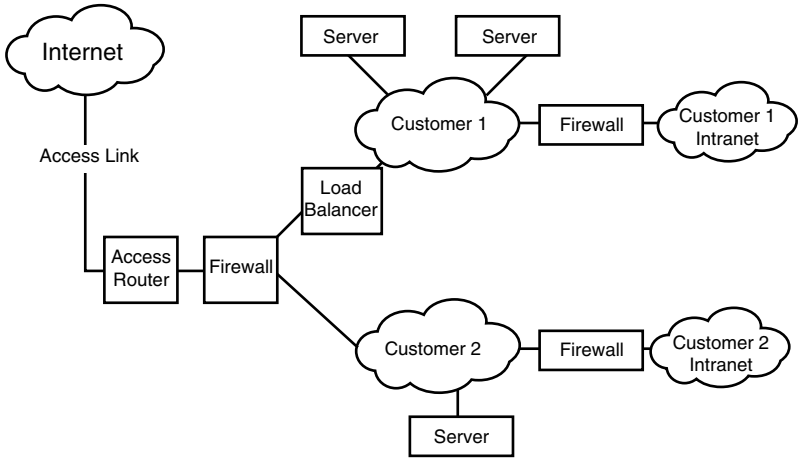
An *application hosting provider* is a company that hosts and supports different types of servers on behalf of their customers. Such companies are commonly referred to as *Application Services Providers* (ASP), which is the acronym I will use in this book. The most common of these providers are Web-hosting companies that provide servers to operate Web sites for individual companies. Examples of such companies are Exodus and IBM Global Services. Exodus primarily provides location services to its customers. For example, it offers space and power supplies to its customers close to one of the major ISP's exchange points. IBM Global Services offers a more comprehensive service, which includes ensuring that the applications and services are up and available. In addition to these major players, many small companies provide services for hosting Web sites as well as Lotus Notes database servers. In 1999, Intel announced a move into a similar business with a plan to open large application-hosting locations.

The most common type of servers outsourced for support and operations are Web servers. However, it is also common to find other types of application servers being hosted. Typical types of application services that are hosted are Web servers, electronic commerce transaction servers, mail servers, groupware servers, and directory servers.

Figure 3.4 shows a simplified configuration of a hosted site. It shows the service provider accessing the Internet through one access router. The access router is typically followed by a firewall that restricts public access from the Internet to only the servers that are intended for this access. The firewall also prevents denial-of-service attacks and otherwise validates access to the server farm. The figure shows two customers, of which Customer 2 has the simpler configuration. Each customer has its own separate LAN on the premises to which

a number of servers are attached. Each customer might also have a back-end connection to this LAN through a firewall to one of the customer's intranet campuses. This connectivity is provided so that the customer can administer or update the applications running on its servers. Customer 1 has a somewhat more complex configuration that involves a load balancer. The load balancer is a device that can spray connection requests across multiple machines running the same applications. Several such types of load balancers are available on the market. This customer also has a back-end connection to its intranet for the sake of easy administration.

Figure 3.4 A Simplified Model of an Application Hosting Environment



In this figure, many variations are possible. Instead of a dedicated connection to the customer's intranet campus, the connectivity might be via a secure VPN over the Internet connection. Similarly, the figure shows a firewall insulating the customers from the Internet as well as from each other. In practice, you might have multiple layers of firewalls. For example, one layer might protect the customers from the Internet, and the other layer protects the customers from each other. Layering firewalls in this manner simplifies the filtering rules to be configured at each layer and helps in checking inadvertent security loopholes. Other variations might include replicating many of the functions, such as access routers and firewalls, in order to ensure some level of tolerance of failures of individual boxes.

High-Level Policies

Within each of the environments described in the previous sections, different types of business needs motivate the deployment of the different technologies. This section looks at some of the business needs that can arise in each of the different environments.

The Enterprise Environment

Within the enterprise environment, the most common needs are that of satisfying the business SLAs that exist between the different parts of the enterprise. For security purposes, an enterprise might need to ensure that access to its network is secure and is accessible only to the employees or other entities that are authorized to have access to specific enterprise resources.

Note

These business needs in this section are intended as illustrative examples for use in subsequent chapters of the book. These examples are not intended as an exhaustive enumeration of all the business needs that can arise in the enterprise environments.

Business SLAs

It is a common practice in many business organizations to have SLAs in place among their different divisions. An SLA is an explicit statement of the expectations and obligations that exist in business relationships between a service provider and its customer [SLAREF]. In the case of an enterprise, the SLA defines the details of the services that an IT department in an enterprise is expected to provide to the other departments that use the computing/networking infrastructure.

Many enterprises choose to outsource the operation and maintenance of their computing infrastructure to another company. This is often the case when the enterprise does not consider running and operating a computer network to be its core skill. The subcontracting company is usually responsible for satisfying an SLA that specifies the performance objectives that are expected from the other company.

An SLA typically contains the following:

- Description of the type and nature of service to be provided
- Expected performance level of the service
- Process for reporting problems with the service
- Time frame within which a response or resolution of a reported problem is expected
- Process for monitoring and reporting the service level
- Credits, charges, or other consequences for the service provider in not meeting its obligation
- If applicable, any escape clauses describing the conditions under which the SLA might not be valid

Although a significant part of the SLA deals with business aspects, such as the process for reporting problems, service-level compliance, the terms of credits, or escape clauses, I will focus more on the technical aspects of SLAs, such as how you meet the performance objectives outlined in the SLA.

Note

As a matter of terminology, some organizations differentiate between an SLA and an SLO (Service-Level Objective). The SLA refers to the overall business agreement and consists of multiple SLOs that need to be satisfied. One of these SLOs is a performance objective that must be satisfied.

A service's expected performance level has two major aspects: reliability and responsiveness. *Reliability* includes availability requirements, such as when the service is available and what bounds on service outages may be expected. *Responsiveness* includes how soon the service would be performed in the normal course of operations. In the context of a computer network, reliability is usually measured in the network's uptime, and responsiveness is measured as bounds on round-trip delays between two customer sites. In order to meet its SLA, the operator of an enterprise network needs to ensure that applications running on the network are available and are performing at the desired level for the target.

The goal for an enterprise IT networking provider is to ensure that the applications supported by to it have a performance that is good enough to meet specific response-time criteria. The performance would be the end-to-end response time of the applications running in the enterprise network.

Chapter 4 describes how business SLAs can be supported effectively by technologies such as capacity planning, Integrated Services, and Differentiated Services.

Extranets

Although the security needs of many enterprises are quite diverse, we will look at a security scenario that often arises in an enterprise environment. This scenario relates to the provisioning and configuring of extranets among different enterprises.

In an era where all enterprises are connected to the Internet, there are many reasons to move many common business-to-business functions to an Internet-based infrastructure. For example, a company might require that its suppliers conduct business with it electronically. Consider a soft drink bottling company that needs to procure various types of containers from its suppliers. Examples of such containers include plastic bottles, metal cans, and glass bottles. A large bottling company typically procures the containers from a variety

of suppliers. Other items supplied by a contractor include labels, bottlecaps, and distilled water. Different bottling plants that are located in different geographic locations might have requirements for different quantities of each type of container or other supplies.

When the bottling company wants to obtain the containers or other supplies, it typically sends a bid out to its list of approved suppliers, who send back quotes and terms for providing the supplies. The bids might be accepted by the company according to a variety of criteria. Although this negotiation is typically done via paper contracts in a traditional fashion, the cost savings that can be realized by migrating the process to an electronic one are significant.

One of the ways in which this system can be realized is by having a bidding server that runs an application that manages the bids received for any specific component, such as containers. The bidding server would be placed so that it is accessible to the suppliers contracted by the company over the Internet. However, for security purposes, the bidding server must be accessible to only a selected subset of suppliers through the Internet.

Such an extranet can be supported and established using the network-level mechanisms of *Internet Key Exchange* (IKE) as well as the transport-level mechanisms of SSL or TLS. The details of how to exploit these technologies to support the notion of extranets are described in Chapter 4.

The Network Connectivity Provider

Within the context of the network connectivity provider, we will look at the business needs of supporting business SLAs and the creation of a VPN service.

Business SLAs

In the networking services provider environment, customers may use the ISP network in one of the three manners:

- To access the Internet
- To interconnect two or more of its sites
- To access proprietary, industry-specific networks

Most customers in the real world probably want to do a combination of these methods. One of the business objectives of any ISP is to support the communication needs of its customers at a reasonable performance level. These target performance levels are often specified as part of an SLA between the customer and the service provider.

When a customer uses the ISP to connect two or more of its sites, SLAs can be defined to ensure some performance level on the network communication between the pair of access routers that connect those two sites. When accessing the Internet, the customer is present at one of the access routers but can communicate with any of the IXPs in the network. The end objective of the customer communication on the Internet is quite likely to be outside the administrative domain of the ISP. Although the ISP cannot honestly offer any assurances about the performance level of the network outside its domain, it can provide some assurances about the performance of the communication within its own domain.

The SLAs provided to the customer are often specified in terms of the delays that can be provided among the different access points within the ISP network. For example, the ISP might provide connectivity among two private campuses of its customer. It might offer an assurance about the maximum latency a packet would experience in the network between the two campuses. An example of such an SLA is the one offered by UUNET to its customers with leased-line access [UUNETSLA].

Another common performance metric used within SLAs relates to the maximum amount of bandwidth than an ISP will accept from the customer for transport across the network. The ISP promises to transfer the specified amount of bandwidth across its network without a significant loss rate.

The terms specified in the SLA might be the same for all the customers, or they might be different for different customers. The former is the more prevalent case in most ISP environments. However, there are many cases where the SLAs would be defined differently for different customers. A customer trying to distribute real-time stock quotes over the network is likely to demand tighter bounds on network latency from its ISP than a customer dealing primarily with storing and forwarding electronic mail.

As in the case of the enterprise environment, the SLAs within the network may be supported by a variety of techniques, including traffic capacity planning, rate control devices, or the deployment of Differentiated Services.

Virtual Private Networks

The most attractive customers for an ISP are enterprises that need to interconnect their campuses. Quite often, these enterprises have their own private network—their intranet. A typical enterprise network (such as the one shown in Figure 3.2) consists of several campuses with their individual LANs and a WAN connecting the campuses. The WAN usually consists of serial links, such as T1 or T3 links, or where bandwidth demands are not that intense—56Kbps dialed or leased lines. Other techniques used for WAN connectivity are frame relay and ATM connectivity services, which gained popularity in the 1990s. You

typically run routers where the IP protocol treats the underlying frame relay or ATM links as a lower-layer protocol. The cost of leasing the links with the right bandwidth is usually the most important factor in the cost of operating the private network.

Because of the growing popularity of the Internet, most enterprise campuses (at least the large ones) are connected to the Internet. The Internet is an open IP network that connects a large part of the world. This results in any pair of enterprise campuses having two paths between themselves—one through the WAN that forms the corporate intranet, and the other one through the Internet. Of course, the intranet path is more secure than the open Internet and is also likely to have much better performance characteristics. As a result, most large corporations maintain their own private corporate networks.

To the ISPs that provide Internet-based connectivity to the enterprise corporations, the emerging scenario provides an attractive option to provide virtual private networks. An ISP can give an enterprise customer the option of eliminating its intranet and replacing it with a virtual network that has comparable performance and security. Such an offering is the *virtual private network* (VPN).

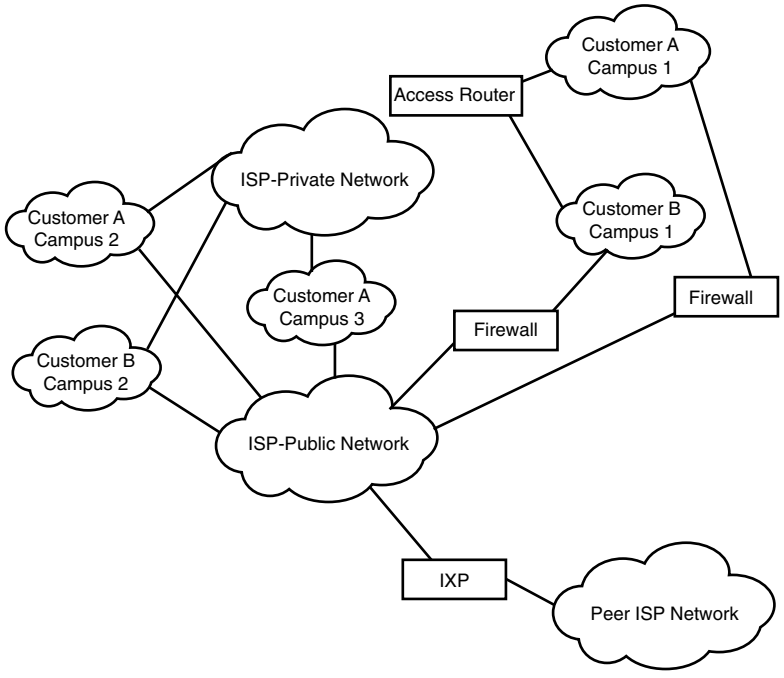
A VPN provides logical connectivity among the different sites of an enterprise while insulating them from each other. Basically, only the sites involved in the VPN should be allowed to communicate with each other. The only exception to this rule is that every campus needs access to the public Internet through some security device, such as a firewall.

The cheapest VPN is obtained if the campus sites simply connect to the Internet and use encrypted tunnels to carry their traffic across the Internet. However, given the wide variety of threats that exist on the open network, as well as the wide performance fluctuations, it is much more likely that the VPN service would be offered on a private network physically separate from the one that the ISP uses for Internet connectivity. A customer network would get access to both the Internet and the ISP-private network. Several customers would be multiplexed to the ISP-private network. This would reduce the overall cost to the ISP of maintaining its network. The ISP would also be able to offer VPN access to the customer at a reduced cost instead of the customer's having an intranet.

Such a scenario is shown in Figure 3.5. An ISP offers VPN services to two customer enterprises on an ISP-private network. The first customer (Customer A) has three sites connected to the network. The second customer (Customer B) has two sites connected to the network. The ISP connects to other peer ISPs at an exchange point, but it typically supports the private connections from the customers within its own network. The customer sites are connected to both the ISP-private network and through firewalls to the ISP-public network. The firewalls connecting the customers to the Internet typically

belong to the customer. At the access routers connecting the customer to the ISP-private network, the ISP has the onus of making sure that the customers are protected from each other. For the sake of brevity, the figure shows access routers and firewalls only at the first campus of both customers. It's implied that a similar structure needs to be in place at each campus.

Figure 3.5 A VPN Deployment Scenario



The high-level policies for deployment—in this case, for the ISP—would be to define the proper set of VPNs and to enable access among its customers in the environment shown in Figure 3.5. Such a solution may be obtained by using IKE and IPsec-based encrypted communication.

Application-Hosting Provider Environment

Within the ASP environment, we will look at the problem of supporting business SLAs and access control as examples of business needs.

Business SLAs

In an application-hosting environment, a service provider can provide its customers with assurances about the availability and reliability of its hosting service. Most service providers promise a 24×7 uptime. In other words, the sites and systems are always operational. From a performance perspective, the application-hosting service provider needs to ensure that the service is running properly and adequately. The service provider also needs to provide SLAs regarding the performance of the applications that will be hosted on the site.

The SLA between the service provider and the hosted customer can be defined in many ways. The simplest and most intuitive way to specify the performance would be to define a response time for a hosted application. The system's response time depends on a variety of factors, including the application design as well as the amount of load in the system. The goal of an application-hosting provider is to ensure that the application operates with specific response targets.

Note that the response target for an application is different from the end-to-end response time that a user of the application service might see. The application-hosting provider has no control over the network that is used to access the application. The network, if congested, can degrade the throughput and end-user response time significantly. However, the application-hosting provider can provide assurance on the part of the server reaction time, which is the time that elapses between the arrival of a request and the beginning of the servicing of the request by the server.

A more typical SLA in the service provider environment might provide the hosted customer with limits on the network bandwidth or server capacity that the customer can use. The limits on network bandwidth can be specified in terms of the absolute bandwidth required, offering a specific customer a slice of bandwidth that connects the application hosting provider site to the Internet. Such an SLA might specify that a customer will receive a performance equivalent to the performance it would have received if it has a dedicated leased line of equivalent bandwidth. Similar rates can also be specified on the connection rates that are supported by the server.

The two most important factors comprising service provider SLAs are the total amount of access bandwidth and the server capacity (the rate of connections that the site can support) allocated to a customer. The fulfillment of business SLAs by an ASP typically involves allocating adequate amounts of these two resources to different customers.

Access Control

From a security perspective, the most complex task of a service provider is to manage the many different types of firewalls that typically exist in a server farm environment. These various firewalls are required to protect the different parts of the server farm from each other. Here are the security functions that need to be implemented in a server farm:

- Protect the customer's applications from hackers on the Internet
- Protect the different customers from each other
- Secure each customer's access into his or her network
- Secure the administration and management applications within the server farm from the customers

These functions are usually implemented in different firewalls. The goal of these multiple firewalls is to ensure that each customer has the right access to its own part of the server farm. Therefore, the best representation of the security policies in the server farm is provided by specifying each customer and the section of the ASP network that it is allowed to access.

Low-Level Policies

The different business needs outlined in the preceding section are satisfied by a variety of techniques. Many of these techniques can be used interchangeably. The choice of scheme depends on the specifics of the particular environment.

In order to support business SLAs, you can follow the approaches of capacity planning, Differentiated Services, or Integrated Services.

The basic idea behind supporting SLAs using capacity planning is fairly simple: Provide enough link bandwidth and processing capacity that the SLA requirements are satisfied. If an SLA is feasible at all (if it can be met on an unloaded network with unloaded servers), it should be possible to determine the link bandwidth and processing capacity that satisfied the performance objectives under normal operating environments. When the appropriate network has been designed, it can be operated as a best-effort network. Of course, the network needs to be monitored to ensure that the SLAs are being complied with.

There are many cases in which capacity planning might not be adequate. An example is the case in which loads on the network are not readily predictable or show a sudden spurt of growth. Situations in which capacity planning has failed are quite common on the

Internet, where many companies have launched an advertising campaign or hosted an event that caused their servers to become overwhelmed. In these cases, QoS techniques can help ensure that the performance of a subset of the users in the network can be maintained.

As described in Chapter 2, there are two main approaches to support QoS in IP networks—Integrated Services/RSVP and Differentiated Services. The RSVP approach can be loosely described as a signaled approach, and DiffServ is a provisioned multiclass approach. In a signaled approach, applications communicate (or signal) their QoS requirements to the network routers and remote workstations. Each router that is signaled reserves enough local resources (link bandwidth or buffer space) to support the application's QoS requirements. The other approach is to support multiple preprovisioned and differentiated classes of service in the network. These multiple classes are provisioned so as to deliver different levels of average performance. Different service classes have different expectations of average network delays and loss rates. With the provisioned multiclass approach, the network decides to map an application's packet flow into one of these preprovisioned differentiated classes of service and schedules them appropriately. A subset of Differentiated Services capabilities can also be used to provide rate control within the networks. Rate control devices can also be used to effectively control SLAs within the network.

In order to support the security needs in the different environments, you can use the protocols associated with IPsec or use the analogous transport layer scheme of SSL. Both of these technologies were described in Chapter 2. Either of these approaches can be used to support the different security needs within the different environments.

The next few sections take a closer look at the policy requirements of the different devices within each of the technologies.

Policy Issues with IntServ

The main issue with policy in an integrated services network is to try to answer these questions:

- Who is entitled to signal a reservation request using RSVP?
- Which requests should be honored by a router, and which ones should be rejected?

Because QoS mechanisms are intended to provide an assured performance level for a set of specific applications, their goal is to provide preferential treatment to those applications. These applications can obtain the desired performance by means of reservation. However, nothing prevents other applications in the network from invoking RSVP to reserve bandwidth to improve their own performance. Any application can signal that resources be

reserved for it. If no internal charge-back is associated with any reservation, there is no incentive to not ask for the maximum possible reservation that you can extract from the network. Obviously, a free-for-all reservation architecture is not likely to perform any better than a best-effort service. It can even perform worse. A user who is relatively sloppy at ending reservations might hog a large amount of bandwidth and never give any of it up.

The policy control module in RSVP decides who should be allowed to make reservations and also limits the number and duration of reservations that can be made. When reservation requests are received by the routers, they check the policy control module to ensure that the reservation should be honored. Thus, an enterprise can allow only reservations invoked by some key applications to succeed. Furthermore, it can also determine the amount of bandwidth that should be reserved by each flow belonging to the particular application.

Some routers in the network might not be capable of making policy decisions on their own. In that case, the routers can obtain policy decisions from an external policy server using a protocol called COPS.

When signaling for the reservation using the PATH message or RESV message, the endpoints involved in an RSVP flow can include a policy object as part of the message. This policy object can (among other things) identify the user, organization, or application requesting the reservation. The policy server can thereby enforce policy decisions at various levels of granularity.

Policy decisions can prevent someone from hogging resources or allow reservations to be made only by specific applications that are considered business-critical.

Policy Issues with DiffServ

A DiffServ network consists of two types of boxes—access routers and core routers. The access routers classify the various packets depending on the contents of the packet headers. This classification is marked into the DiffServ field of the IP header. An access router must know the rules which determine how different packets should be marked.

In addition to the marking, DiffServ access routers also can implement various types of rate control, limiting the amount of network bandwidth to be used by a particular type of traffic to specific limits. The policy definition for an access router needs to specify any such limits if they exist.

The core routers interpret the DiffServ field according to the set of PHBs defined for them. Thus, the policy definition for core routers must specify the type of queuing behavior that corresponds to different packet markings. Such a behavior can indicate the queuing priorities of the different network devices, as well as the rate limits or bandwidth shares that can be allocated to the different classes of traffic.

With the availability of any level of differentiation, you have to decide who or what gets which class of service. The answer to this question constitutes policy in DiffServ networks. In order to manage the performance of a DiffServ network, you must obtain the configuration information for all the DiffServ access routers so that the classifiers and rate controllers at DiffServ boundaries can be managed to meet expectations. Similarly, the core routers that make up a DiffServ network must be configured to have the correct configuration corresponding to that of different applications.

Communication in any network is bidirectional, and improving the quality of communication requires improving performance in both directions. Thus, trying to improve the performance of a specific application session would require configuring at least two access routers (plus the core routers that lie along the path). Coordinating a consistent configuration of multiple access routers is not a trivial task. The goal of the policy management tools described in Chapter 5, "Resource Discovery," is to ensure such a consistent configuration in the various network configurations.

Policies and Device Configuration

There is a subtle but important distinction which needs to be made between the notion of a device configuration and the low-level policies associated with a technology. The low-level policy definition for DiffServ consists of the rules that determine the behavior of the network and devices in a manner that is independent of the details of a specific device. These rules are represented in a format that can be understood and interpreted by any of the devices within the network. As an example, such policies may be represented in an LDAP directory using a commonly accepted schema. Also, these policies may be specified for a group of devices (or for the entire network), rather than for each device individually.

Corresponding to the policy specification, each device can generate its configuration which implements the set of policies which are relevant to it. The semantics of the configuration must match the semantics of the low level policies.

Thus, there are two important differences between the low-level policies and device configuration: representation, for example, where policies are represented in a device-independent manner; and scope, for example, where policies are applicable to more than one device.

This book discusses an application of the policy technology, namely how to get all the devices in a network configured in order to meet some high level goals. The primary use of low level policies for this application is generating the device configuration. As a result, the line between device configuration and low level policies may appear blurry at times, but the reader should keep in mind that the two are different.

Policy Issues with Servers

Providing adequate performance within any environment depends on ensuring that performance is assured on all parts of a system, including the clients, the network, and the servers. Thus, when QoS features are being used within the network, they need to be augmented by similar functions within the servers. In some specific environments, such as the ASP environment, server controls might be more important than network controls.

If the server operating system supports any notion of different levels of service offered to different applications, that service-level information must be encoded into the appropriate configuration for the server platforms. The set of priorities that are needed to manage the performance of the various applications must be specified in some manner in the server configuration.

In cases where server differentiation mechanisms such as support different priority levels are available, the appropriate configurations for the various platforms need to be generated. These configurations must include the appropriate performance priorities (or other suitable information) for the different classes of applications that are supported on a given server or cluster of servers.

Policy Issues with IKE/IPsec

The policy issues associated with IKE involve defining the set of parameters that specify how secure communication using IKE is to be implemented. A typical IKE configuration is specified in terms of the characteristics of the Phase 1 and Phase 2 tunnels that need to be established in order to exchange the keys required for IPsec communication.

The typical IKE configuration consists of specifying three types of records:

- **Phase 1 characteristics.** This defines the characteristics associated with a Phase 1 security association of IKE. These characteristics define how long a key used in Phase 1 would be valid and the type of authentication mechanisms used by the communicating parties to validate each other. Two common techniques that can be used for authentication are the use of a shared common secret and the use of public certificates. With a shared common secret, both parties in the IKE establish a secret key that they use to identify each other. With public certificates, they both trust a certificate-issuing authority that can be used to obtain the public keys of the other party.
- **Phase 1 transform lists.** During the Phase 1 negotiations, the communicating parties discuss a list of encryption and authentication algorithms that they would be willing to accept in communication over a Phase 1 security association. This

transform list would be used to secure the exchange of keys for establishing Phase 2 security associations. A transform list would indicate whether encryption or authentication or both should be used for the communication, and which algorithms should be used for this purpose.

- **Phase 1 tunnel descriptions.** This specifies which phase 1 characteristics and phase 1 transform lists should be used for communication between a pair of source and destination machines. The granularity of the source and destination can be further refined by the use of port numbers at the source and destinations.

> **Note**
>
> Here's a quick note on terminology: What I call tunnel descriptions are usually referred to as policies in the IKE/IPsec implementation and RFCs. Because this usage might cause some confusion with the definition of policy I have been using all along, I have opted to call these tunnel descriptions.

The other three types of records are the corresponding Phase 2 characteristics, Phase 2 transform lists, and Phase 2 tunnel descriptions. There are differences in the exact set of characteristics that is specified among the two phases, or for the transform lists that make sense in the two phases of communication.

It is probably apparent to you by now that configuring the IKE correctly is a daunting task. It doesn't help that the configuration must be done not for one firewall, but consistently across multiple firewalls in order to enable some business needs.

Policy Issues with SSL

The typical SSL configuration for the bidding client or bidding server application consists of parameters such as the type of authentication that should be used for the different communication types—such as based on shared secrets or public certificates. Furthermore, the configuration should indicate whether only the server is authenticated or if both the client and the server are authenticated.

Other SSL parameters, such as when the security keys should be renegotiated, also need to be specified as part of the SSL configuration.

Although SSL configuration is simpler than the corresponding IKE configuration, it is essential that the configuration be consistent across the bidding client and bidding server applications so that the establishment of the secure SSL connection succeeds.

The Policy Management Tool

As must be apparent to you at this stage, the policies described in the section "High-Level Policies" and the policies described in the section "Low-Level Policies" of this chapter have very little in common. There exists a large gap between the business needs of an enterprise/ISP/ASP and the technologies that are required to satisfy them. Clearly you need to develop solutions that bridge the gap. One possible way to bridge this gap is by exploiting the IETF policy architecture and the policy definition framework.

Note

The policy definition framework is being done jointly within IETF as well as DMTF. Because it is network-centric, I will refer to it as the IETF architecture. The IETF does the bulk of the architecture, while the DMTF does the bulk of schema definition.

To recap the IETF policy architecture introduced in Chapter 1, it consists of four components: a policy management tool, a policy enforcement point (or the policy consumer), a policy decision point (or the policy target), and a policy server. The management tool populates the policy server, the policy decision point takes policies from the server to determine the configuration and interprets it, and the policy enforcement point enforces that decision.

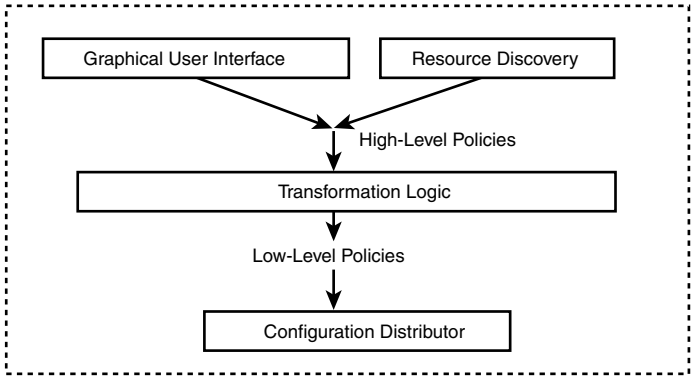
One of the key components in the policy architecture (and the various scenarios just described) is the policy management tool. The *policy management tool* is the component that translates the specified requirements in terms of the business needs of the deployers into the detailed specifications needed for technology deployment. This section describes how such a policy management tool can be constructed. The rest of this book describes the various components of such a management tool in much more detail.

The purpose of the policy management tool is to take a high-level representation of the business goals or desired functions within the network and translate them into the appropriate configuration of the various devices in the network. The input is called the *high-level policies,* and the output of the tool is the *low-level policies* in the network. Such a tool is composed of the four basic functions shown in Figure 3.6:

- The *graphical user interface* is the means by which an administrator can input the high-level policies within the network. These policies are the ones that need to be enforced in the network, and the configuration of the network must be made conformant to these high-level policies.

- The *resource discovery* component determines the topology of the network and the users and applications that are operational in the network. In order to generate the configuration for the various devices in the network, the capabilities and topology of the network must be known. For any moderately sized networked system, such topology and capabilities must be discovered automatically.
- The *policy transformation logic* component is responsible for ensuring that the high-level policies that are specified by the network administrator are mutually consistent, correct, and feasible with the existing capacity and topology of the network. It also translates the high-level policies into low-level policies that can be distributed to the different devices in the network.
- The *configuration distributor* is responsible for ensuring that the low-level policies are distributed to the various devices in the network. The distribution of the configuration can be done in a variety of ways, such as storing them in a repository from where different devices can retrieve them.

Figure 3.6 Policy Management Tool



In addition to these components, there needs to be an additional component that is responsible for monitoring the network state and ensuring that the policies are being satisfied by the various devices in the network. For performance SLA, such monitors take the form of the different SLA monitoring tools that are available in the marketplace. For security policies, such monitors take the form of different "watcher" products that are responsible for detecting intrusion and other abnormal behaviors in the network traffic.

The next sections discuss each of the modules in turn.

The Graphical User Interface Module

The graphical user interface module is the component of the policy tool that provides the look and feel of the policy management tool to the end-user. Its purpose is to allow an administrator to enter the high-level policies to the management system. The user interface is largely responsible for making the task of entering policy information simpler. The input from the user is generated into an internal representation of the high-level policies that are to be processed by the policy validation module.

The design of such a user interface must take into account the factors that affect the experience a user would have. In particular, the tool must provide appropriate help screens and useful diagnostics messages when exception conditions are encountered in the tool.

Two common ways of providing the user interface module are via a command-line interface or via a graphical user interface. The command-line interface is preferable when the management tool needs to interface with other automated programs. The graphical user interface is useful to interface with a human administrator. With the increasing emphasis on Web-oriented management, the user interface (in many instances) is implemented as a program that can be accessed via a browser. One way to achieve this goal is to implement the user interface as a Java applet. Other methods include creation of specific Web pages and server-side programs that let a user input the desired high-level policies.

The user interface is a very important component of any practical management tool. Many tool developers consider it the most important component of the management tool. However, for the goal of this book, which is to discuss the algorithms and applications of the policy architecture, the user interface module is just a way of obtaining high-level policies. The focus of this book is on processing high-level policies, so more attention is given to the other modules in the management tool. I will not discuss this module further in this book, but I want to reemphasize that this component is the one that can most dramatically affect a successful adoption of any tool for policy management.

The Resource Discovery Module

The first component of the policy management tool that I will discuss in detail in subsequent chapters (Chapter 5) is the resource discovery module. The purpose of this module is to obtain information about the different types of devices that are active within the network, their capabilities, and their characteristics that might affect policy generation and management.

The resource discovery module is needed in any practical policy management tool. The goal of policy management is to simplify the task of system/network administration. If this management must be done, the tool needs to maintain a current snapshot of the various

PEPs and PDPs that are operational within the network. Because it is not possible for any administrator to correctly enter the topology and configuration of any moderately sized network manually, these characteristics must be discovered by the tool automatically.

Several of the resource discovery capabilities are included in many traditional systems management/network management tools. Such resource discovery capabilities include SNMP-based tools to collect network routing and topology information, as well as many inventory tools that can obtain a summary of all the applications that are installed and active on a server or a desktop. When a policy management tool is included as a component within a larger systems management/network management suite, it can leverage the capabilities of the existing tools in the suite.

I will discuss the issue of resource discovery in more detail in the next chapter.

The Policy Transformation Logic Module

The policy transformation logic module validates the information provided in the high-level policies and transforms them into the configuration of devices in the network. The logic furthermore ensures that the policies specified are mutually consistent and that they cover all aspects of interest to the network administrator.

The validation process must incorporate syntactical checks as well as semantic checks. The semantic validation of high-level policies consists of the following three types of checks:

- **Bounds checks.** This validates that the values taken by any parameter in the policy specification are within specific constraints that are determined by the network administrator. For example, a network administrator should be able to specify that all response times in any defined class of service be less than 1000ms.
- **Relation checks.** This validates that the value taken by any two parameters in the policy specification are within constraints that make sense. For example, two attributes of a class of service are response time and the duration over which the response time must be measured. The latter must be larger than the former, and the network administrator should be able to specify how large the response time should be.
- **Consistency checks.** These checks ensure that each traffic flow is mapped onto exactly one service class, and that each service class is properly defined at all the interfaces. These checks are applied in the manner described next.

After validating that high-level policies are consistent and well-formed, the transformation logic translates them into technology specific low-level policies. These low-level policy definitions may be grouped so that only one set of policy information is generated for many

boxes that need identical policy information. In general, boxes that play a similar role in the network (for example, access routers as opposed to core routers in network) are likely to have the same set of policy rules guiding their behavior.

Policy validation logic is discussed in more detail in Chapter 6, "Policy Validation and Translation Algorithms."

The Configuration Distributor Module

As soon as the low-level policies (which drive the configuration of different devices) are generated by the policy management tool, they need to be distributed to the different devices within the network. Several means of distributing devices are possible:

- **Populating a repository.** The management tool can write the device configuration rules into a configured repository in the network. Individual servers and routers pull the policy information from the repository, use the policy information to generate their local box configuration and subsequently configure themselves. The preferred approach in IETF policy working group has been to define a LDAP directory with a standardized schema as such a repository. In this approach, the low-level policies are stored at the repository, and the configuration is generated individually by each of the participating devices.
- **Distributing configuration files.** The management tool can translate the low level policies into the configuration files that would be needed at each device and copy them remotely over to the appropriate router and server. This approach works for all types of devices and does not require any specific software to be running at the device. However, the management tool must understand each type of device and the format of configuration file that can be used with it. The distribution mechanism is available in many systems management products. In this approach, the translation from low-level policies to box configuration is done by the management tool.
- **Command-line interfaces.** Most routers permit remote administration by means of a telnet session and specific command lines. The QoS management tool can use automated scripts to specific routers and control the configuration of the router using commands specific to the router. As in the case of configuration files, the QoS management tool must translate the low level policy definition to the configuration scripts, and must understand the scripts that can be used for different types of routers.

Further details and a comparison of the different distribution approaches are discussed in Chapter 7, "Policy Distribution Mechanisms."