1

Large Enterprise Networks

Modern networks are divided, in terms of their operations, into essentially two main categories: enterprise and service provider (SP). In this book we focus on the principles and concepts of managing large enterprise networks. Examples of such networks are government departments, global corporations, and large financial/healthcare organizations. Most such enterprises employ the products and services of SP networks, so we try to balance the discussion by including some general comments about managing SP networks as well. It is in the latter network type that we tend to describe Multiprotocol Label Switching (MPLS), a widely deployed technology. In passing, we mention that MPLS is also finding its way into large enterprise WANs.

An important point to note is that network management is a distinct and separate discipline from both enterprise and SP networking. For this reason, our study of enterprise, SP, and MPLS network management should be seen merely as applications of network management technology. As we'll see, many elements of network management are common across all such application areas. We have six main aims:

1. To illustrate some important aspects of network management, especially enterprise networks but also SP networks.
2. To describe some increasingly important problems facing Simple Network Management Protocol (SNMP)-based network management systems (NMS[1]).

3. To describe some Management Information Base (MIB) improvements that would assist manageability.
4. To illustrate the construction of a rudimentary NMS using Visual C++ and Java.
5. To describe MPLS and the advantages that it provides to enterprise and SP networks.
6. To illustrate the need for increased (policy-based) intelligence in managed devices.

We set the scene by describing in general terms some of the components of large enterprise networks. These networks are big and geographically dispersed (often spanning many countries), have lots of legacy equipment, and are hard to manage—scalability is an issue affecting both their manageability and usability. After introducing the general area, we begin our discussion of network management.

Generally, enterprise networks are owned by a single organization, such as IBM, federal government bodies, and financial institutions. These networks exist to provide data and telecommunications services to employees, customers, and suppliers. Services can include:

- File and data storage
- Print
- Email
- Access to shared applications
- Internet access
- Intranet
- Extranet
- E-commerce
- Dial tone
- International desk-to-desk dialing (using voice-over-TDM or voice-over-IP)
- Video

1. The term *NMS* is used throughout this book. Depending on the context, it may be either singular or plural: "an NMS is…" or "many NMS provide this feature…." Rather than using the unwieldy NMSs, we opted for just NMS and let the context indicate either singular or plural.

- LAN and virtual LAN (VLAN)—often heavily overengineered (more bandwidth than necessary) to avoid congestion
- Corporate WAN—can be used for data and also voice-over-IP
- Virtual private network (VPN)—can be used for securely joining multiple sites and remote workers and replacing expensive leased lines
- Disaster recovery—maintaining network service after some cataclysmic event

Enterprise networks achieve these and other services by deploying a wide variety of different technologies and systems. Some services encompass several technologies, such as voice-over-IP (VoIP) [Tanenbaum2003], which can be transported over a WAN link to achieve toll bypass or migration away from overlay networks (for voice, video, and data).

An enterprise[2] uses its network as a means of providing or improving business processes and saving money rather than as a vehicle for profit. This mindset influences enterprise decisions to deploy solutions like VoIP telephony. The guiding principle is service enhancement and business advantage rather than reductions in spending (though the latter is also extremely important). In fact, some global organizations are so big that they can often negotiate reduced tariffs with their local telecommunications carrier—in many cases a quicker and easier way to save money than rolling out expensive, complex, new technology. It may be more important for an organizational department, such as a provider of frontline PC support, to direct a minimum of incoming phone calls to voicemail. This can influence the decision to deploy in-building mobile telephony (e.g., IEEE 802.11 a/b, DECT in Europe) so that call handling is not restricted to the desk phone. In other words, service levels are enhanced because calls are less frequently routed to voicemail.

2. An enterprise may decide to outsource or even sell its entire network infrastructure to a third party. The enterprise can then lease back the network from the new owner. The net effect is that the enterprise outsources much of its network management workload and exchanges a depreciating asset for cash. The same type of sale and leaseback can be done on pretty much any type of asset, including buildings. The buyer leases the asset back to the enterprise for a fixed annual outlay. The burden of ownership then resides in the leasing company. The merit of doing this with network hardware is that the enterprise usually gets good terms for upgrades.

Figure 1-1 illustrates a typical simplified enterprise network. Figure 1-1 is highly simplified in order to give us a flavor of enterprise networking issues. Real enterprise networks tend to feature additional technologies, such as Asynchronous Transfer Mode (ATM), VLANs, broadband connections, and redundant configurations. Later (in Figure 1-4) we will see a portion of an enterprise network realized using VLANs.

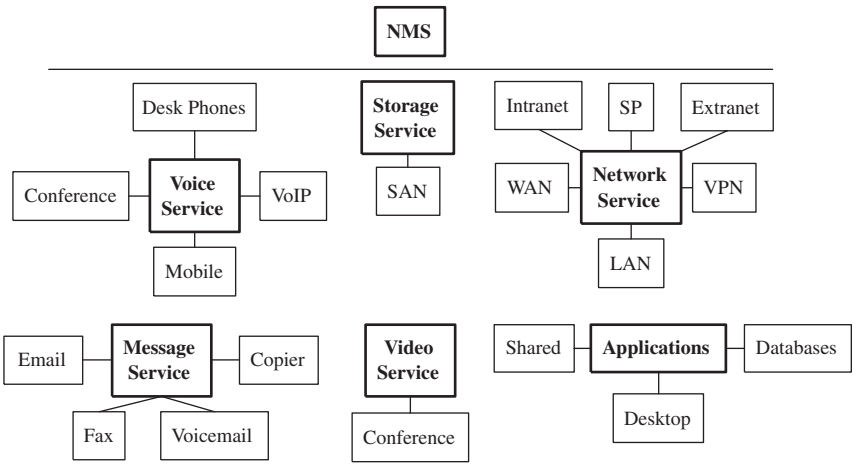

Figure 1–1 Enterprise network functional components.

All the boxes with bold text and borders in Figure 1-1 provide some type of service—for example, Voice Service. The connected boxes provide access to the service—for example, VoIP phones (in the VoIP box). In fact, the network in Figure 1-1 can serve a large, geographically distributed corporate user population. Alternatively, Figure 1-1 might be a corporate headquarters with hundreds of remote branch offices. It's easy to see why the provision and management of enterprise networks are so critical to modern organizations.

The networks and systems in Figure 1-1 add value to the organization, and later we'll see how the enterprise network managers (in many cases, IT groups) can play an important role in assisting the developers of network management software. In this way, IT initiatives are closely aligned with broader business objectives [EnterpriseIT].

Also noteworthy (as mentioned above) is the use of IP phones in a LAN environment, reducing the need for legacy PABX equipment and prompting migration to a packet-based infrastructure. The migration to layer 3 mentioned

here is discussed in Chapter 2, "SNMPv3 and Network Management," and is a recurring theme throughout the book.

One point about Figure 1-1 is that many or all of its components may be repeated on other sites linked to this one via a WAN. These other sites include normal branches of the organization as well as unmanned backup sites. This means that essentially the same corporate services are offered to all employees regardless of their location, whether it is in New York City or the West of Ireland. Many organizations fund this type of arrangement by charging a straight percentage from the revenues of each local site. Also, different sites can offer services, such as audio conference bridges, to other sites. In this case, the site hosting the bridge bills the users dialing into it from remote sites. There are many reasons for using this geographically distributed approach to enterprise network deployment:

- Expensive systems, software applications and licenses can be shared across time zones.
- Valuable data, such as subscriptions to ETSI and ITU, can be shared.
- Remote sites can help the company gain access to specific local markets.
- Access can be gained to specialized labor skills, such as software development or manufacturing.
- Organizations can take advantage of different tax regimes to improve revenues.
- Some configuration can be handled from a central location—for example, PABX maintenance can be carried out by a centrally located specialist team.

Notable features of Figure 1-1 are the incorporation of separate networks for storage (i.e., storage area network, or SAN), WAN, SP networks, and telephony. SANs provide access to data storage facilities. WANs provide access to remote network facilities. SP networks provide Internet access (among other services), and the Public Switched Telephone Network (PSTN) provides access to the global telephony networks (fixed and mobile). Typically, an enterprise will use several service providers, each providing one or more of the above services.

The enterprise network enables access to a wide variety of devices and services. The important point about the structure depicted in Figure 1-1 is its flexibility: Large numbers of users can share the corporate, productivity-enhancing

services using a wide range of access methods. By this means, an employee working from home can be at least as effective as one based in the office without the need for commuting. Similarly, sales staff can access (e.g., via a VPN) the enterprise network during business trips.

Another trend is unified messaging for integrated access to email, voicemail and fax mail messages using an email client. PCs can also be used for access to videoconference broadcasts and even videophone calls. Audio conference calls can also be accessed via unified messaging or by using a desk phone. Some organizations even use broadcast voicemail to make important announcements. Another aspect of enterprise networks is linkages between desktop calendars and the reservation of meeting rooms. Rooms are booked and invitees are reminded via their email client.

Intranets provide official enterprise information channels for employees. Many organizations use intranets for posting important information such as product announcements and corporate media coverage. Another intranet facility is integration of productivity tools such as document management systems. In this sense, the intranet becomes just another desktop tool accessible using a Web browser. As we'll see later, the Web browser is often an indispensable part of an NMS.

Enterprise data flows can become very complex once extranets and e-commerce are employed. Extranets are parts of intranets that are extended to organizations external to the enterprise, such as software contractors. E-commerce allows for secure financial transactions between external customers and a given organization. The data flows in the latter case feed into various systems, such as finance, stock control, and manufacturing.

Other important aspects of maintaining secure enterprise networks include:

- Automated software distribution (e.g., of anti-virus software)
- Policy setup (e.g., auto-logout after a specified interval of nonactivity)
- Software application license checking

Many organizations distribute enterprise software in a centralized fashion, for example, using Microsoft Systems Management Server. This can include defensive procedures such as anti-virus software updates. Likewise, productivity software such as word processors and spreadsheets can generally be updated in the same way. Many end users of enterprise systems tend not to log out, so policies

can be applied to host machines that will log the user out after, say, 15 minutes of inactivity. This can be done for security reasons and also in order to update anti-virus software once the user logs back in again. A full virus scan can then occur at night. The important area of software license checking can also be handled remotely to verify that the number of end users who have installed software packages does not exceed the license limit.

These various uses of enterprise facilities clearly illustrate the power of the underlying network. Following are some general features of enterprise networks:

- They incorporate a wide range of multivendor devices, such as routers, switches, exchanges, PCs, servers, printers, terminal servers, digital cross-connects, multiplexers, storage devices, VoIP telephones, servers, and firewalls.
- Network elements (NEs) can incorporate other intelligent devices, such as PCs with network interface cards (NICs) and possibly modems. Likewise, desk phones can contain computer-telephony integration (CTI) hardware for applications like call centers and e-commerce bureaus.
- Individual NEs provide a variety of different shared services; for example, a legacy PABX or a soft switch provides basic telephony and can form the foundation of a call center. In this way, a base system is leveraged to provide another system or service.
- Backup and restore of NE firmware are important for rolling out new network services.
- Specialized servers are deployed to provide advanced services such as SANs.
- Many users are supported simultaneously.
- The overall network services, such as email and video/audio conferencing, are used by employees of the organization as essential business process components.

Enterprise systems and networks all have individual lifecycles comprised of:

- Planning
- Deployment
- Operation and management
- Retirement, replacement, or upgrade

In this book we focus mostly on network operation and management, but the other lifecycle stages are equally important. An example of this is a SAN in which the following steps typically occur:

- Planning the required storage capacity, server links, and network connection
- Deploying the SAN in a production environment
- Operation and management of the SAN: discovering SAN components in a vendor-independent fashion, monitoring faults, checking performance, and backup and restore
- Extending the SAN as storage requirements grow

Growing storage requirements in enterprises can have the effect of reducing backup time windows. This and other storage issues may cause loss of service and require that administrators deal with problems such as:

- Devices going offline
- Capacity being exceeded
- Performance degradation
- Application software with rapidly increasing storage requirements

All of these require some type of reactive (after the problem has occurred) manual intervention. Clearly, there is a relationship between storage planning and the incidence of storage capacity being exceeded. The same is true for the ever-increasing storage demands of application software. Network administrators need tools to help them balance these dynamic requirements. Where possible, the NEs should be engineered to facilitate this type of advanced management. In conjunction with NE-resident self-management capabilities, there is a need for high-quality management systems. The latter should then provide features that match the organizational workflows (broadly speaking, these are plan, construct, and operate).

Another very common enterprise technology is the VLAN. Many organizations employ VLANs in order to provide a switched layer 2 infrastructure with designated broadcast domains. A broadcast domain is a set of layer 2 devices with a defined boundary (typically an IP router) beyond which broadcast traffic will not flow. For example, an organization could group the NEs on each floor of a building into a different VLAN (i.e., broadcast domain). All of these floor-level VLANs could then be connected to a single high-speed switch that is in turn connected to another set of VLANs. One of the merits of VLANs is scal-

ability—to add more devices, you can just create another VLAN. This helps to avoid the problem of running out of broadcast domain capacity on a single medium (such as a large Ethernet network).

Building and operating VLANs can be carried out using either an element management system (EMS) or an NMS. A typical workflow for adding a new PC to VLAN X is as follows:

- Physically connect the host PC to a port on the switch containing VLAN X.
- Using the switch element manager, add the port to VLAN X.
- Specify no tagging (the legacy case), that is, the PC NIC adds no IEEE 802.1p/Q fields to its Ethernet headers (these are two fields contained in the Ethernet frame header: 3 bits for priority and 12 bits for a VLAN ID value).
- Verify host PC connectivity (by logging into the network, pinging servers, etc.).

As far as possible, the NMS—or EMS in this case—should facilitate this type of workflow. For example, when adding a port to a VLAN, only options appropriate to that hardware should be presented. So, if a port does not support 802.1Q, then the EMS/NMS should not present an option to set a VLAN ID. This information can be acquired by the EMS/NMS (via automatic dialog with the NE) and greatly assists in managing such devices.

There is a downside to the rich environment provided by enterprise networks. They are expensive to build and run, and they require skilled maintenance and support personnel. Traditionally, the network support effort (excluding PC support) has been divided into two camps, data networking and telecommunications, but these two areas are rapidly converging. PABX technology is gradually being phased out and replaced by server-based solutions [CiscoVoIP]. Multiple incompatible networks for voice, video, and data are gradually being migrated onto a packet-based infrastructure.

Many organizations seek to centralize servers in secure locations and then lease WAN lines from there to branch offices and divisions. This reduces remote site support but increases dependency on communications lines, an increasingly cheap commodity [GlobalCross2002]. Services are resolving down to the process of transporting bits from location X to location Y over a single physical network.

Managing Enterprise Networks

Why is enterprise network management important? First, it helps keep the overall network running—end users are kept happy and the business processes are not blocked by downtime. Second, good network management facilities assist in all the lifecycle stages. Third, such facilities should help to reduce the cost of running the network. This last point is particularly important during periods when IT budgets and staff numbers are cut.

An important issue concerning enterprise networks is the presence of multiple incompatible management systems. While expensive resources are shared using the underlying network, these resources are generally not centrally managed in a technology-independent fashion. An example of this is the SAN facility shown in Figure 1-1. The individual components of SANs (disk subsystems, network switches, and SAN servers) typically each have a dedicated management system. This substantially adds to the cost of ownership. Gartner Group Research claims that the cost of managing storage is five to seven times the price of purchasing storage [NovellSAN]. Generic enterprise management systems, such as HP OpenView, already exist, but not all of the networked systems (such as in Figure 1-1) have the necessary infrastructure that would allow them to be managed in an integrated fashion. An illustration of this in Figure 1-1 occurs if one of the digital phone cards in the PABX (the Voice Service in Figure 1-1) fails. If the PABX does not emit some type of message to this effect, then the desk phones connected to the card in question will lose service until the problem is fixed. Likewise, if a WAN access switch fails, then the WAN connection may be lost. If there is no integrated NMS in place to detect and signal these types of problems, then service loss will occur until the problems are reported and fixed.

It is a central theme of this book that the vendors of as many systems as possible should include SNMP (preferably version 3) management capability as a priority. This would allow for all managed elements to emit traps (or messages) as soon as a problem occurs. The necessary minimal components required for making a system manageable are:

- MIBs
- Agents/entities—hosted on network devices to provide management facilities
- Scripts for manipulating MIB objects
- Java/C/C++ software modules for manipulating MIB objects

MIBs provide a detailed description of the managed data objects. Typically, the description of each MIB object consists of:

- Accessibility (read-only, read-write, not-accessible)
- Status (mandatory, deprecated)
- Description

Agents (or entities in SNMPv3) are software components that implement the MIB and map the objects to real data on the NE. It is the agent's job to maintain, retrieve, and modify MIB object instance values. The network manager delegates this important task to the SNMP agent. The agent also emits special messages called notifications to signal the occurrence of important events, such as a device restarting or a network interface going down. Finally, the agent must implement all of this using some preconfigured security scheme ranging from simple passwords to stronger techniques involving authentication and encryption.

On the manager side, it is important to be able to manipulate the various agent MIBs. This can be done using scripts or via binary software modules built using various programming languages such as Java/C/C++. In either of these two cases it is often necessary to load the associated agent MIB module files into a management application. An example of this is a MIB browser: an application that allows for MIB objects to be viewed (some browsers allow for MIB object instances to be modified). Most MIB browsers merely require MIB module files to be loaded; that is, they are preconfigured with the necessary SNMP protocol software.

Another very important topic is the management of both newly commissioned and legacy NEs. It is rare (particularly during periods of economic recession) for large networks to have forklift upgrades in which the very latest NEs are deployed in place of legacy devices. Normally, new NEs are added and old ones are replaced. For this reason, one can expect a rich mixture of deployed devices, both old and new. This generally also means a complex set of MIBs deployed across the network. As we'll see, this can result in problems related to the support of backwards compatibility (a little like saving a word-processed document using version 4 and then experiencing problems opening the document with version 3 on your laptop).

MIBs provide the managed object definitions (type, structure, syntax, etc.) for the underlying system; for example, a terminal server may implement the following principal managed objects:

- Serial interfaces
- Serial interface attributes such as bit rate, word size, and parity
- IP address

To provide baseline SNMP management for a terminal server, the relevant MIB must be consulted for the requisite managed-object definitions. The instance values of these objects can then be looked up using a MIB browser. The SNMP software modules (along with the MIBs) can be integrated into a management system and used to monitor and configure the associated agent. This approach (using SNMP) obviates the need for a proprietary management system. More details on the topic of terminal-server serial-interface MIB objects can be found in Appendix A, "Terminal Server Serial Ports." Later, we'll see that the quality of the MIBs has an important bearing on the manageability of a given NE.

Figure 1-2 illustrates a different view of an enterprise network.

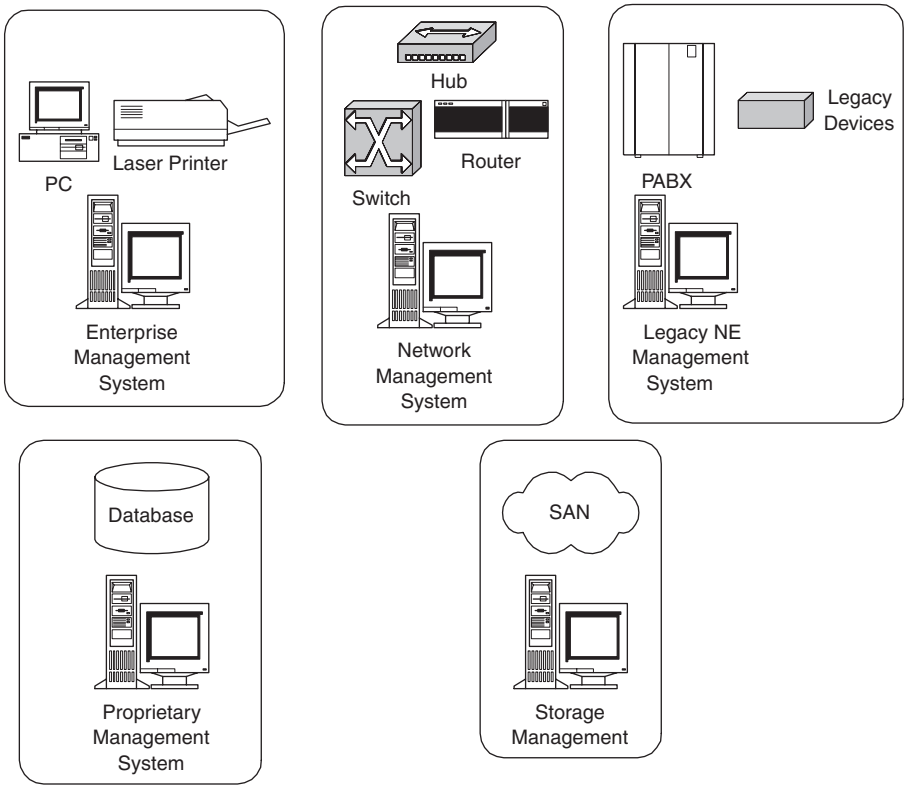

Figure 1–2 Enterprise management systems.

In this diagram, the NEs are grouped alongside their associated management systems. The multiplicity of management systems is one of the reasons why enterprise network management is so difficult. This is what we mean by multiple incompatible management systems: Problems in a device attached to the PABX are not reflected back to the enterprise network manager. Instead, they register by some proprietary means in the legacy NE management system (if one is deployed), and it is up to IT to discover and resolve the problem. Many smaller devices (such as terminal servers) support only a simple text-menu–based EMS or command-line interface (CLI). The absence of SNMP agents (or the deployment of only SNMPv1) on these devices contributes to making them difficult to manage in an integrated, vendor-independent, and centralized fashion.

In order to manage enterprise networks as seen in Figure 1-2, it is necessary to learn all of the deployed technologies as well as their proprietary management systems. This is an increasingly tall order. In many organizations, the management facilities consist of simple scripts to configure and monitor devices. While many enterprise network managers may implement ingenious script-based facilities, all such solutions suffer from being proprietary. An added problem is seen when the author leaves the organization—the requisite knowledge often leaves at the same time. Adoption of standards-based network management technology helps in avoiding this. Standards-based consolidation of management systems can help enterprises to achieve the following:

- Fewer and simpler user interfaces for managing networked systems
- Reduction in the time required for IT staff training
- Faster resolution of NE problems, such as switch interface congestion

A single (or a reduction in proprietary) management technology in the network contributes to making that network easier (and cheaper) to operate and maintain. It is for this reason that we say as many as possible of the components of enterprise networks should implement SNMPv3 agents (or entities, as they are called). Figure 1-3 illustrates a modified enterprise network with SNMPv3 entities deployed in the SAN, legacy NEs, and in the switch/router/hub NEs.

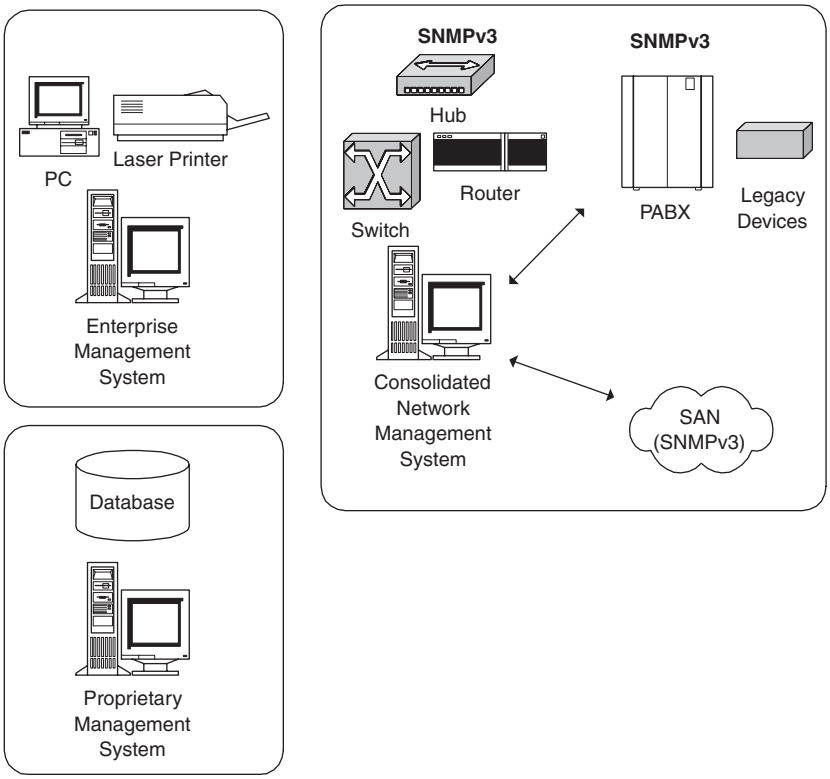

Figure 1–3 Example of consolidated enterprise NMS.

If all of the NEs deploy SNMPv3 entities, then it is possible that one or more of the proprietary management systems (in Figure 1-2) can be removed and consolidated into one NMS. Of course, it's not so easy to just add SNMPv3 capability to all of these NEs (particularly the legacy NEs). The point is that it has a substantial benefit.

The other enterprise systems in Figures 1-2 and 1-3 (the networked PCs, print servers, and database management system) generally tend not to deploy SNMP for their management and operation. This is largely for historical reasons. Since this book is about network management rather than system management, we do not consider this area any further. However, before moving on, we should say that there are no major reasons why SNMP technology should not be used for managing such systems.

Manageability

For a number of reasons, not all NEs lend themselves to flexible, integrated, centralized management. This tends to add to the cost of ownership and arises for a range of reasons:

- The NE is a legacy device with proprietary management infrastructure.
- The NE implements only SNMPv1 *with* support for set operations (a set operation is an update to a network-resident managed object).
- The NE implements only SNMPv1 *without* support for set operations.
- The NE supports SNMPv3, but it has been poorly implemented.
- The NE supports SNMPv3 but has a number of low quality MIB modules.

Proprietary management infrastructure may consist of just a simple CLI with no SNMP deployment. It is difficult and costly to incorporate these NEs into an NMS because customized software must be written to send and receive messages to them. NEs that support just SNMPv1 and set operations are generally felt to be a security risk (because the relevant password is transmitted across the network as clear text). As a result, no set operations may be allowed. Configuring such NEs is usually achieved via CLI scripts. While this is a fairly standard approach, it negates some of the benefits of using an NMS, such as security, audit trails, and GUI-based help facilities. Much the same applies for those NEs with SNMPv1 and no set operation support. Configuration must be achieved using non-SNMP methods.

Poor implementation of SNMPv3 might consist of low resource allocation (process priority, message buffers, caching, etc.) with the result that the management system regularly gets choked off. This may be seen during periods of high device or network loading (often the time when network management is most needed).

Badly written MIBs are the bane of the NMS developer's life. We'll see examples of good MIB design later on, but for now we illustrate this with a simple example of adding a new row to a table indexed by an integer value. To add a new row to this table, a new index value is required. Often, MIB tables do not implement a simple integer object to store the value of the next free index. This may require a full (expensive) walk of the table in order to calculate the next free index. This is inconvenient when the table is small (less than 100 entries), but when the table is big (many thousands of entries), a MIB walk becomes an

expensive option because of the number of agent operations and the associated network traffic. The inclusion of a specific index object to facilitate new row addition can greatly assist the management system. We will see all of these considerations in action later on.

In summary, an NE is considered to have good manageability if it supports a well-implemented SNMPv3 agent and a high-quality MIB.

Operating and Managing Large Networks

Running networks such as the ones described above is difficult. The growing range of services offered to end users means that traffic levels are always increasing. Deploying more bandwidth can offset rising traffic levels but, unfortunately, the nature of this traffic is also changing as the associated applications become more resource-intensive and mission-critical. This is seen in Figure 1-1 with LAN-based voice, video, and data applications, which (except for data applications) impose stringent timing requirements on the network. Some way of guaranteeing network transport (and NE) availability is needed, and best-effort IP service in the long run is probably insufficient for large, distributed enterprises. This is one of the biggest challenges facing all network operators—how to provision bandwidth-intensive, time-constrained applications on layer 3 networks. Many enterprises and SPs have used overengineering of the network core bandwidth to cater to increased traffic levels. This is ultimately not scalable, and later on we examine the solution MPLS offers to hard-pressed network operators. It is increasingly important for the network to provide defined quality of service levels for traffic.

Some important aspects of enterprise network management are:

- Availability of NEs, interfaces, links, and services
- Discovery and inventory management
- Monitoring the status of NEs, interfaces, links, virtual circuits, VLANs, and so on
- Measuring traffic levels and checking for network congestion
- Configuration—VLAN setup, SAN volume setup, storage allocation, remote-control software (Microsoft Systems Management Server), and database redundancy (e.g., Informix)

- Service level agreement (SLA) reporting, SLA verification between an enterprise and SP
- Security control—resistance to attacks from both sides of the firewall
- Scalability—handling increased numbers of users, traffic, NEs, and so on
- Disaster recovery

We will cover many of these topics. In the next sections we look at those OSI network layers of greatest relevance for the forthcoming discussions.

Layers 2, 3, and 2.5

Reference is made throughout this book to layer 2 and 3 devices [Puzmanova2001]. Some confusion seems to surround the use of these terms both in the industry and in the literature. Issues affecting layers 2 and 3 on enterprise networks are a recurring theme throughout this book. Our use of the terms layer 2 and layer 3 follows the guidelines of the OSI model. A layer 2 device is one that operates no higher than the data-link layer—for example, ATM, Frame Relay (FR), and Ethernet switches. The basic unit of transmission at layer 2 is the frame (or cell for ATM). A layer 3 device operates at the network layer and deals only in packets. An example of a layer 3 device is an IP router. Layer 2.5 is a special mode of operation where some of the advantages of layer 2 are leveraged at layer 3. The different layers are described in the following sections.

Layer 2 and VLANs

Figure 1-4 illustrates the core of a fictitious enterprise network operated exclusively using ATM/MPLS multiservice switches. This is a layer 2 network that is logically divided into VLANs (well described in [Tanenbaum2003]). VLANs, as we noted earlier, are broadcast domains that allow communication between member devices as if they were all on the same physical LAN segment.

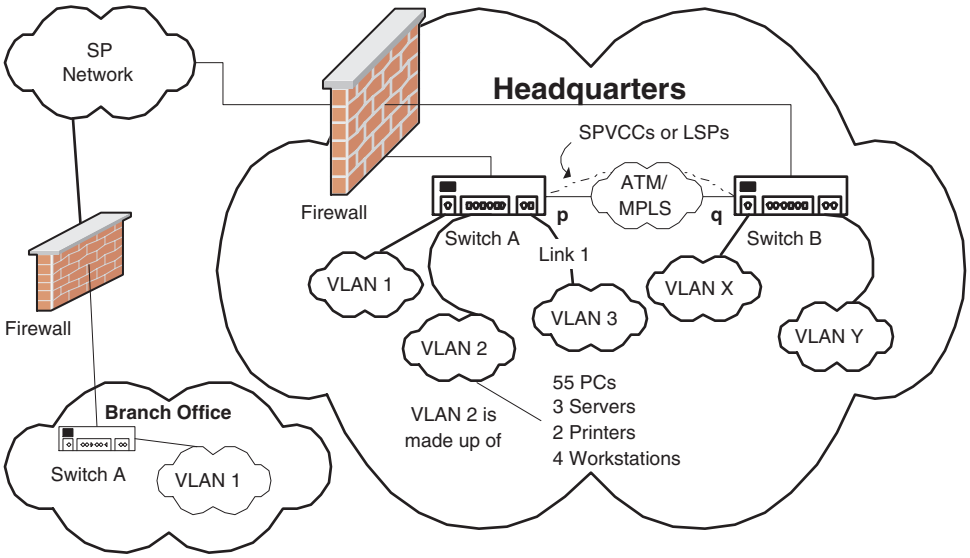

Figure 1–4 VLANs in an enterprise network.

The switches in Figure 1-4 serve to partition the VLANs by forwarding only appropriately addressed frames. In an effort to improve convergence time, some switches support, on a per-VLAN basis, the spanning tree algorithm (the means by which loops are avoided [Tanenbaum2003]). Spanning Tree Protocol is usually implemented across all LANs, not just VLANs. If it is implemented on a per-VLAN basis, it improves convergence.

The constituents of any of the VLANs in Figure 1-4 can include a number of machines; for example, VLAN 2 consists of 55 PCs, three servers, two printers, and four workstations. Layer 2 broadcasts originating inside any of the VLANs do not cross the boundary of that VLAN. One possible configuration is to allocate a specific VLAN for each layer 3 protocol—for example, IPX in VLAN 1 and IP in the other VLANs. Since VLAN 1 has nodes that understand only IPX, there is no reason for pushing IP traffic into it. Likewise, the nodes in the other VLANs might not understand IPX, so there is no reason for pushing IPX traffic into them. Only layer 3 traffic that needs to exit a VLAN crosses the boundary (via routing) of its container VLAN.

The merit of a VLAN arrangement is that traffic between the constituent devices does not pass needlessly into the other VLANs. Also, if one of the VLANs fails (or if a node inside that VLAN becomes faulty), then the other VLANs can continue to operate. This allows for a more scalable and flexible network design than using IP routers in conjunction with Ethernet segments.

Typically, the hosts in each of the VLANs support layer 3 routing capabilities (e.g., IP, IPX). This is required for communication outside the VLAN boundary. Each such host supports layer 3 routing tables with at least one entry pointing to an external router. The latter may be implemented on the local switch (A or B in Figure 1-4) and serves to direct outgoing and incoming IP traffic across the VLAN boundary. To illustrate this, Table 1-1 depicts an excerpt from a routing table from one of the 55 PCs in VLAN 2. The data in Table 1-1 is obtained by using the `netstat -r` command from within a DOS console.

Table 1–1 IP Routing Table for a Host PC in VLAN 2

Network Destination	Netmask	Gateway	Interface	Metric
127.0.0.0	255.0.0.0	127.0.0.1	127.0.0.1	1
Default Gateway	142.159.65.17	N/A	N/A	N/A

Table 1-1 illustrates two routing table entries: one for the loopback address and the other for the default gateway. Any packets addressed to the loopback address are sent back to the sender. So, if you ping 127.0.0.1, your host machine (i.e., the sender) will reply. The second entry in Table 1-1 is for the default gateway. This is the IP address of last resort (Internet core routers do not have default gateway entries), that is, the address to which packets are sent for which no other destination can be found. In Figure 1-4 this address (142.159.65.17) would be located on Switch A. It is by this means that hosts in VLAN 2 can exchange messages with entities outside their VLAN boundary. Appendix B includes examples of using some of the Windows NT/2000 networking utilities.

Another important point about VLANs is that the backbone network (between switches A and B) may be implemented using ATM. If this is the case, then the backbone may implement ATM LAN Emulation (LANE). This serves to make the ATM network behave like a LAN. The backbone can also run MPLS.

Greater flexibility again is afforded by the use of IEEE 802.1Q VLANs. In this technology, the 802.1 Ethernet frame headers have a special 12-bit tag for storing a VLAN ID number. This allows for traffic to flow between different VLANs. It is also possible to use another tag in the 802.1 header for storing priority values; this is the IEEE 802.1p tag—a 3-bit field. This allows different types of traffic to be marked (with a specific priority number) for special treatment.

Traffic that must pass across the ATM/MPLS backbone is destined for another VLAN (e.g., VLAN X in Figure 1-4). This traffic can be transported using either ATM or MPLS. ATM cells are presented at interface p of ATM Switch A. An ATM Switched (Soft or Smart) Permanent Virtual Channel Connection (SPVCC) has been created between switches A and B. This virtual circuit traverses the ATM/MPLS cloud between switches A and B. An SPVCC is a signaled virtual circuit, which forms a connection between interfaces on a number of switches. An SPVCC is conceptually similar to a time-division multiplexing (TDM) phone call: An end-to-end path is found, bandwidth is reserved, and the circuit can then be used. The SPVCC in Figure 1-4 starts at interface p on Switch A, travels across the intermediate link, and terminates at interface q on Switch B. This bidirectional virtual circuit transports traffic across the backbone between switches A and B. An important point about circuits that traverse the backbone is that some switches allow the mapping of IEEE 802.1p values to specific circuits. This allows for quite fine-grained quality of service across the backbone.

The SPVCC is a layer 2 connection because the constituent switches have only layer 2 knowledge of the traffic presented on their ingress interfaces. The layer 2 addressing scheme uses a label made up of two components: the Virtual Path Identifier (VPI) and Virtual Channel Identifier (VCI) pair. Each switch does a fast lookup of the label and pushes the traffic to the associated egress interface. The switches have no idea about the underlying structure or content of the traffic, which can be anything from telephony to IP packets. As indicated in Figure 1-4, the virtual circuit can also be realized using MPLS label switched paths (LSPs). Such LSPs carry layer 2 traffic encapsulated using MPLS labels (more on this later).

The layer 2 technology that we describe has the following general characteristics:

- Paths through the network can be reserved either manually (by using ATM PVCs or MPLS LSPs) or using signaling (such as ATM PNNI,[3] MPLS LDP/RSVP-TE).
- Paths can be assigned different classes of service, a crucial component for SLAs.
- Layer 2 forwarding is fast because addresses can be looked up with hardware assistance. This is no longer an advantage of layer 2 devices because line-rate forwarding is now also possible with layer 3 devices (i.e., routers).
- ATM layer 2 forwarding allows for traffic policing where contract noncompliant cells can be tagged or dropped. It is also possible to shape traffic so that its arrival rate is controlled. As we'll see when we look at DiffServ, policing and shaping are also available at layer 3.

The SPVCC/LSPs in Figure 1-4 represent our first example of virtual circuits. The different categories of traffic (TDM, IP, etc.) presented at interface p can be transported across appropriate virtual circuits. These circuits can be provisioned with different quality of service (more on this later) characteristics to ensure that the traffic receives specific forwarding treatment. So, far, we've only hinted at some of the elements of MPLS but it will be seen that many of the advantages of layer 2 technologies can be obtained at layer via MPLS.

Layer 3

Figure 1-5 illustrates an IP network with an intermediate WAN that crosses an SP network. A client PC in Dallas has some IP data to send to a server in Boston, and the traffic is carried to the destination via the SP network. Each router along the path performs a lookup of the destination IP address (142.159.65.17) and forwards the packet to an appropriate output interface.

3. Strictly speaking, PNNI (Private Network-to-Network Interface) is both a routing and a signaling protocol.

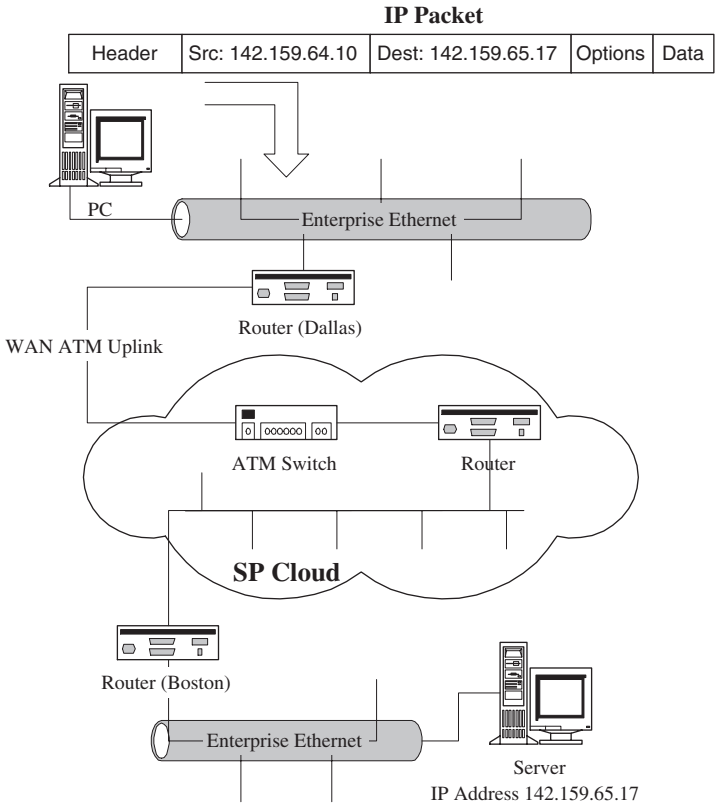

Figure 1–5 An IP network.

One of the other major differences between layer 2 and IP is that the latter cannot reserve either resources (such as bandwidth) or paths ahead of time. Even with static routes installed, a full IP address lookup is required at each router, and the direction that the packet takes can change at each hop (for example, if a static route goes down). So, IP packets from a given source can travel over different routes at different times, and ordering is not guaranteed. The TCP protocol gets over some of these problems, but TCP can't reserve bandwidth and full address lookups are still required at each hop.

Layer 2.5 (or Sub-IP)

A further possibility exists for transporting layer 3 traffic: MPLS. MPLS operates at what is often called layer 2.5, that is, not quite layer 3 but also higher

than layer 2. MPLS operates by adding a fixed-length (4-byte shim header) label to the payload, which includes an unstructured 20-bit label. This label is then used in forwarding the encapsulated packet. The label is structured for compatibility with ATM VPI/VCI addressing and allows for ATM[4] switches to be upgraded to MPLS. MPLS can also be deployed on routers and brings numerous benefits to IP networks:

- Paths can be reserved before traffic arrives at the network for transport. These can be created either manually or via a signaling protocol.
- Different classes of service can be applied to the reserved paths; for example, VoIP traffic would need a higher class of service than email traffic. This facilitates differentiated services that can be applied to fulfill customer SLAs.
- Traditional IP routing protocols, such as OSPF, IS-IS, and BGP4, can be used. This reduces the cost of developing and deploying MPLS because it leverages proven routing protocols (when they are appropriately extended).
- Traffic engineering becomes possible, allowing every packet to be individually and dynamically processed, resulting in different routes being taken. This helps avoid congested routes.

One disadvantage of MPLS is that all nodes in the path must run the MPLS protocols—an additional burden on network operators. Traffic engineering is often called the MPLS killer app because it permits connection-oriented operation of IP networks. Incoming IP traffic can be redirected to a higher or lower bandwidth path.

Apart from traffic engineering, an emerging function of MPLS is the generic transport of legacy layer 2 services, such as ATM, FR, TDM, and Ethernet. This is an effort to provide a standards-based migration path for network operators who do not want to fully deploy MPLS throughout their networks. In other words, the legacy services continue to be deployed, but they are transported across a fully or partially deployed MPLS core.

4. Where ATM switches are upgraded to function as MPLS nodes, there is no shim header. Instead, the ATM VPI/VCI fields are used for conveying the label.

Ports and Interfaces

The terms *port* and *interface* are often used interchangeably. In this book they have a specific meaning. Ports are taken to be underlying hardware entities, such as ATM or Ethernet ports. Interfaces exist at a higher level of abstraction and are configured on top of ports. This is similar to the way an Ethernet port on a PC is configured to run IP. Interfaces are sometimes referred to as logical ports. Examples of interfaces are:

- Routing, such as OSPF, IS-IS, BGP-4
- Signaling, such as RSVP-TE and LDP
- MPLS
- IP

In many cases, the user must manually configure interfaces. The key difference is that ports work out of the box, whereas interfaces generally do not. A lot of action takes place at interfaces—for example, quality of service (QoS) imposition in a DiffServ domain. QoS is a scheme by which traffic is marked prior to or at the entry point to a network. Each node traversed by the traffic then examines (and possibly updates) the marked values. The function of the traffic markings is a signal to the network nodes to try to provide the indicated level of service. Required service levels differ depending on the traffic type; for example, VoIP traffic has specific timing requirements that are more stringent than those for email. The point is that network node interfaces are an integral part of the provision of the QoS scheme. We will see more on this later.

Many SPs provide customer premises equipment (CPE) as part of an enterprise service. CPE is a term that describes some type of switch or router owned by the service provider but located on the customer premises. Examples of CPE devices are seen in Figure 1-5, such as "Router (Boston)". The CPE provides access to the SP network from within the enterprise network. Typically, the CPE provides access to services such as Metro Ethernet, VPN, ATM, FR, and TDM. All of these tend to take the form of one or more ports on a CPE device. Depending on the service purchased, CPE management may be executed either by the service provider, the enterprise, or some combination of the two.

In Chapter 6, "Network Management Software Components," Figure 6-8 illustrates some issues concerning the automatic configuration of IP interfaces. In Chapter 8, "Case Study: MPLS Network Management," Figure 8-3 illus-

trates a MIB table that provides details of MPLS interfaces on a given NE. One use for the MPLS interface table is selecting MPLS-specific interfaces in an NMS. Selected interfaces can then be used for inclusion in LSPs.

Why Use Network Management?

Devices deployed in networks are increasingly intelligent, so it is interesting to ponder the need for network management. If devices are so smart, then why bother with network management? Can't NEs just self-heal in the event of problems like interfaces going down? Many enterprise networks do not employ NMS—this may be just a matter of policy or even history. There are a number of reasons why network management is a crucial enterprise and SP component:

- NEs don't tend to have an overview of an entire network; management systems do, and this helps in creating objects like connections such as the ones shown in Figure 1-4. The NMS overview is particularly useful for aggregate objects, as we'll see later in this chapter.
- An NMS maintains useful records and audit trails of past configuration actions.
- If NEs do not support SNMP, then an NMS can facilitate a superior CLI because security can be imposed, actions are recorded, and scripts can be managed (stored, updated, etc.).
- NMS can facilitate useful networkwide services like traffic engineering, QoS, planning, modeling, and backup/restore (of firmware or configuration data).
- NMS enables fast access to faults. Some network faults can be meaningfully processed only by an NMS. For example, if a network contains many ATM permanent virtual circuits (PVCs) and an unprotected link fails, then the switches cannot automatically recover, because PVCs do not use signaling. In this situation, management intervention is required to restore the broken link and then the connection. As enterprise networks become increasingly mission-critical with IT offering stringent service contracts, downtime is a luxury few enterprises can afford. So, if a connection fails and has no backup, then the NMS needs to detect it as soon as possible and assist in recovery.

- NMS assists in rebalancing networks after new hardware is added. As networks expand and new switches and routers are added, it is often necessary to bring the new devices into service quickly. Often, such reconfigurations are done during periods of low traffic.[5] A management system can assist this process by allowing automated bulk operations, such as simultaneously creating or moving hundreds (or even thousands) of virtual circuits such as ATM PVCs or MPLS LSPs.
- Management systems can provide networkwide object support for service profiles. Subscriber management on large mobile phone networks is a good example of this. The management system can be used to create thousands of subscriber records and write them into a service database. Individual subscribers can then be updated as they connect to the network.

A good quality NMS broadens the operator's view of the network. This can help to leverage the increasing intelligence of modern NEs.

What Is Network Management?

The preceding sections have described some typical large networks now in common use. They have hinted at issues concerning network management, which is now described more fully. Network management provides the means to keep networks up and running in as orderly a fashion as possible. It includes planning, modeling, and general operation. It also provides command and control facilities. Broadly speaking, the functional areas required for effective network management are:

- **F**ault: All devices at some point can become faulty, and virtual connections, links, and interfaces can go up or down. These can all cause the generation of network fault data. Events are similar to faults except that they do not necessarily signify anything is wrong with the network. They exist to inform the management system of important occurrences, such as an LSP becoming operational (i.e., ready to forward incoming traffic).

5. Such reconfigurations can result in both signaling and routing storms as the network attempts to converge. MPLS networks tend to carry routing and signaling in-band, whereas optical networks carry routing out-of-band. This makes MPLS networks less resilient in the face of massive reconfigurations; hence the need for management systems.

- **C**onfiguration: All devices tend to require some type of configuration or tuning. Configuration settings may be both written to and read from devices. In Chapter 8 we illustrate the configuration of MPLS MIB objects.
- **A**ccounting: Billing for service is an important component of enterprise network management (e.g., for departmental service billing). This function can be used for charging back the use of resources, such as dial-up facilities, to individual departments as well as for verifying the bills submitted by a service provider.
- **P**erformance: As user populations and bandwidth needs grow, it is essential to be able to measure performance, particularly for SLA fulfillment. Performance checks can assist in predicting the onset of congestion.
- **S**ecurity: Attacks against networks can include unauthorized access, data modification or theft, and so on. Security is needed to ensure that both data and the underlying network are protected.

The above points describe what are known as the OSI functional areas of network management, **FCAPS**, described at length in [Stallings1999]. A good management system should fulfill all the FCAPS areas (many products provide only fault and performance management, leaving the other areas to proprietary means or ancillary products). An important point is that the FCAPS areas are inter-dependent. Fault management has to know about the network configuration in order to provide meaningful reports. The same is true of performance monitoring, particularly for complying with SLAs. Likewise, billing (or accounting) has to have some knowledge of the underlying configuration. Providing all this management capability is a big challenge, especially for large distributed networks containing lots of legacy equipment.

Organizations implement their FCAPS functions in interesting ways: Some do all the management inhouse while others outsource it to third parties. Large enterprises can operate a number of different types of networks, such as broadband, IP, and telephony. Each network has to be managed, and in many cases each has its own management system, making overall management complex, error-prone, and potentially very time-consuming. In cases where network management is felt to be too difficult or no longer a core activity, an organization can turn to outsourcing. Outsourcing to a third party can help to alleviate some of the duplication of multiple management systems by connecting the network to a

Network Operations Center (NOC). The owner of the NOC then provides billable services in any or all of the FCAPS areas. One merit of using a NOC is that network management costs can become more predictable. However, there are no hard and fast rules about this: An enterprise can also have its own NOC. The use of CPE is another example of outsourcing.

Who Produces Network Management Software?

Equipment vendors such as Cisco, Nortel, Hewlett-Packard, and Alcatel generally provide SNMP agents on their devices. Separately purchased, integrated management systems are also available from these and many other organizations. These management systems typically run on UNIX or Windows NT/2000 platforms and feature GUIs, object palettes for topology definition, and fairly extensive FCAPS facilities.

The Management System Pyramid

The owners of large networks tend to functionally separate their software tools, and FCAPS provides a conceptual framework for this. Management systems for large service providers tend to follow a distinct layered structure, illustrated in Figure 1-6 as a pyramid.

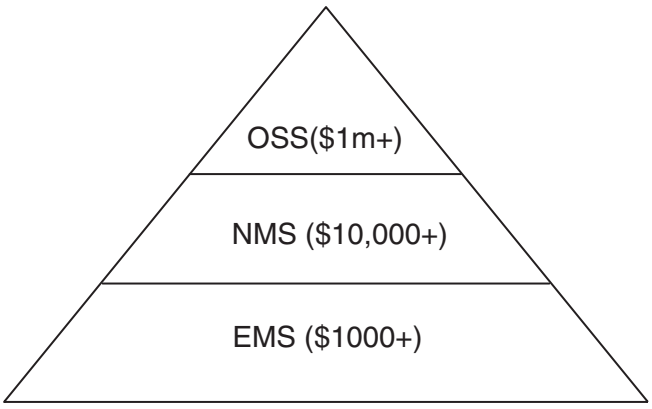

Figure 1–6 The management system value pyramid.

There is good reason—both technical and financial—for this functional separation. At the top of the pyramid is the Operational Support System (OSS), which is used for SP business support (feeding into other corporate systems) and overall network support. In passing, we should note that there is some interest in enterprise operational support systems for converged enterprise networks (e.g., AceComm is one vendor in this area). The OSS layer is expensive to develop and deploy. Below the OSS is the NMS, which tends to be used for network-facing operations such as creating, monitoring, and deleting virtual connections—ATM PVCs and MPLS LSPs, for example. The NMS is generally focused on multiple devices at any given time; in other words, it has a network-wide usage. This is reflected in the value of NMS, which can be priced per network node. Very often, NMS are sold in conjunction with NEs in order to help in quickly bringing up and subsequently maintaining the network. Below the NMS is the EMS, which—at the bottom of the NMS food chain—is generally focused on a single device at any given time.

Sometimes the EMS is a separate application hosted on an external platform in just the same way as the NMS. This can be the case where devices are:

- Primitive (not enough onboard capacity for adding management)
- Old or near the end of their lifecycle

Vendors of such devices can add an external EMS to extend the device utility. In other cases, the devices themselves host the EMS, allowing device-centered management facilities such as:

- Software upload/download
- Configuration database backup/restore
- Alarm processing and storage

The guiding principle is always the same for management systems deployment: It should make the object of attention (NE, network, virtual connections, etc.) easier to use.

As we mentioned, the NMS has a networkwide perspective and provides facilities to:

- Create, delete, and modify multiple NE objects, such as VLANs and VPNs.
- Create virtual connections between network devices.

- Create, monitor, and delete various soft objects on network devices such as connections, profiles, and paths.[6]
- Correlate alarms with connections when a failure occurs.
- Apply actions, such as software (or configuration data) uploads and downloads, on a networkwide basis.

It is important to note that the gap between the EMS and NMS is not always clear cut. Sometimes, an NMS is called upon to process raw device alarms or even to distribute new firmware across a range of devices. The dividing line between EMS and NMS is that most NMS operations tend to simultaneously involve more than one NE. EMS operations tend to center on a single NE. In some cases, an EMS can be developed that runs on a system (such as a PC or UNIX system) external to the NE. The EMS then handles the NE interactions. EMS development for devices without SNMP agents can be quite cumbersome, often involving some type of automated interaction with an NE menu system or CLI. In this, the NE presents its menu options and the EMS emulates a human user and selects the required option, moving to the next menu level.

Even with an EMS that interacts with an NE CLI, it is still possible for an NMS to then interact with the EMS. So, even though the EMS is not onboard the NE, this does not concern the NMS. However, the whole scheme tends to be proprietary in nature. In effect, the approach taken is message-based: The EMS directly exchanges messages with the NEs. This approach is not without difficulties: Different versions of firmware may support slight variations in CLI message format or content, making it difficult to formulate a generic approach to such EMS-NE exchanges. Having SNMP on the devices makes EMS development much easier because then the operations are based on standard SNMP message exchanges.

As we mentioned earlier, many network operators use a script-based approach to setting up and monitoring devices. This consists of writing large and complex vendor-specific scripts that are then sent to the NEs for batch execution. Clearly, this requires scripts that adhere to a given manufacturer's CLI. This is fine if all network devices support the same CLI (i.e., same vendor and CLI ver-

6. A path in this context is a specific set of nodes and interfaces (on the nodes in question) between two points in the network. Both ATM and MPLS support path objects for use in creating virtual connections. Typically, preconfigured paths are provisioned in the network via signaling. For this reason, paths have an independent significance in an NMS.

sion), but this cannot be guaranteed. SNMP, as part of an NMS, provides a better (standard) mechanism for such operations. In Chapter 6 we will see an example of using an NMS to configure IP interfaces. An NMS offers facilities such as security, script management, and audit trails.

Up from the NMS is the OSS layer [Tele2001]. As mentioned, these are very large bodies of software typically deployed in big SP (and some enterprise) networks. OSS provide a variety of business- and network-support functions such as:

- Subscriber setup and management
- Switching new services on and off
- Workflow ordering for device configuration, connection creation, etc.
- Trouble ticketing (forwarding notifications about faults)
- Asset management

Many OSS can use the services of the underlying NMS to do some or all of the following:

- Retrieve faults and network configuration
- Retrieve performance and billing data
- Execute provisioning

In this way, an OSS uses the NMS services in the manner of an API.[7] The TeleManagement Forum (TMF) [TeleMgmtForum] has made great progress in modeling SP operations processes and defining vendor-independent interfaces between OSS and NMS. An example of such a model is the Telecommunications Operations Map (TOM). The TMF has defined Interface Definition Language (IDL) specifications for this purpose. The OSS and NMS both use the IDL for communication.

One consequence of the connection between OSS and NMS is that it is often hard to decide where and how specific management software should be written. For example, an operator might require the ability to create an ATM SPVCC (or an MPLS LSP) connection between two nodes on its network. Should the NMS vendor provide this capability through its NMS GUI or as an IDL API

7. An API in this context means that the NMS exposes a software interface (e.g., a CORBA interface) to the OSS. This interface might provide services like `retrieveAllAlarms()` for a given NE. When the OSS uses the interface, the underlying NMS executes the request, retrieves the data, and presents it to the OSS. The use of a standard OSS-NMS API frees the OSS from the need to understand any details of the NMS structures.

function? Probably both, but it's possible that the GUI version might never be used in an SP environment, so should the vendor provide two solutions when only one is needed? These are heady design questions that have a profound effect on the way in which NMS are both built and used.

Other Management Technologies

SNMP is not the only management technology. Other proprietary systems approaches include:

- Microsoft Systems Management Server (SMS)
- Telnet-based menu systems
- Serial link-based menu systems
- Desktop Management Interface (DMI)

Microsoft SMS allows system administrators very flexible control of networks of Windows machines. Software applications deployed on host machines can be determined by remotely viewing the local Windows registry (a type of configuration database) on each machine. This can be very useful for verifying on large sites that software licenses have not been exceeded—too many users installing a given package. SMS also allows software to be distributed to destination machines. This is very useful for updating applications like virus detectors (indispensable nowadays). Remote operations like this greatly facilitate IT support call centers. A shortcoming of SMS is that it works only on Windows machines. SNMP differs from SMS in one crucial way: SNMP is technology-independent. The only local facilities needed for SNMP management are distributed agents with encapsulated MIBs. Management applications then interact with the agents to monitor and control operation.

Telnet refers to a menu-based EMS/CLI style of management. This approach requires the management user to connect to the IP address of a given device using telnet. The device then provides a text menu-based application with which the user interacts. This is useful and is a widely adopted approach for device management. It is generally possible to use telnet to configure devices such as laser printers, routers, switches, and terminal servers. The problem with it is that menu-based management systems are proprietary by their nature and don't easily lend themselves to centralized, standards-based management (as does SNMP).

Serial link-based menu systems are very similar to NEs that support telnet. Just the access technology is different. Normally, a serial link-based system includes simple text menus (accessed via a serial interface) that are used for initial configuration. Typical devices for these facilities include small terminal servers. Often, these devices do not have an IP address, and the user configures one via the menu system. Connecting the device to an appropriately configured PC serial port facilitates this. Again, by its nature this is proprietary.

DMI was developed by the Desktop Management Task Force and is completely independent of SNMP. Its purpose is the management of desktop environments, and it includes components similar to those of SNMP, such as DMI clients (similar to SNMP managers), DMI service providers (similar to SNMP agents), the DMI management information format (similar to the MIB), and DMI events (similar to SNMP notifications).

Network Convergence and Aggregate Objects

The provision of services such as Metro Ethernet and layer 3 VPNs is prsenting an interesting network management challenge. Not only are new services being deployed, SP (and to some extent, enterprise) network cores are migrating to layer 3. Managing these converged networks in a scalable, end-to-end fashion is a necessity, especially when competitive SLAs are sold to end users. The service sold to the user may consist of Ethernet (with different priorities supported for specific traffic types), cross-connected into an MPLS core. Modeling this for network management support requires the use of what we call aggregate objects. Aggregate objects are comprised of a number of related managed objects. Examples are VLANs, VPNs, and cross-connect technologies (e.g., Ethernet over MPLS). As the range of technologies and services deployed on networks continues to grow, aggregate objects are becoming increasingly important. We'll see this in Chapter 8 when we discuss LSP creation.

Figure 1-4 introduced us to VLANs. From a network management perspective, VLANs are aggregate objects made up of:

- Switches
- Ports, MAC addresses, IEEE 802.1Q VLAN IDs
- Links between separate VLANs

Generally, there are two ways for an NMS to build up a picture of a VLAN:

- Manual creation by an IT manager
- Automatic discovery from the network

Manual creation requires a combination of human input and network-side provisioning. The user selects the switches required for the VLAN and adds the VLAN members. Provisioning software in the NMS then updates the appropriate MIBs. This is the textbook way of operating networks, but in reality networks may tend to change quite often.

NMS Discovery

In many cases, changes are made to individual switches via the EMS (usually via the onboard CLI) and unless the user manually updates the NMS, then the EMS-NMS pictures may differ. This is where an NMS feature called *network discovery* is important. Network discovery is the process by which an NMS uses SNMP (and also ICMP) to read, process, and store the contents of designated MIB tables. In this way, the NMS picks up any changes made via the EMS. This process of ongoing discovery and update is an important aspect of managing large networks. Network discovery also picks up the details of both simple objects and complex aggregate objects.

Not all NMS products provide automatic network discovery, because it introduces traffic into the managed network. Also, the workflows of the operator may (manually) provide the same service with no need for an automated solution. We will tend to assume that an automated network discovery function exists.

So (using either manual or automatic network discovery), we now have our picture of the network and its higher level constructs (including aggregate objects such as VLANs and VPNs). Having a clear picture of the network objects leaves the operator free to effectively manage the network.

What kinds of things can the operator expect to happen to the network? Links and interfaces can go down; for example, if Link 1 in Figure 1-4 goes down, then VLAN 3 will become isolated from the enterprise network. The NMS (not shown in Figure 1-4) should receive a notification from the network that the link has gone down. The NMS then has to cross-reference the notification with the associated aggregate object (in this case VLAN 3) and infer that VLAN 3 is no longer connected to the network. The NMS should indicate the problem (usually in a visual fashion, such as via a GUI color change) to the operator and possibly even suggest a fix. Similarly, if an NE in one VLAN

becomes faulty—for example, if a NIC starts to continually broadcast frames—then the NMS should figure this out (by looking at interface congestion indicators) and reflect it back to the user. The user can then resolve the problem.

The Goal of an NMS

This mechanism of important events occurring in the network and the NMS (and operator) racing to figure out what happened is crucial to understanding NMS technology. **The difference between the NMS picture of the network and the real situation in the network must be kept as small as possible. The degree of success attributed to an NMS is directly related to this key difference.** This is an important NMS concept that we will refer to frequently.

Notifications

We use the term *notification* to mean any one of three different things:

- Events
- Faults
- Alarms

An event is an indication from the network of some item of interest to the NMS, for example, a user logging into an NE CLI. A fault is an indication of a service-affecting network problem, such as a link failure. The NMS must respond as quickly as possible to a fault, even suggesting some remedial action(s) to the operator. An alarm is an indication that a potentially service-affecting problem is about to occur, perhaps an interface congestion-counter threshold that has been exceeded. Clearly, in most cases, faults should be processed by the NMS ahead of events and alarms.

SNMP: The De Facto Network Management Standard

Two efforts were made in the 1980s to standardize the area of network management: OSI and SNMP. The intention was that the OSI approach would eventually replace SNMP, but this never happened. In the end, the OSI approach was found to be too complex for widespread adoption and was overtaken by its simpler counterpart. Some OSI inventions, such as ASN.1, did find their way into SNMP. SNMP is a much lighter variant that was globally adopted and has now become the de facto standard for network management. SNMP-based

management system components are distributed throughout a network in the form of agents and managers. The appeal of the lightweight SNMP entities is that they consume minimal resources: This can be an aid to scalability (discussed in Chapter 3). The latter is an important management requirement even on modern networks with highly powerful hosts. The principal components of SNMP are:

- Agents
- Managers
- MIBs
- A communications protocol

The SNMPv3 standard replaces the terms agent and manager with *entity*. While entity is the correct term for SNMPv3, we need to distinguish between the manager (server) side and device (agent) side. So, for clarity, we will continue to use the terms agent and manager. Unless otherwise stated, these refer to SNMPv3 entities. Also, any reference to SNMP from here on should be interpreted as SNMPv3 unless otherwise stated. We now describe these principal components.

The SNMP Agent

SNMP agents are the entities that reside on managed devices. They listen on UDP port 161 for incoming SNMP messages; they use UDP port 162 for sending notification messages. Agents are the workhorses of management and provide the following functionality:

- Implementing and maintaining MIB objects
- Responding to management operations such as requests
- Generating notifications, both traps (unacknowledged) and informs (acknowledged)
- Implementing security—SNMPv1 and SNMPv2c support community-based security with clear-text passwords; stronger security (authentication and encryption) is available with SNMPv3
- Setting the access policy for external managers

SNMPv3 also provides an access control framework, which consists of:

- MIB view—the set of managed objects in an agent MIB accessible to an SNMP manager. This is the manager's client view with respect to the agent.
- Access mode to managed objects—either READ-ONLY or READ-WRITE. A READ-ONLY access mode means that no agent MIB objects can be written by a manager. MIB views are associated with specific access modes.

SNMP agents can be hosted on almost any computing device, including:

- Windows NT/2000 machines
- UNIX hosts
- Novell NetWare workstations and servers
- Many network devices, including hubs, routers, switches, terminal servers, PABXs, and so on

The agent listens on UDP port 161 for the following SNMP message types:

- `Get` requests the values of the specified object instances.
- `Get-next` requests the values of the lexical successors of the specified object instances.
- `Get-bulk` requests the values of portions of a table.
- `Set` modifies a specified set of object instance values.

The above messages either retrieve (`get`) or modify (`set`) NE data as defined in the MIB. The agent uses UDP port 162 for sending notification messages to a preconfigured IP address. Agents reside in the managed network and communicate with managers (described in the next section).

The SNMP Manager

SNMP managers are the entities that interact with agents. They provide the following functionality:

- Getting and setting the values of MIB object instances on agents
- Receiving notifications from agents
- Exchanging messages with other managers

It is unusual nowadays to have to write either SNMP agent or manager programs. Many system software vendors include them as standard software com-

ponents. For example, all of the following products include an SNMP agent and manager:

- The pSOS [PSOS] real-time, embedded operating system
- The VxWorks [VXWORKS] real-time, embedded operating system
- The Java JDMK toolkit

In the cases of pSOS and VxWorks, the SNMP agent can be ported to an embedded system, such as a switch or router. This device then constitutes an NE and can be managed by an NMS. The SNMP agent on the NE can be considered part of another component called the EMS (which we met earlier). This is software dedicated to managing the NE. Various mechanisms for accessing the EMS are allowed, including:

- Serial
- Telnet
- SNMP

The NMS generally interacts with the EMS on its managed NEs using one of the above access methods. When an NE is first deployed in its factory-default state, it is often necessary to configure it via a serial interface. The other services and protocols available on the NE can then be enabled so that it can subsequently be accessed over a network. The major focus of this book is the NMS.

In Chapter 7, "Rudimentary NMS Software Components," we build basic Visual C++ and JDMK Java SNMP manager programs. Normally, an SNMP manager is a low-level software entity embedded in a larger body of software called the management application. The combination of the user, management application, SNMP manager, and multiple distributed SNMP agents constitutes the management system. Facilities offered by a management system are:

- FCAPS
- A centralized database
- Reporting
- Support for many simultaneous client users
- Topology discovery (not all NMS provide this)
- A full-featured, multilevel GUI representing the managed network

Both agents and managers support MIBs. Agents implement their MIB objects and (where appropriate) map them to real NE data. An example of such

a mapping is between the `ipInReceives` object (from the IP MIB table) and the underlying NE IP protocol implementation. Strictly speaking, this mapping holds true for a host. For a router, the `ipInReceives` object is maintained by the interface statistics. However, in either case, the `ipInReceives` object maps to a piece of data maintained by a designated section of the NE.

The MIB

The importance of MIBs cannot be overstated. This is a recurring theme throughout this book. MIBs are a crucial component—perhaps *the* crucial component—of an NMS because they contain the data definitions for the managed objects. In Chapter 8 we use the MPLS MIBs to create LSPs. A MIB is simply a managed-object data description. The MIB defines the syntax (type and structure) and semantics of the managed objects. SNMP managers and agents exchange managed object instances using the SNMP protocol.

Managed objects may be defined using what are called *textual conventions*. These are essentially refinements of basic types (that are very loosely analogous to programming language data types or even Java/C++ classes), and some of those included in SMIv2 (Structure of Management Information) are:

- `MacAddress` is an IEEE 802 MAC address.
- `TruthValue` is a boolean value representing true (1) or false (2).
- `TestAndIncr` prevents two managers from simultaneously modifying the same object. Setting an object of type `TestAndIncr` to a value other than its current value fails. We will see a similar mechanism used in the MPLS tables.
- `RowStatus` is a standard way for adding and removing entries from a table (we will see this object used many times in the MPLS configuration examples).
- `StorageType` specifies how a row should be stored.

As discussed in the previous section, an example of a MIB object is the number of IP packets received by a host TCP/IP protocol stack from its interfaces. The MIB object called `ipInReceives`, in the IP group, fulfills this function (see Figure 1-8). Each IP packet received from a registered interface (including those received in error) results in the host agent incrementing the MIB object instance value for `ipInReceives`.

In addition to using textual conventions, MIB objects have additional attributes that are now described.

MIB Object Attributes

All SMIv2 MIB objects have a number of common attributes, including:

- `SYNTAX`: This is the object format—for example, `Unsigned32` (an integer), `TruthValue` (a Boolean true or false), and `SEQUENCE` (a container of other objects).
- `MAX-ACCESS`: This specifies the accessibility of the object—for example, `read-only` means that the object can only be read (but not written) by managers.
- `STATUS`: This is the state of support for the object in the MIB—for example, `current` means that the object is relevant and can or should be supported.
- `DESCRIPTION`: This is a text description of the object.
- `DEFVAL`: This is a default value that the agent can use when the object instance is first created.
- `OBJECT IDENTIFIER`: This is the unique name for a MIB object, described in the next section.

Managers use the object attributes in order to manipulate and understand MIB objects. Figure 1-7 illustrates an object called `mplsFTNAddrType` from the MPLS Forwarding Equivalency Class-To-Next Hop Label Forwarding Entry (FTN) MIB. This important MPLS MIB is described in more detail in Chapter 9, "Network Management Theory and Practice," to illustrate the way in which policy-based management is finding its way into the operation of MPLS NEs. For now, we examine the elements of a single object from this MIB in order to describe the above attributes.

Line Number	Object Attributes
1	**mplsFTNAddrType OBJECT-TYPE**
2	**SYNTAX InetAddressType**
3	**MAX-ACCESS read-create**
4	**STATUS current**
5	**DESCRIPTION**
	"The type of IP packet against which this entry will be matched. If this object has the value ipv4(1), then the objects in this entry of type InetAddressIpv6 MUST be ignored by management applications."
6	**DEFVAL { ipv4 }**
7	**::= { mplsFTNEntry 6 }**

Figure 1–7 MIB object example.

It's very important to be able to read MIBs, so we briefly describe the object in Figure 1-7. The first line is added for information only. It describes the columns in the figure. In the left-hand column is the line number, and the right-hand column shows the attributes (or characteristics) of the object. Real MIBs *do not* contain line numbers or headings like this. So, the real version of this MIB (in an agent or NMS) would not contain either the top line or the line numbers in the left-hand column.

On line 1, we see a MIB object called `mplsFTNAddrType`. This identifies the MIB object with a symbolic name. An NMS (or MIB browser) can do `gets` and `sets` using this name. We know this is a MIB object because of the keyword `OBJECT-TYPE`.

Line 2 indicates the syntax of the object (`mplsFTNAddrType`). It shares the syntax of another object called `InetAddressType` (defined in a MIB called `INET-ADDRESS-MIB`). This illustrates the way SNMP reuses legacy components to build new ones. The `SYNTAX InetAddressType` is imported from the latter MIB and represents an IP address string.

Line 3 indicates that the `MAX-ACCESS` (or operational permissions) allowed on object instances of type `mplsFTNAddrType` is read-create. This means that a manager can either read an existing object instance or create a new one.

Line 4 indicates that the `STATUS` of `mplsFTNAddrType` is current, meaning that this object should be supported.

Line 5 gives a `DESCRIPTION` of `mplsFTNAddrType` and provides a useful textual reason for the use of this object.

Line 6 provides an acceptable default value for instances of `mplsFTNAddrType`. This is indicated by the `DEFVAL` clause and in this case has the symbol value `ipv4` (this has the value `ipV4`, or `1`, as seen in the `DESCRIPTION`). Later we will see the importance of default values in the NMS.

Line 7 indicates the name used to access this object via SNMP—in this case it is column number 6 in the table row called `mplsFTNEntry` (defined earlier in this MIB).

Understanding the contents of Figure 1-7 takes us a long way on the road to understanding MIB objects. We now delve a little more deeply into the overall structure of MIBs.

OIDs and Lexicographic Ordering

All MIB objects have unique names called object identifiers (OIDs). An OID is a sequence of 32-bit unsigned integers that represents a node within a tree-based structure (with a single root). Only an instance of a MIB object can be retrieved from an agent. An instance of a MIB object is identified by an OID concatenated with the instance value. The instance value is a sequence of one or more 32-bit unsigned integers.

The order of the OIDs is an important aspect of SNMP. All objects can be traced from the root in a process called *walking the MIB*. During a walk, each branch of the MIB tree is traversed from left to right starting at the root. For example, the standard IP group or table has the OID 1.3.6.1.2.1.4, as illustrated in Figure 1-8. The IP group and some of its constituent objects are shown in this diagram.

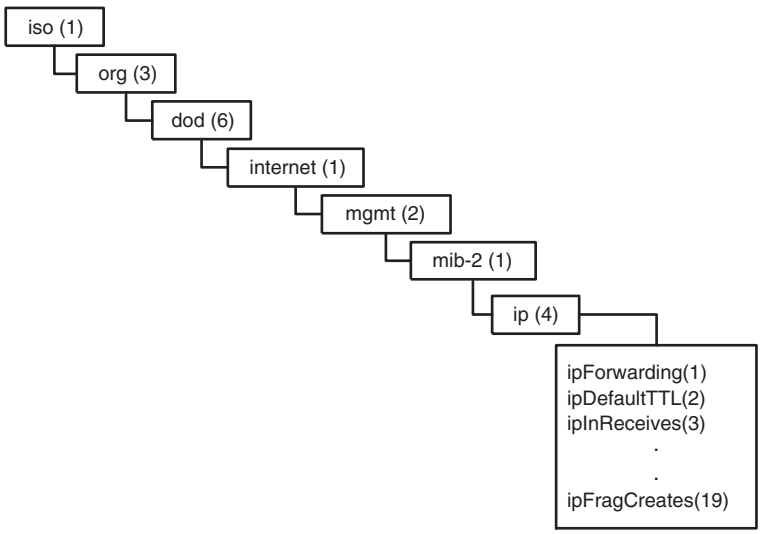

Figure 1–8 The MIB-II IP group.

MIBs are plain-text files. They are compiled into the agent source code and become part of the executable file. If a manager wants to access some agent MIB objects, then either the associated MIB module file is needed or a MIB walk can be attempted.

All MIB objects follow the model depicted in Figure 1-8. The IP object is a table that contains scalar (single-value) objects. It is also possible to have non-tabular scalar objects in the MIB, as we'll see in Chapter 2, Figure 2-5. Appendix C contains a list for part of the IP table retrieved from a real device.

Another important aspect of lexicographic ordering is that a manager can use it to "discover" an agent MIB. This is for that case in which the manager does not have a copy of the agent MIB and needs to determine what objects the agent supports. The discovery process consists of walking the MIB. It should be noted that this is not a very good way of retrieving agent data. It is far better to have the MIB details at the manager side because the structure and meaning of the NE data will then be apparent.

SNMP Protocol Data Units (PDU)

SNMP managers and agents communicate using a very simple messaging protocol. This is a straightforward fetch (`get`), store (`set`), and notification model

[ComerStevens]. Managers retrieve agent data using `get` operations, and they modify agent data using `set` operations. When agents want to communicate some important event, they do so by sending a notification message to a preconfigured IP address. If the agent wants to receive an acknowledgment from the manager, then it sends an inform message.

Table 1-2 illustrates the protocol messages provided by the different versions of SNMP.

Table 1–2 Protocol Data Units in the Different Versions of SNMP

SNMPv1	SNMPv2c	SNMPv3	Response PDU
GetRequest	GetRequest	GetRequest	GetResponse
GetNextRequest	GetNextRequest	GetNextRequest	GetResponse
SetRequest	SetRequest	SetRequest	GetResponse
Trap	Trap	Trap	None
	GetBulkRequest	GetBulkRequest	GetResponse
	InformRequest	InformRequest	GetResponse

In Chapter 2 we illustrate details of the SNMPv3 message types and their interactions between agents and managers.

Summary

Enterprise and SP networks are complex, interdependent entities. Enterprise network managers seek to improve business processes and workflow efficiencies by leveraging their technology. Service providers can help them achieve this by offering advanced managed or unmanaged (billable) services, such as VPNs. Both types of network have to be managed effectively using dedicated technology. We focus on SNMP-based network management, but it is important to note that this not the only approach. The trend in networking is towards what we refer to as *aggregate objects*. These can be seen in the many variants of interconnection technologies, such as VLANs. VLANs allow for LANs to be scaled upwards in a controlled fashion because the broadcast domain can be partitioned. This means that individual VLAN members (e.g., the software engi-

neering department) can communicate within the one broadcast domain without its traffic crossing into a neighboring VLAN (e.g., the sales and marketing VLAN). Traffic crosses VLAN boundaries only as required, and this occurs using layer 3 routing. The mix of technologies involved in VLAN-based environments gives rise to aggregate objects. These objects in turn present scalability challenges to network management.

A successful NMS is one that maintains an accurate and up-to-date picture of the managed network. This is a lot harder than it sounds, particularly with the complex mix of technology and traffic types (many now have stringent real-time requirements) found in networks.

NMS constituent technology tends to follow a client/server architecture with many products based on Java technology. A typical NMS product offers a range of applications that fulfills the basic FCAPS areas as well as others, such as reporting and multiclient control.

SNMP provides a distributed model that uses managed-object schema definitions (MIBs) on remote devices. Instances of managed objects can be retrieved from agents on remote NEs. This can be done by a manager in conjunction with a local copy of the agent MIB; that is, there are two copies of the MIB. MIB structures often must be reflected in the data model (more on this later, but for now the data model is the way the NMS looks at the information relating to the managed objects). For this reason, the NMS quality can suffer if the MIBs are badly written. The mapping of MIBs to real NEs is reasonably easy to understand, particularly after using a MIB browser application (some are freely available on the Web).

A security scheme protects the agent data as well as the data in transit from the agent. A notification mechanism allows agents to asynchronously send messages to a manager when important events (such as faults) occur.

SNMPv3 offers a small number of protocol messages designed to allow effective management of NEs.