Telecom

Network Management SNMP

Regis J. Bates

Excerpted from:
Broadband Telecommunications Handbooks

McGraw-Hill/Professional
A Division of The McGraw-Hill Companies

McGraw-Hill

A Division of The McGraw-Hill Companies

0-07-139183-5

The material in this eBook also appears in the print version of this title: 0-07-134648-1.

TERMS OF USE

DOI: 10.1036/0071346481.32

Contents

CHAPTER 32

Network Management
SNMP

Introduction

Simple network management protocol is the Internet standard for monitoring and managing devices connected to the Internet or your own Intranet. It defines a data set structure of information that each device may provide called a *Management Information Data Base* (MIB). It also makes provision for individual vendor-based, customized MIBs. A vendor then can provide the standard set of parameters and also extend that set with "custom," for example "vendor-specific," information. These data are collected, kept, and reported by an agent that runs on the managed device. We call this the managed function.

The Management Function resides on a workstation or host, and interacts with the managed function or agent. The management function must have a copy of the MIB profile loaded for each managed function, plus any vendor-specific profile.

In this manner, a standard management function can interact with the standard MIB, while a vendor-specific management function may be used to glean more specific information. These custom MIBs are also registered with the *Internet Advisory Board* (IAB).

Thus, you find standards-based systems, such as HP's OpenView, and vendor-specific systems, such as Fore's ForeView and Cisco's CiscoView, that are compatible and work with OpenView.

Network Management Goals

The ultimate goal of network management is to improve system uptime. The CEO of an organization may see it as another overhead expense. The CIO should see it as a tool for maintaining a near 100 percent uptime.

The higher the system availability, the less equipment is needed, and the lower the cost of running the network and the more efficient the organization will be that utilizes its resources.

Everyone in the corporate management structure needs a clear understanding of the goals of the network management, and the amount of time and effort required to reach those goals. Network management is a great tool for proactively managing the network and collecting alarms and alerts at threshold levels that are below the "Houston, we've got a problem" level. It has often been said that if you do have a failure, then the network management didn't work! Unfortunately, many network management systems are viewed as disaster recovery mechanisms rather than disaster avoidance and prevention mechanisms.

One must also consider the point of diminishing returns. Whereas you may be able to achieve 99.9999 percent uptime, the cost in additional redundant equipment and failover systems may not be justified for your enterprise. Here is a good point to insert a plug for disaster recovery plans. Every enterprise must have a disaster recovery plan. In the context of network management systems, what is the plan of action if multiple (unlikely) failures occur? The likely events should already be handled by the network design. (In this context, the building burning down or an earthquake rendering it uninhabitable is an unlikely event.) At one extreme, the plan may state "We close up shop and go out of business," or "We wait until the flood waters recede, clean up, buy new equipment, and start over". While these aren't recommended plans, they are plans and indicate that management has done some thinking about the problem. At the other extreme is a company, for example, a stock exchange or a hospital, which must keep going at all costs. In this case, alternate facilities are identified which contain computing equipment that can take over if their own equipment fails. A set of network connections is in place to provide connectivity to all of the customers/branch offices including a backup network management. The downtime is minimal. Namely, the downtime would be the time it takes to load the backup configuration on backup machines, plus the time it takes to switch over the network connections and — we're up!

History

In the days of centralized systems, the host or mainframe was about the only smart device in the system. The rest of the devices in the network were relatively dumb and could be reset when necessary by the host.

Figure 32-1 shows a typical centralized network configuration. Most of the remote devices in the network were connected to the host by dedicated lines. At the larger remote locations, a remote concentrator, or controller, is used to multiplex multiple terminals on the single data line. At smaller locations, a single terminal may exist. Dial-up lines may serve some low-traffic locations. But, in any case, there were not multiple paths in the network. It was strictly hub and spoke. Network management consisted of monitoring the status of all phone lines and modems. Dial backup for the dedicated lines was often used when high availability was required. This technique is still used today to back up our Frame Relay connection. Today, an ISDN line can provide 128 kbps dial backup.

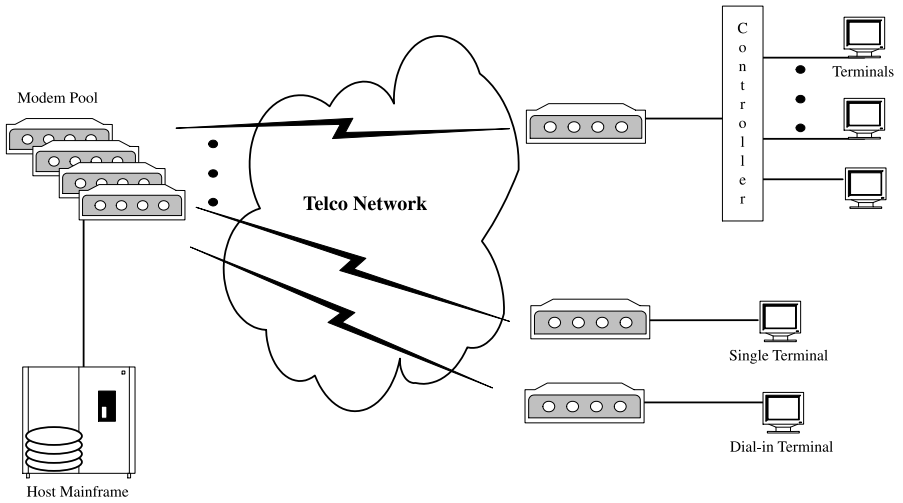

Figure 32-1
Typical centralized network configuration

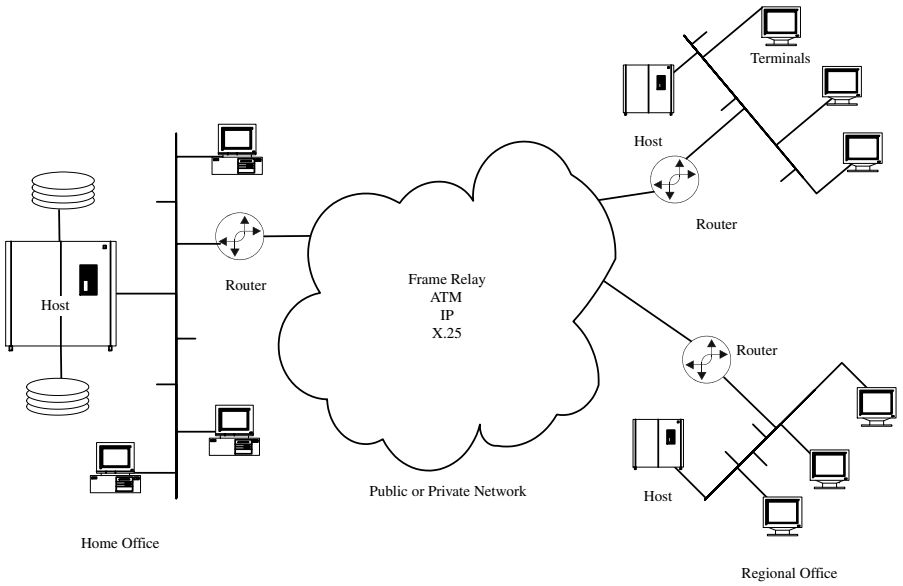

Figure 32-2
Typical distributed processing system

As we have migrated from the centralized model to the distributed computing model, every element in the network is now intelligent.

Figure 32-2 depicts a typical distributed processing system. The WAN technology isn't important per se in that it may be leased as a service from a network provider who will generally not allow our management system

visibility into his network components. What is important is that the WAN connection must be available in all of our locations, and that it provides an inexpensive high performance interconnection.

If the WAN is of our own construction, then we consider it to be part of our overall network. That is, we lease lines between locations and provide our own networking equipment, whether it is the Frame Relay switches or Routers.

In either case, the distributed system has many more elements to manage and most importantly that can be controlled managed. The failure of any one of these elements could cause a significant number of our users to be cut off.

In Figure 32-2, the "Router" may be taken to represent a router specifically, but in a general diagram like this it could be representative of other network elements such as a *Frame Relay Access Device* (FRAD), an ATM switch, a LAN bridge, or even a lowly hub.

Network Management Function Interaction

Philosophically, there is a manager function and a managed function. Their physical location in the network is not important. Managed elements may be physically collocated with the management function, or may be at the far edge of the network. The management function is typically located at the headquarters' location. However, it does not have to be that way. As a case in point, the major network providers, such as AT&T, Sprint, and World Com, have distributed *Network Operations Centers* (NOC).

The manager function is the user's interface to the network management system, and it is typically a GUI, which provides status, reports, and statistics gathered from the managed function.

Figure 32-3 depicts the logical relationship between the managed and manager functions. The managed element, or object, is the actual device, whether it is a hub, bridge, or router. The agent function is the program, or process, that runs on the managed system and provides the interface to the manager function. Associated with the managed element or device is a set of attributes. These attributes may include memory size and utilization, interface speed, traffic load, and so on, depending on the type of device.

As one might suspect, there is a protocol defined between the management function and the agent that defines the format and content of each of the commands, responses, and traps. RFC 1905 governs these protocols.

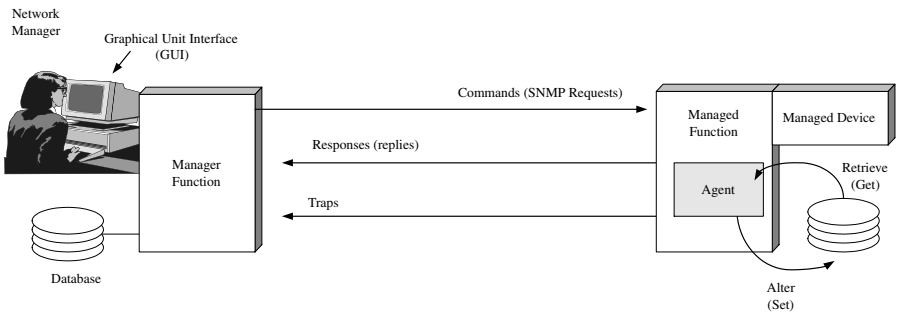

Figure 32-3
Logical relationship between the managed and manager functions

RFC 1907, in turn, governs the content of the SNMP messages. SNMPv2, also known as MIB-II or mib-2, is the dominant definition set today and has superseded the SNMPv1 mib-1. Version 2 greatly expands the detail available from the managed agent.

The agent is the interface to the manager function, and may accept commands to set parameters (such as timeouts, thresholds, and, in some cases, system operational parameters, such as group membership addresses or routing tables), as well as provide status upon request. The agent can also send unsolicited messages (called traps), indicating alarm conditions. (The name "trap" comes from the capability of the manager function to set a threshold (trap), and should this condition be met, the agent sends off an unsolicited message. All other messages sent by the managed function agent are a result of GetRequest or GetNextRequest queries.)

It should be clear that the data reported by a device like a hub is a lot less complex than that from something like a router. The hub needs to report Ethernet collisions on the LAN, the number of frames handled, the number of errored frames, and so on. The router, on the other hand, has routing tables, queue sizes, memory utilization, CPU utilization, and so on.

To keep this all straight, the management function must know about the content of each MIB on each device. In other words, it must have a complete database identical to the one on the managed device so that when it receives the information, it can place it in the appropriate location. When a new device is installed in the network, its MIB content and format must be loaded into the machine (typically, a host or a workstation), providing the management function. Likewise, the agent software must be installed in the managed device.

This emphasizes an important point. Setting up and running a network management system is more than a casual undertaking. It requires a commitment on the part of management to provide the resources and purchase the appropriate tools.

Database Structure

The MIB content is defined by the RFCs listed in Table 32-1.

Table 32-1 is a listing of the relevant standards that apply the network management, in general, and SNMP and RMON, in particular. The complete listing is available from: http://rfc.fh-koeln.de/rfc/html_gz/rfc.html.gz.

It is worth noting that the contents of the MIBs are defined by using ANS.1 *(Abstract Syntax Notation 1)*. This is a language that allows a succinct definition of the content in terms of numeric or alpha characters. It takes a little practice to be able to read the descriptions because ANS.1 is very much like a programming language. If it becomes necessary to delve into the contents of the MIBs, we recommend one of the books in the bibliography.

The content is organized according to a logical structure, called a *Structure of Management Information,* or SMI. The actual SMI structure is defined internationally by the RFC as a tree, using branches of the tree for various organizations. Figure 32-4 is a partially filled-in example of this tree. Each data element is unique because its path from the root through the various branches and twigs to the leaf is unique. Vendors may choose to customize the content of their MIBs under their internationally assigned vendor number.

Those people who are familiar with computer file system structures will recognize this tree. Whether you call the nodes or levels "folders" or "directories," one contained within another allows different leaf notes to have the same name, yet be unique because the path to that data is different. E-mail presents a simple example of this. JohnSmith@BigCompany.com and JohnSmith@BigUniversity.edu are different people with a common name. They are distinguished globally because the path to each is different. (First, you must go to the .edu as opposed to the .com domain and then you go to the BigCompany or the BigUniversity).

Figure 32-4 shows the internationally agreed to structure and the numbers to each level. Like a file system, this sequence is read from the highest level (tree root) to the lowest level leaf, where the actual data element resides. For example, you can find out what kind of a system you are managing by going to 1.3.6.1.2.1.1.1. This is the path: iso.identified-organization.dod.internet.management.mib-2.system.SysDescription.

The SNMP data are contained in 1.3.6.1.2.1.11.

Vendor specific MIBs are under 1.3.6.1.4.1, that is: iso.identified-organization.dod.internet.private.enterprises.

Table 32-1

Relevant standards that apply the network management

RFC	Definition
1155	Structure and identification of management information for TCP/IP-based Internets
1156	Management Information Base for network management of TCP/IP-based Internets
1157	Simple Network management Protocol (SNMP)
1158	Management Information Base for network management of TCP/IP-based Internets: MIB-II
1159	Message Send Protocol
1160	Internet Activities Board
1161	SNMP over OSI
1270	SNMP communications services
1271	Remote network monitoring Management Information Base
1900	Renumbering needs work
1901	Introduction to community-based SNMPv2
1902	Structure of Management Information for Version 2 of the Simple Network Management Protocol (SNMPv2)
1903	Textual Conventions for Version 2 of the Simple Network Management Protocol (SNMPv2)
1904	Conformance Statements for Version 2 of the Simple Network Management Protocol (SNMPv2)
1905	Protocol Operations for Version 2 of the Simple Network Management Protocol (SNMPv2)
1906	Transport Mappings for Version 2 of the Simple Network Management Protocol (SNMPv2)
1907	Management Information Base for Version 2 of the Simple Network Management Protocol (SNMPv2)
1908	Coexistence between Version 1 and Version 2 of the Internet-standard Network Management Framework
1909	An Administrative Infrastructure for SNMPv2

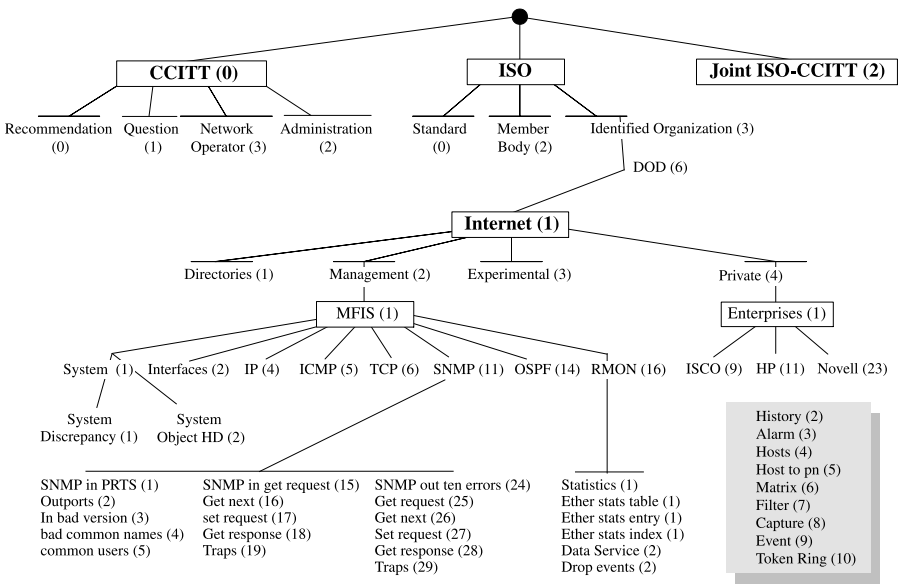

Figure 32-4
SMI structure defined as a tree

Architecture

Commands are sent from the network management function to the agent, either to set parameters, or to collect status information. When status is requested, the agent replies with a Response message. The agent is also capable of sending an unsolicited status message if some previously defined threshold is exceeded.

Figure 32-5 shows the architecture of the SNMP function and how the functions are related to the ISO's OSI model.

As shown, SNMP runs on top of a standard Internet protocol stack where *User Datagram Protocol* (UDP) is used at the transport. UDP is a connectionless protocol and, like IP, does not set up a connection between communicating entities. In a query/response protocol like SNMP, a connection is not necessary. If the target device fails to respond to a query, simply send the query again. If it were necessary to set up a connection for short queries, it would actually make the system slower.

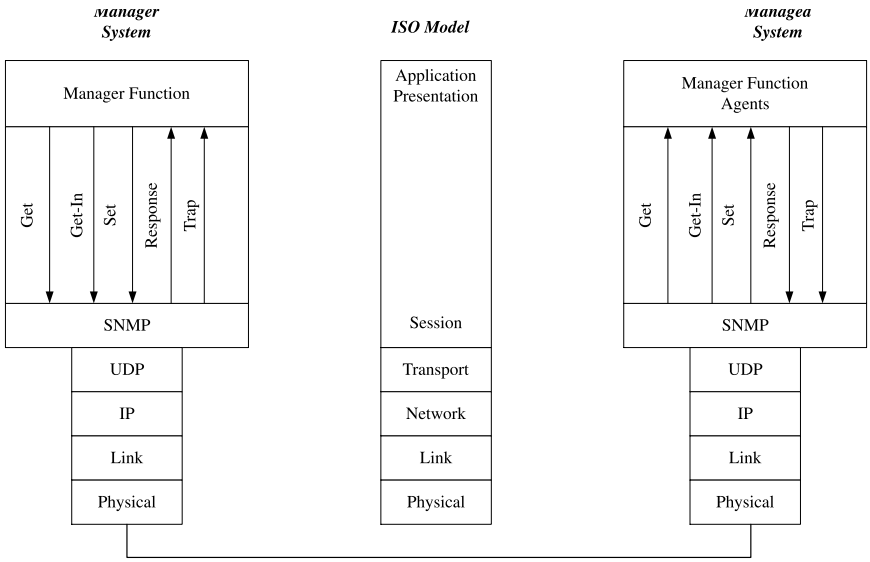

Figure 32-5
SNMP function and how they are related to the ISO's OSI model

The underlying physical network is irrelevant because it can be Ethernet or a Frame Relay network. From an OSI point of view, SNMP and its interaction with the manager and managed functions encompasses the upper three layers of the protocol stack. A discussion as to which functions belong to which layer usually winds up in a religious argument, so we will consider that SNMP and the management functions are an undivided upper layer set of protocols.

The commands and responses are indicated in Figure 32-5. The actual names (and assigned numbers) for them are delineated in Figure 32-4.

Figure 32-6 shows how the SNMP message is carried nested within the protocol layers. Since our discussion must remain general, we take it that the link layer protocol typically has some leader and header information and some trailer bits. The link layer frame carries the IP packet across the

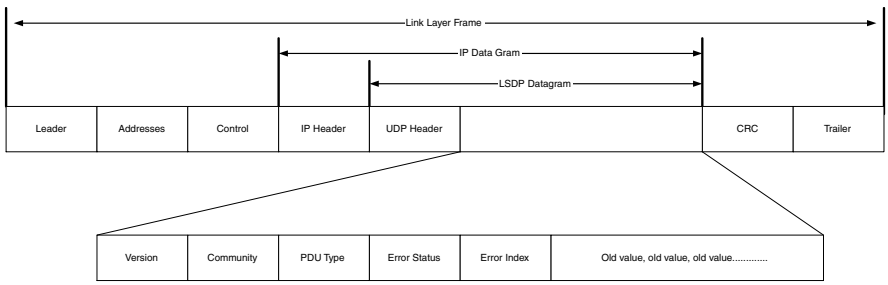

Figure 32-6
The SNMP message carried within the protocol layers

link layer connection to the next router. The IP datagram header has the routing information that lets the routers direct this packet to its final destination. The UDP header, frankly, doesn't do much. The SNMP message is contained in the UDP datagram. The version field insures that we are conversing with another agent of the same version. The community field is important because it is the security function. Our SNMP function can therefore only collect information from members of our own community. This prevents your competitor from picking the brains of your equipment attached to the Internet. The PDU *(Protocol Data Unit)* type specifies whether there is a GetRequest, GetNextRequest, SetRequest, Response, or a Trap. The error fields are used to identify SNMP errors, for example, tooBig, noAccess, or badValue. The error pointer points to the location of the offending data field. Each *Object Identifier* (OID) then is included, followed by its value (see Figure 32-4).

It is clear that this is a Simple Network Management System. The messages are simple and straightforward, and all of the items in the MIB are defined either by the standards bodies or by the vendors themselves. (If it is so simple, why are the books on the subject two inches thick?)

RMON, or *Remote Monitoring,* although not nearly as widely used as SNMP, is simply another mechanism for keeping a database on the performance of network elements. Figure 32-4 breaks out a small portion of the Ethernet statistics, part of the statistics category under RMON. These tables exist for Token Ring as well.

Network Management System Issues

If one had to ascertain the performance of one's network by evaluating the detailed content of each of these message types, it could drive one to drink. Fortunately, there are network monitoring systems (the network management function) with GUI interfaces. Unfortunately, these products are not fully mature and easy to use. In the following paragraphs, we provide some insight as to what to look for when purchasing one of these systems.

Bundling

It has always been in the interest of the seller to bundle features into feature packages. They can then charge more for the package because it contains so

many features. Unfortunately, the package usually contains only a few of the features that we want. It is in the interest of the user then to have things unbundled so that you only need to buy the specific features that you want.

The first question that needs to be asked of the vendor is whether his feature sets are bundled. What is included in the basic package? What features are extra? Is installation included? Is implementation support included? (As indicated above, the installation and configuration of a network management system is not easy for the uninitiated. You can anticipate the need for plenty of vendor help.)

The GUI

The fact that the network management has a Graphical User Interface does not mean that it is intuitive or even easy to use. Each vendor has a different approach. The people who use the tool to watch the network don't necessarily understand the interrelationships of all of the protocols and elements that are being managed. The GUI should permit high level management to be easily accomplished, while providing the capability to see deeper into the system as the operators become more sophisticated.

Network Size

How large a network will the network management system handle? Most network management systems will handle the average to small network with no problem. It should not be assumed that the network management system is infinitely extensible. This issue goes to the core of the network management system's internal database design. (Ideally, the purchaser/user/operator shouldn't have to know or worry about this, but limitations in the database design will affect the number of nodes that can be handled.) We point out here merely that the design of the database affects the efficiency with which data can be extracted. It, therefore, has a direct impact on system performance. Performance can be improved by keeping the entire database in memory. Unfortunately, the size of the memory either limits the size of the database or creates a performance impact when the database can no longer be contained within the system memory. The other alternative is to grow the system memory to match the size of the network. In the worst case, the network management system application may become unstable

and crash as the database size is exceeded. (This behavior has been noted in published tests of the leading network management system products.)

The user of the GUI, therefore, should not have to know how the database objects are structured in order to retrieve information. For instance, it is frequently desirable to be able to compare the performance statistics of one device with another similar, but not identical, device. This may be accomplished with varying degrees of success with different products. Although routers and bridges perform different functions, it is frequently useful to be able to compare the total number of frames, or packets, handled by each one as the load on the network varies. Another useful metric is the number of errored frames, or packets, per period of different network loads from these different devices. Unfortunately, not all network management systems permit such comparisons. In some cases, it is impossible; in others, it is merely difficult.

Another important question is: What percent of your network bandwidth is used by the network management system? Ideally, it should be one percent or less. This is significant since most SNMP data are collected by polling (sending GetRequests). Trap data only arrives if the thresholds that have been set in the devices have been exceeded. Therefore, it is important that the network management system have a user programmable polling priority system so that important devices, like routers and firewalls, are sampled more frequently than end devices, such as workstations. This maintains the current status of the important devices and eventually collects information on the end stations, while minimizing the impact on network traffic by the network management system.

Web Enabled GUI

Being able to view reports via a browser eliminated the need to collect, duplicate, and distribute reports. With this feature, anyone who wants the report (and has the proper permission) may collect the report from the network management system. This worthwhile feature is far from universal.

Alarm History

Keeping a history of the exception condition is so useful that we count it as a necessity. It is not only necessary to be able to identify problem areas and

equipment, but also to keep a history of similar problems on that and similar equipment. Keeping history alone is not sufficient. The network management system must permit searching the data and displaying historical trends. Trend analysis is the key here. Remember that the goal of a network management system is to maximize uptime. This cannot be achieved by reacting to problems. It can only be accomplished by anticipating problems. Therefore, in addition to keeping a historical record that lets us find out when and where this problem last appeared, the network management system can, with the proper software, do a linear regression on the historical data and predict when (but probably not where) the problem might next appear. (If, for example, the problem were a hardware failure associated with a particular component, the frequency of failure might tell us when it might again appear. If the problem were traffic related, for example, buffer overflow, and the consequence was a system slow down or crash, then the historical traffic data could allow us to identify potential trouble spots that exhibit similar traffic profiles to the failed node. Then we could go back and look at traffic through the failed node from last week/month/year and use that data as a threshold, or trap value, to predict failures of similar nodes under similar traffic conditions.)

Alarm Presentation

Alarms must be programmable and presented in a logical fashion so that they convey the relative importance of the message. Programmable filters are a "must" to permit concentration on the most important alarms first.

Statistics

It should be noted that the more statistics the network management system keeps, the more data it must collect from all components (agents) in the network, and therefore the more traffic will be added to the network by the network management system. Again, the goal is to try to keep the bandwidth requirements of the network management system down to one percent of network traffic. In small networks, this is easy to achieve. In networks of thousands of nodes, a thoughtful use of alarms and a prioritized polling sequence must be employed to achieve this goal.

Free Trials

There is no better way to learn about a network management system than to live with it for a few months before you buy it. This provides the opportunity to find out how user friendly the network management system really is. The vendor's demonstration always looks good. They showcase their strengths and minimize their weaknesses. Their presenter is thoroughly familiar with the product, and can really make it sing. (How long did it take her to become that proficient?)

In reality, it is often difficult to take advantage of a prepurchase trial offer. Although the vendor is willing to let you try the software for a few months, it takes management commitment to dedicate resources within the organization to this trial. A network management system is not something that can be evaluated in someone's (nonexistent) spare time. Remember, it isn't the initial outlay that is important, but the ongoing lifecycle costs.

Network Mapping

Much is made of this "gee whiz" feature. And, to be sure, it is a convenience to have a network management system that will create a network map. (One could argue that we should have been keeping the network map up-to-date as we designed and grew the network. Unfortunately, most networks and especially LANs were never really designed, they just grew.) How the mapping function works and what information is available from the map is more important than the map itself. First, the map should be readable. Second, it should permit zooming in and out. In the "show-me-the-whole-network" mode, the map of a large network will be too dense to read. It should provide the capability to zoom in until the subsystem of interest is displayed.

The mapping function from different vendors works with varying degrees of success. There are two basic ways of implementing the mapping function. The network management system can use SNMP information from each agent, or it can observe the traffic on the network. The SNMP approach yields a map more quickly, but can only display managed devices. Tracking traffic flow can theoretically find every interface or port on the network, but could take months of operation to finally determine a map. (In some cases, it is impossible to create a map.)

Thus, two points must be made. First, accumulating the information to draw the map is not a 10-minute job. Depending on the size of the network, it could take hours to days to months. Second, the accuracy of the map is dependent on how extensively each node on the network supports SNMP. For example, several prominent vendors don't support standard MIBs. As noted, there are standard MIBs and vendor-specific MIBs. In order for your network management system to utilize the vendor specific MIBs, they must be loaded into the manager function. For normal (here read "simple") networks, the mapping function works well and fairly quickly. That means, it works for Ethernets connected with routers and bridges, where each interface on the router is its own subnet. Today, however, we tend to use fully switched Ethernets where each device is on its own switched port from a LAN switch. The advantage is that each port is buffered so that there are no collisions on the network. A second advantage is that one can create virtual LANs (VLAN) where devices on different ports of the switch can be made members of the same broadcast domain or subnet. This permits you to design traffic flows on the various LAN segments in order to minimize response times and keep broadcast traffic confined to those devices that are members of the same logical net. The disadvantage from a network management system point of view is that it is between difficult and impossible for the mapping function to create a map because there are managed devices hiding behind, or within, other managed devices. This is especially true if the LAN switch doesn't support transparent bridging or standard SNMP MIBs. The next fly in the ointment is the development of ELANs *(Emulated LANs).* These are similar to VLAN *(Virtual LANS),* where we are emulating LANs by using an ATM switch and edge devices that are LAN switches. Here again, the ATM switch and edge device may not provide the MIB support necessary to build an accurate map of your network.

We tend to expect too much from the mapping function. Consider the difference between the logical and physical network. In the first case (see Figure 32-7), the logical and physical networks are the same. We would expect the mapping function to find or draw this picture.

In Figure 32-8, we have a common configuration for today's networks. All the devices (1–9) are on the same IP subnet. However, for traffic reasons there are only two VLANs. A and B segments comprise one, while the other is made up of segments B and D. For example, devices 1, 2, 3, and 5 are in one broadcast domain, while devices 4, 5, 6, 7, 8, and 9 are in another. This appears logically as shown in Figure 32-9. Different vendors' network management systems will yield different network maps in this case. The best (for example, as close as you are going to get) will be from a network management system that watches network traffic.

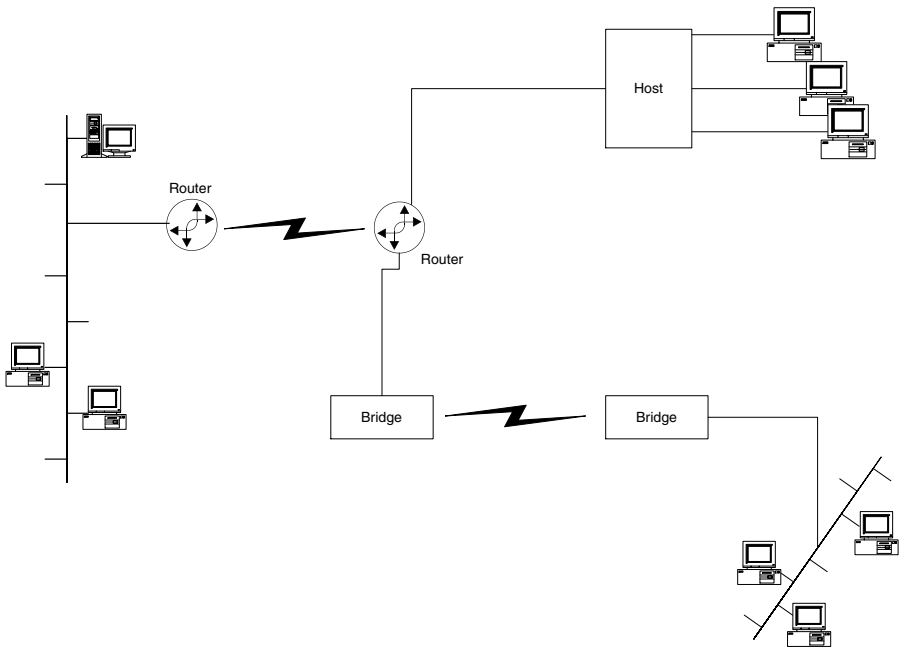

Figure 32-7
Logical and physical networks

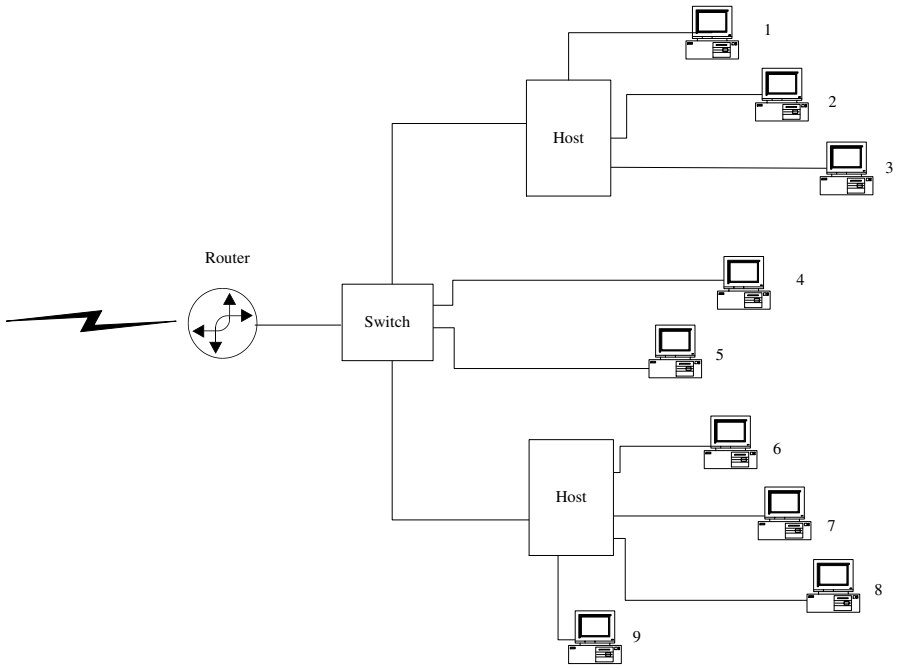

Figure 32-8
The common configuration for networks

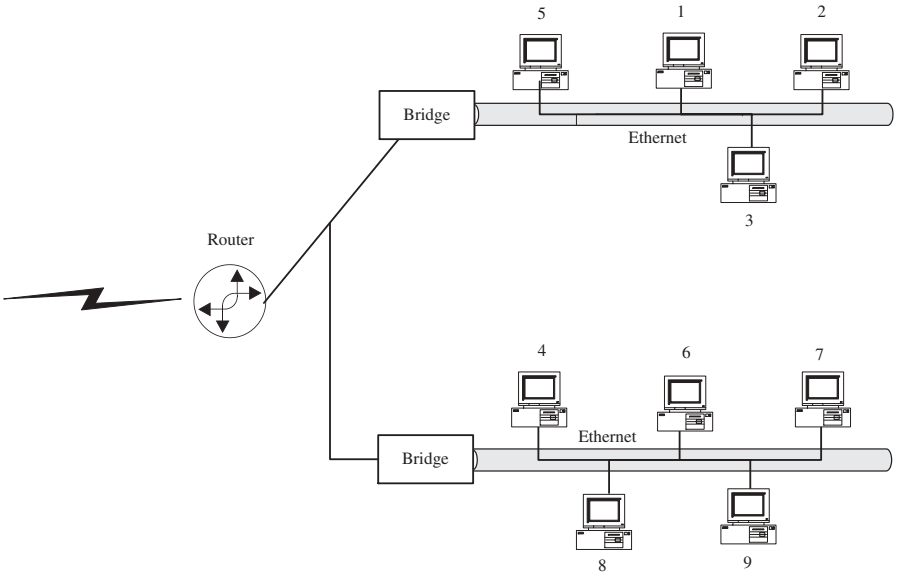

Figure 32-9
Two VLANs

An exercise left to the reader is to make up a traffic scenario and then see what you can learn about this from the other side of the router. If the MIB in the switch has good data, you are in fairly good shape. If it doesn't support the standard MIB, it is hopeless. Although most of the industry has supported SNMPv2 for several years, there has always been contention among the vendors and a significant amount of noncompliance. The noncompliant vendors are opting for their own custom MIB.

SNMPv3

Part of this contention is due to shortcomings of SNMPv2 that require fixing. There are also some additions that are needed to support higher speed networks. SNMPv3 adds a GetBulk command, a better Set command, a unique ID for each SNMP agent, and 64-bit counters to accommodate Gigabit Ethernet. Perhaps the most important new feature is the addition of real security in the SNMP packets. The GetBulk command essentially says "gimme the whole MIB." This is important because it reduces the polling traffic that was created by multiple GetNext command. While this will have

a minimal impact on reducing the network management system traffic on a small network, it will be a major benefit to large networks.

Although MIB data can be fetched remotely, there is no way to remotely manage SNMP agents. SNMPv3 adds this feature, along with the ability to describe agents within agents (the problem identified in Figures 32-8 and 32-9). Namely, the problem is that switches and hubs behind other switches and routers are difficult (to impossible) to find. SNMPv3 provides the mechanism to do this *if* (and it is a big "if") SNMPv3 is supported by all the vendors involved. The same problem exists here as with the previous versions of SNMP—standards compliance is not universal. Many vendors have opted for vendor-specific MIBs (shown in Figure 32-4) under 1.3.6.1.4.1. This means that the network management system must have a current complete copy of the MIB in order to be able to manage the device (agent). The *Desktop Management Task Force* (DTMF) is trying to standardize the various data types in a more useful form via the *Common Information Model* (CIM).

Although the forgoing seems critical of SNMPv1 and v2, one must remember the era in which they were born. The functionality and memory in managed devices (for example, hubs) were extremely limited. Today, with cheap memory and processing power in every device, many more capabilities can be added at little or no cost.

Security

The initial design of SNMP included a modicum of security by using the community field (see Figure 32-6) to identify devices belonging to our management community. This was satisfactory before the days of inexpensive and capable network sniffers. With today's technology, it is not difficult to collect SNMP packet header information and either steal the information or substitute the content. Stealing content allows the hacker to build a map of our network for later exploitation. Substituting the content of our Set commands could be disastrous. One hopes that there are no hackers on our internal network, but in today's world, one is never sure. Our paranoia turns to prudence if we are running interconnections of our network over the public Internet without the benefit of a *Virtual Private Network* (VPN). Before becoming too paranoid, one must remember that the hacker must have physical access to the network backbone, a sniffer, the time and will to hack, and the desire to do mischief. The probability of the confluence of all these attributes is relatively low in absolute or real terms.

Given our paranoia, if the hacker is able to collect frames and packets, he can collect the community information and therefore control any of our devices that are preconfigured for remote management.

SNMPv3 solves this problem by encrypting the packet content and by establishing an authentication mechanism. All this is possible today because memory and processing power are readily available in the managed devices.

The new RFCs 2274 and 2275 provide for a *User-based Security Model* (USM) and a *Views-based Access Control Model* (VACM). These RFCs set forth a scheme for today and provide for future expansion to include possibly public key authentication and directory integration.

The User-based Security Model uses a distributed security control mechanism with user name and password disseminated from a centralized system. The user list, or access list, is distributed (securely using encryption) to each managed element. The remote agent does enforcement. Thus, the network management system user must log into each managed agent. Access to each MIB element can be put under password control. The User-based Security Model specifies authentication and encryption functions, while the Views-based Security Model specifies the access control rules. Each managed device can therefore keep a log of accesses and by whom (just as a firewall or proxy server logs each access). While perhaps not important for hubs, it is very important for sensitive devices such as routers, switches, and firewalls. Previous versions of SNMP could not create this important audit trail.

In addition to encrypting the packet contents (for example, using DES, the *Data Encryption Standard*), each packet contains a time stamp to synchronize the network management system with the managed agent. This prevents the man-in-the-middle from recording the queries and commands, analyzing them, substituting his content, and playing them back at a later time. The User-based Security Model also specifies its own MIB so that the passwords and user name can be remotely and securely maintained. Thus, although enforcement of access is done by each MIB agent, the user name and password can be centrally maintained and distributed to each managed agent by using the specified encryption technique. This means that a network authentication server must exist just for SNMP.

In the final analysis, it is your choice whether or not to implement the security features. There are definite benefits and costs associated with installing and maintaining the authentication server and its database.

JAVA

Sun Microsystems is pushing Java as a mechanism for enhancing the flexibility of SNMP. As we have seen, SNMP defines MIB-1, MIB-2, and MIB-3. Even though we can capture data on higher speed elements and control network components securely, if changes are needed to the SNMP agents, software must be loaded (typically, manually) in each managed device.

The Java concept is that now each agent has memory and CPU cycles to spare. Why not have it as a Java virtual machine? The network management system would then download the new functionality to each or all devices.

SNMP was designed to manage elements like routers, switches, and hubs. Managing remote servers, desktop machines, or applications processes was never considered. The introduction of a Java-based system permits the management of higher layer functions, while maintaining compatibility and coexistence with SNMP.

There are several problems to overcome before Java can become a popular network management system tool. The first is performance. As an interpretive language, it is processor intensive and slow. Solutions for this problem involve compiling the Java code on the fly as it is downloaded to the target machine. The code then runs in native mode. Sun's Hot Spot compiler and Microsoft's *Just in Time* (JIT) compiler are designed to do this.

The good part is that the network manager can easily maintain the same revision level of the agents in all devices throughout the network. Unfortunately, this threatens the market for vendor-specific network management system solutions. The best advice is to stay tuned as vendors and standards organizations try to solve the problems of complex networks of today, and the even more complex networks of tomorrow. History has shown that during the early stages of development and deployment, there are multiple solutions from a variety of vendors—all incompatible with one another. As the technology (and markets) mature, the industry hones in on the essential features and functions that are accepted and supported industry-wide. These then become the core functionality on top of which each vendor builds his proprietary extensions.