Managing Enterprise Networks

Karl Jeacle B.A.(Mod)

M.Sc. Computer Applications

Dublin City University
School of Computer Applications

Supervisor: Brian Stone

August 1996

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of M.Sc. Computer Applications is entirely my own work and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

Signed: ID No: 94157294

Date: 21 August 1996

Contents

List of Figures

Abstract

In the last few years, managing, connecting and internetworking local and wide area networks has become an increasingly difficult task. Gone are the days of 10Mb Ethernet and X.25 — today we face Switched Ethernet, 100-Base-T, FDDI, ISDN, Frame Relay, SMDS and ATM. Network devices which implement these technologies are increasing in complexity, while networks which use them are growing at a rapid pace. Good management is now an essential component of any enterprise network.

This thesis examines the issues faced by network managers given the task of managing today's enterprise networks. It provides an overview of current LAN and WAN technologies, and the type of products available which help manage these networks. It explores the relationships between, and explains the merits of, enterprise networking protocols from both standards bodies and the computer industry. It also profiles industry groups such as the Desktop Management Task Force.

Finally, the impact of new technologies such as the Common Object Request Broker Architecture and Internet systems such as the World Wide Web and, in particular, Sun Microsystems' Java language are examined to assess how these technologies can impact the future of enterprise network management.

Chapter 1

Introduction

1.1 Enterprise Networks

In the last few years, managing, connecting and internetworking local and wide area networks has become an increasingly difficult task. Gone are the days of 10Mb Ethernet and X.25 — today we face Switched Ethernet, 100-Base-T, FDDI, ISDN, Frame Relay, SMDS and ATM. Network devices which implement these technologies are increasing in complexity, while networks which use them are growing at a rapid pace.

Such advances have brought great benefits to the end-user and these complex data networks are now considered essential components of any successful organisation. However, there has been a lack of co-operation and co-ordination between network equipment vendors in providing network managers operating these large-scale enterprise networks with a single, uniform, and coherent network management platform [17].

1.2 Thesis Aim

The aim of this report is to examine the issues faced by a network manager given the task of managing today's enterprise networks. This is achieved as follows:

- Provide an overview of current LAN and WAN technologies.
- Explore the relationships between, and explain the merits of, enterprise networking protocols from both standards bodies and the computer industry e.g. SNMP vs CMIP.
- Examine the products available which help manage enterprise networks e.g. IBM NetView 6000, HP OpenView, Sun Solstice.
- Examine what the future of enterprise and systems management – who and what are the Desktop Management Task Force?
- Explore how new technologies such as the OMG's Common Object Request Broker Architecture and Sun Microsystem's Java language can impact on the future of enterprise network management.

The primary aim of the report is to provide the uninitiated with a comprehensive overview of the diverse problems which lie in the domain of enterprise network management.

Volumes have been written on each of the topic areas covered, but rather than alleviate the problems at hand, this wealth of information on specific topics often leads the newcomer to lose sight of the overall problem domain, and gives rise to unnecessary confusion.

Ch.	Title	Contents
7	Java	Integrating the Java language
6	Objects	OMG's CORBA
5	Organisations	Desktop Management Task Force
4	Applications	NetView, ForeView, Etherman
3	Protocols	SNMP, CMIP and RMON
2	Networks	Ethernet, FDDI, ISDN, ATM, etc.

Figure 1.1: The 'Report Layout' Stack

1.3 Report Layout

The report structure can be described, rather appropriately, in terms of a networking protocol stack, as illustrated in figure 1.1. Successive chapters progress their way up the stack, thus forming a comprehensive overview of all the areas pertinent to enterprise network management.

Chapter Two discusses the *network technologies* which make up today's enterprise networks.

Chapter Three describes the *protocols* from standards bodies and the computer industry which are in widespread use today.

Chapter Four discusses the *applications* used by network managers when managing a combination of diverse networking technologies in a single intranet.

Chapter Five profiles an *organisation* called the Desktop Management Task Force who play a major role in determining how network management procedures are implemented.

Chapter Six examines *object technology* and in particular, the CORBA standard which has been embraced so rapidly in recent years by the computing and telecoms industries.

Chapter Seven looks at how the *Java language* can be used to improve techniques for building network devices so as to be more manageable by the enterprise network manager.

Chapter Eight is a *summary* of the report.

Bibliography

1.4 Acknowledgments

Thanks go to my supervisor, Brian Stone, for his patience and understanding during the dissertation process. Thanks also to the management team at Broadcom Éireann Research Ltd for their encouragement and financial support throughout the entire M.Sc. course.

Chapter 2

Networks

2.1 Introduction

The complexity involved in managing an enterprise network is due to the diversity of network technologies in widespread deployment in today's corporate networks.

This chapter overviews some of the most popular local area and wide area networks available.

2.2 Local Area Networks

Managing a single local area subnetwork is not an overly complex task. It is the protocol and network level interoperability issues and the interconnectivity problems which arise when scaling these small networks that are the challenge.

In the subsections below, we will look at five LAN technologies which can be used for both desktop/backbone and low-speed/high-speed solutions: Ethernet, Token Ring, FDDI, Fast Ethernet and ATM.

2.2.1 Ethernet

Ethernet is the most common form of LAN technology in use today. It is a 10Mb/s bus technology which uses CSMA/CD to arbitrate host access to the transmission medium [33]. While most commonly used over coaxial cable, it is now often used over a variety of grades of copper cable (and known as 10BASE-T).

When using coaxial cable, connecting hosts to Ethernet is as simple as adding an adapter card to the host and using a 'T' connector to attach to the shared bus. When using copper cable, Ethernet is wired in a star topology with the shared medium located at a central hub. Each attached host has individual copper cable connections to this hub.

Transmitting with CSMA/CD (Carrier Sense Multiple Access / Collision Detect) is done by listening to the bus before transmission, and waiting for an idle period. When the shared medium is free, the host transmits and listens while transmitting. If another hosts attempts to transmit at the same time, a collision occurs, and the host will transmit a jamming signal and back off for an arbitrary amount of time before attempting retransmission. This is called Truncated Binary Exponential Backoff. Essentially, the host will wait longer and longer between each transmission attempt if collisions keep occurring.

A recent addition to the Ethernet technology arena is Switched Ethernet. The network is segmented into a number of hosts or subnetworks with each having their own dedicated 10Mb/s Ethernet. Each of these are then connected to a central Switched Ethernet hub. Obviously, this is only possible when the network is used in a star configuration.

The advantage with this switching is that traffic can be localized to departments in a corporate network. High speed uplinks from the switched hubs can provide increased bandwidth to file servers.

A major drawback for the network manager, however, is that it is much harder to 'snoop' on network traffic for analysis purposes. It is no longer a shared medium, so not all traffic is visible on all hosts. Ascertaining what traffic is flowing where, is suddenly more complex.

2.2.2 Token Ring

During the 1980s Token Ring was the main competitor to Ethernet. Based on a ring topology, Token Ring operates at 16Mb/s and uses a token passing scheme to arbitrate host access to the shared ring [16].

A big win for Token Ring over Ethernet is not just its higher speed, but the use it makes of its available bandwidth. Because only one host can be active on the ring at any one time, and there are no collisions to make hosts backoff, Token Ring networks can achieve 100% utilization of bandwidth compared with Ethernet's 70-80% peak usage.

Because of the low cost of Ethernet and its ease of use, Token Ring is less popular today in the local area. Newer high-speed backbone technologies such as FDDI, however, owe their roots to its token passing technology.

2.2.3 FDDI

FDDI (Fibre Distributed Data Interface) is a 100Mb/s network technology which, like Token Ring, uses a ring topology and token passing for its access method

[39]. Although, originally designed to operate as a high-speed LAN backbone, FDDI has had *some* success as a desktop solution with its copper based variant, CDDI.

Because of its backbone bent, FDDI includes a capability for dual counter-rotating rings, which can provide a redundant data path in the case of a cable failure between two nodes on the ring.

FDDI nodes can be either 'dual attached' or 'single attached'. Dual attached stations are connected to both rings and tend to be servers or network hubs, while single attached nodes are often desktop computers connected via copper into an FDDI concentrator on the ring.

FDDI is the widely acknowledged industry leader for LAN backbone interconnection, but it is expensive and considered difficult to install. Its crown is therefore under attack from Fast Ethernet and ATM.

2.2.4 Fast Ethernet

Also known as 100BASE-T, 100BASE-TX and 100BASE-T4, Fast Ethernet is intended to be a 100Mb/s extension to 10BASE-T. Like regular Ethernet, it uses CSMA/CD to determine when nodes are allowed access to the network [39].

The three names for Fast Ethernet are the result of different physical layer implementations available for 100BASE-T. 100BASE-TX implementations operate using two cable pairs of Category 5 unshielded twisted pair (UTP) cable, two pairs over Type-1 shielded twisted pair (STP) cable, or two optical fibres. 100BASE-T4 implementations operate using four cable pairs over Category 3, 4 or 5 UTP (the higher the category, the higher the quality of the cable).

Because of its backward compatibility with 10Mb/s Ethernet, Fast Ethernet provides a perfect upgrade path for the massive installed base of Ethernet users. Combined '10/100' cards are available for desktop computers which will automatically detect what kind of network is attached and run at the appropriate speed. This means new machines can be bought and installed on 10Mb/s Ethernet systems and then transparently move to 100Mb/s when central hubs are upgraded.

Fast Ethernet is the obvious choice for the next generation of desktop computers. FDDI is the current king of the backbone and can provide higher availability and utilization levels than Fast Ethernet. But as mentioned above FDDI also has its problems, because of its low-cost and industry backing, Fast Ethernet looks set to become dominant in both the desktop and backbone markets. Only one technology can possibly challenge this: ATM.

2.2.5 ATM

Asynchronous Transfer Mode (ATM) is an evolving network standard initially proposed for wide area networking by telecom operators which is now being promoted for use in LANs. ATM uses a switching technology that offers dedicated end-to-end connections between nodes. Fixed length (53 bytes) 'cells' are used rather than variable size 'frames'. ATM's big win is that it offers scalability to operate at speeds from 1Mb/s to several Gb/s.

ATM is a completely switched solution. All attached stations must have a direct connection to an ATM switch, and only nodes attached to the same switch or indirectly accessible via one or more intermediary switches can be contacted using a true ATM connection.

Two versions of ATM are being offered in the LAN marketplace: 155Mb/s and 25Mb/s. While 155Mb/s is considered the norm for a workstation ATM connection, many have argued that in addition to costing too much, ATM networking equipment provides more bandwidth than today's average desktop machine requires [18]. To combat this, 155Mb/s is being delivered to the desktop for high-end workstations and backbone applications, while 25Mb/s is being offered as an alternative upgrade to Switched or Fast Ethernet for existing 10Mb/s Ethernet users.

ATM has a number of problems to overcome before becoming dominant in the local area. Cost is naturally an issue, but there are still too many large network and protocol interoperability questions hanging over ATM for it to be embraced wholeheartedly.

2.3 Wide Area Networks

The technologies used in creating wide area networks are often the domain of the local telephone company. A network manager does not have to worry about how these long distance circuits operate *per se*, but has to understand how to interface the local area networks to the wide area, and what implications this has when routing traffic originally intended to be seen on the local area only.

In its most basic form, a wide area network can be built with a single point-to-point leased line from the telephone company. In recent years however, the number of 'services' available has increased greatly. Below we will look at X.25, ISDN, Frame Relay, SMDS and ATM.

2.3.1 X.25

Now in decline, X.25 was one of the most widely used wide area networking technologies. It is a packet switched network which was predominantly run by government and national telecom operators [38].

X.25 comprises a set of three protocols:

1. The physical layer performs the transfer of serial data streams between the DTE and DCE. This protocol is defined as X.21.

2. The link (frame) layer provides the packet layer with an error free packet transport facility. It performs link setup and closedown, creates logical connections and provides error and flow control.

3. The network (packet) layer is concerned with the reliable transfer of the transport layer messages and with multiplexing multiple virtual circuits onto a single physical link.

As mentioned above, X.25 is no longer a big favourite with network managers. It is still widely available in most countries, but lack of speed and the advent of new, faster, cheaper networks have left it lagging.

2.3.2 ISDN

In 1976 the term ISDN (Integrated Services Digital Network) appeared in the CCITT's (now ITU-T's) Orange Book list of terms. It took so long for it to become available reliably in most parts of the world that a number of rather cynical meanings were suggested for its abbreviation. In the 1990s, however, ISDN has become available from most telecom operators.

ISDN can be thought of as a digital version of the regular telephone network that deals in 64Kb/s data channels rather than 3.1KHz analogue voice channels [21]. Access to the network is delivered in two forms: basic rate (BRA) and primary rate (PRA).

BRA specifies a single access point into ISDN. Known as 2B+D, BRA consists of two bearer channels and one data/delta channel. Each bearer channel operates at 64Kb/s and is a clear channel. The delta channel operates at 16Kb/s and is used for signalling and control information.

PRA is usually used to connect multiple users to ISDN. A common application would be to connect a PBX, LAN or other multiuser switching device to an ISDN network. In Europe, primary rate access consists of 30 'B-channels' of 64Kb/s each and one 'D-channel' of 64Kb/s.

ISDN's strength in the corporate wide area network is its speed in bring up connections. ISDN is relatively cheap to run, costing only minimal line rental after installation. Outgoing lines can remain idle and be brought up automatically on-demand when additional bandwidth is required or when an existing leased line failure occurs.

2.3.3 Frame Relay

With the slow demise of X.25 and the expense of multiple meshed leased line networks, Frame Relay has become a popular choice for corporate networks that need a large number of connected sites yet don't need peak bandwidth at all times [27].

Frame Relay is basically a simplified implementation of the X.25 protocol. The advantages of the X.25 protocol are present: efficient traffic interleaving and the possibility of establishing several connections over the same physical channel. But by removing the error correcting function and simplifying flow control, Frame Relay is a much leaner and faster protocol than X.25. These changes have been made possible due to improvements in the quality of digital transmission lines in recent years.

Frame Relay can offer cost savings over leased line networks as like other systems that use public networks, sites only need a local connection into the 'frame relay cloud'. Once connected to the cloud, traffic can make its way to any site in the country with a Frame Relay connection.

In addition, Frame Relay offers two levels of data rates: the Committed Information Rate (CIR) and Extended Information Rate (EIR). The CIR is a guaranteed sustained data rate from the public network, but the EIR is on-demand bandwidth above the CIR. This means that multiple underutilised 64Kb/s leased lines could be replaced with Frame Relay connections of 32Kb/s CIR and 32Kb/s EIR which would guarantee 32Kb/s from site to site, yet allow traffic to peak to 64Kb/s. Since EIR pricing is considerably less than CIR, significant savings can be made as corporate networks grow in size.

In general then, Frame Relay is used when there are a large number of sites that need to be interconnected permanently (thus excluding ISDN), but not requiring consistently high bandwidth. Typically, Frame Relay is offered at data rates from 64Kb/s to 2Mb/s. For data rates higher than this, a metropolitan area network is required.

2.3.4 SMDS

The Switched Multimegabit Data Service (SMDS) differs from the other network technologies discussed here as it is a 'service' rather than a network — a service for Metropolitan Area Networks (MANs) [30].

The most common MAN technology in use is the IEEE 802.6 DQDB (Distributed Queue Dual Bus). Somewhat similar in principle to FDDI, DQDB uses counter-rotating dual rings to provide resilience, and unique access protocols which provide near 100% loading. SMDS is the service that telecom operators run on top of DQDB to provide a rich set of features (or services) to customers.

SMDS provides access classes from 4Mb/s to 34Mb/s, and can facilitate closed user groups allowing virtual private networks to be built across the metropolitan area network.

At present, there is no major competitor to SMDS, so despite the relatively high costs of telecom operator service offerings, its future looks assured. Once again only end-to-end ATM, this time in the wide area, is making any challenge.

Because SMDS uses a packet size of 53 bytes, a migration path from DQDB to ATM is open. In many countries, SMDS will be used as the gateway to international ATM traffic, so customer requiring high bandwidth data rates between countries will connect to the local SMDS node with either a high-end router or direct ATM connection, and transfer data via SMDS over international ATM, or rather bizarrely, local ATM over international SMDS over ATM.

2.3.5 ATM

Asynchronous Transfer Mode (ATM) has always been a wide area technology, and so unlike in the LAN arena, wide area ATM is a much more mature technology offering. Standards are further advanced, more products are available, and overall deployment is greater [37].

Because of its scalable architecture, ATM technology is no different in the wide area than the local area. The only differences are in the standards used for interfacing and signalling between wide area equipment, and the speeds at which ATM normally runs. In the wide area, 155Mb/s is the normal basic rate, while speeds of 622Mb/s and above are used for interconnection of major switching nodes.

While the lure of ATM to the desktop is its speed, the wide area hopes to make use of its flexibility. ATM has oft been suggested as a panacea for all communications requirements and this is because of its ability to carry multiple media types.

Wide area ATM networking equipment is available which can carry not only data in ATM cells, but can convert analogue and digital audio and video signals into ATM cells. Multiple wide area links between corporate premises carrying separate data, PABX, and video conferencing circuits can be replaced by a single high-bandwidth ATM link. The ATM equipment can convert all these media types to ATM cells, and once a cell, can guarantee its delivery to a specified destination.

At present, ATM is being deployed for major backbone links, both in corporate (private) and public networks. An end-to-end global ATM network sounds great in theory, but given the diversity of existing networks, migration to ATM will present major problems for the network manager.

2.4 Summary

In this chapter we have seen some of the most popular LAN and WAN technologies in use today. At their cores, each of the networks operate in fundamentally different ways, yet from a network manager's perspective, we can hide this complexity and abstract to a view where each network performs a simple task of transporting traffic between end-stations.

Unfortunately, life is not always so simple. In order to manage (and troubleshoot) these networks, one must have a sound understanding of how each one of the networks operate and what their idiosyncrasies are. As technology advances, this becomes an increasingly difficult task.

When interconnected, problems begin to mount. A network manager then needs more than an understanding of how each individual network works, he also needs a 'big picture' – a means of visualizing what is happening on the network. For this, a manager needs support.

In the next chapter, we will see how network management protocol stacks might provide the basis for this support.

Chapter 3

Protocols

3.1 Introduction

This chapter outlines the concepts used at the protocol level in building a standardised network management system. It overviews the standards bodies involved in related activities and discusses the most widespread network management protocols in use today.

3.2 Standards creation

There are many different standards bodies at work in the area of network management, especially in the larger arena of public telecommunications networks. In the enterprise networking area, there are two main bodies who have influenced standards for how corporate networks can and should be managed: ISO & IETF.

3.2.1 ISO

The International Standards Organisation (ISO) plays a major role in developing network management standards. They are responsible for all OSI (Open Systems Interconnection) standards.

An ISO standard becomes a standard after a lengthy process of designs, writings, reviews and votes. It starts with a *new work item* which can take the form of a working paper from any interested party. If, after review by ISO members, it is deemed to warrant further analysis, it becomes a *working draft* and a working group is assigned to refine the material.

After several review processes, a working draft is eventually considered technically stable and becomes a *draft proposal.* It is registered and given an identifier. After more changes and reviews, it is considered both technically and editorially stable and so becomes a *draft international standard.* A short delay ensues before a vote finally renders it an *international standard.*

This whole process takes about 3 to 4 years.

3.2.2 IETF

The process for creating and approving standards in the Internet world is much faster and simpler than in the OSI world. These standards are known as *Requests for Comments* or RFCs.

RFCs can come about in a number of ways. An individual can document and submit an idea which could be accepted as an RFC. RFCs can come from commercial organisations who are working on a problem area in which they believe standardization is needed. But most often, RFCs tend to originate from the Internet

Engineering Task Force (IETF) who can be considered as the technical/research arm of the Internet.

RFCs do not go through as many formal procedures as OSI standards. Because of their extensive working groups and review processes, OSI standards go through a considerable amount of referring. RFC documents, on the other hand, have no formal referee process, but receive substantial comment from Internet members.

RFCs, therefore, can be standardised in less than a year.

3.3 Concepts

A number of key terms and concepts are common to almost all network management standards adhered to. These concepts are the foundations for today's network management systems and therefore dictate the purpose, design and implementation of network management protocols [4].

Network management standards define responsibility for a managing process (or *network management system*) and a managing agent (or agent process). Strictly speaking, a network management system contains nothing more than the protocols that carry information between the managing process and the various agent processes (network elements) which comprise the network.

One other component is vital to a network management system: the *management information base* or *MIB*. The MIB is a library or database which is shared between managers and agents that provides information about managed network elements. The MIB is structured like a tree: at the top of the tree is the most general information available about a network. Each branch of the tree then gets more detailed into a specific network area, with the leaves of the tree as specific as the MIB can get.

In summary then, a network management system comprises:

- *Agent:* reports to the managing process on the status of managed network elements and receives directions from the managing process on actions it is to perform on these elements.
- *Managing Process:* directs the operations of the agent.
- *MIB:* Used by both the agent and managing process to determine the structure and content of management information.

In a real network, agents will be implemented as software or firmware in networking equipment such as routers, hubs and switches. There can be as many agents on the network as there are network elements (managed devices). There will typically be one PC or workstation which will host the managing process and act as a 'network control station'. The MIB will usually be co-located with the managing process, with parts of the MIB pertaining to the agents located at the agent devices.

3.4 Protocols

There are many protocols available. The two mainstream protocols, however, are known as SNMP and CMIP. SNMP hails from the TCP/IP and Internet world. CMIP is from the OSI camp. They are both discussed below.

3.4.1 SNMP

The Simple Network Management Protocol (SNMP) was designed in the 1980s as an answer to the communications problems between different types of networks [6]. Its initial aim was to provide a temporary, but functional, solution until a more complete network management protocol became available. As time passed, however, no obvious successor emerged, and so SNMP became the *de facto* enterprise network management protocol [32].

As its name suggests, SNMP is a very simple protocol. Network information is passed from manager to agent in protocol data units (PDUs). These PDUs can be thought of as objects containing a number of variables.

There are five types of PDUs that SNMP employs to monitor a network: two deal with reading network element data, two deal with setting element data, and the 'trap' is used for monitoring network events such as terminal startups or shutdowns. These are described in table 3.1

Briefly, *Get Request* and *Get Response* will be the most common PDUs, being used to transmit queries and responses between the manager process and network element agents. *Set Request* will be used when the manager wants to set a variable on a device, and an unsolicited *Trap* is generated by a device when an event at the network element should be reported to the manager process.

Advantages

SNMP's main advantage is its simplicity. It is relatively easy to implement on a large network, neither taking a long time to install nor placing any major stress on the network. It is also quite simple to program, with each MIB variable consisting of just:

- the variable title
- the data type of the variable

PDU	Description
Get Request	This PDU is used to access the agent and obtain values from a list. It contains identifiers to distinguish it from multiple requests as well as values to provide information about the status of the network element.
Get-Next Request	This PDU is similar to Get Request, except it permits the retrieving of the next variable in the MIB.
Get Response	This PDU responds to the Get Request, Get-Next Request and Set Request data units. It contains an identifier that associates it with the previous PDU.
Set Request	This is used to describe an action to be performed on an element. Typically, it is used to change the values in a variable list.
Trap	This PDU allows the network management module to report on an event at a network element or to change the status of the network element.

Table 3.1: SNMP Protocol Data Units

- whether the variable is read-only or read-write
- the value of the variable

Another benefit to implementing SNMP arises from its widespread deployment. Having had little competition, equipment vendors have had no qualms in embracing such a simple standard and designing their products to be fully SNMP capable.

Disadvantages

The first problem that SNMP has concerns security. Some large gaps allow potential network intruders to access information carried on the network, and so use a number of basic techniques to perform attacks on devices attached to the network.

Because SNMP tends to be expandable, the next generation of SNMP, SNMPv2, has added some security mechanisms to help combat the three largest and most common (from a security point of view) problems with SNMP:

1. **Data Privacy** to prevent intruders from gaining access to information carried along the network.
2. **Authentication** to prevent intruders from sending false data across the network.
3. **Access Control** to restrict access of particular variables to certain users (thus removing the possibility of a user accidentally crashing a network device).

Security aside, SNMP's biggest problem is that it is generally considered to be just *too simple*. The information it handles lacks the detail and organisation required to deal with the rapidly expanding enterprise networks of the 1990s. This is no surprise, of course, remember that SNMP was originally conceived as a 'quick fix'. This issue has been addressed in SNMPv2.

SNMPv2

The new version of SNMP attempts to overcome the 'too simple' problem by allowing for more in-detail specification of variables, including the use of a table data structure for easier data retrieval [7]. Two new PDUs have been added that are used to manipulate this table data structure. If page count is anything to go by, SNMPv2 is a big improvement. The original SNMP specification was just 36 pages long. SNMPv2 runs to 416 pages.

Some have argued that SNMPv2 has lost the protocol's simplicity and hence ease of use, however, the changes were inevitable — the system was rapidly becoming out-dated and overwhelmed by today's networks.

Problems are afoot though. Having reached general consensus on a set of 11 documents comprising SNMPv2, the SNMP working group was told that it was to be disbanded. The IETF hope to restart the group by the end of 1996, and get to work on filling the vacuum resulting from the lack of secure and interoperable SNMPv2 products.

RMON

RMON, Remote (network) MONitoring, is an SNMP MIB developed by the IETF in order to provide traffic statistics and analysis on local area networks. While other MIBs usually support network devices whose primary functions are other than management, RMON was created to provide management of a network [13, 23].

3.4.2 CMIP

The Common Management Information Protocol (CMIP) was to be the protocol to replace SNMP. Backed by governments and large companies, it had a huge potential market. Unfortunately, after a grandiose design, problems with implementation have delayed its widespread introduction and deployment [36].

CMIP was intended to build on SNMP's shortcomings by becoming a larger, more sophisticated protocol. Its basic design and operation is similar to SNMP but it employs eleven PDUs for network operations compared to SNMP's five.

Variables in CMIP are much more complicated than in SNMP. Complex data structures with detailed attribute descriptions can be constructed. CMIP variable properties include:

1. attributes: which represent the variable characteristics
2. behaviours: what actions of the variable can be triggered
3. notifications: the variable generates an event report whenever a specified event occurs (e.g. device power failure)

Advantages

The major advantage of CMIP over SNMP is the behaviour associated with variables. Variable information can be used to initiate tasks that would be impossible with SNMP. For example, if a network device is continually failing to perform an operation, the CMIP variable can be set to notify a human for intervention. With SNMP, the network manager would have to manually keep track of how many failures had been logged by the device.

Another advantage to CMIP is how it addresses many of the shortcomings of SNMPv1. As discussed earlier, security it a major concern with SNMP. CMIP has built in security management devices that support authorisation, access control and logging.

It must also be noted that CMIP enjoys considerable support from the telecommunications community. As we will see below, a large problem with CMIP is its complex implementation requiring large resources. CMIP is not seen as too great an overhead for the telecoms industry given its features and the resources they have available.

Disadvantages

CMIP is resource hungry. It is many times more powerful than SNMP, but it also far more complex. Because CMIP requires such a complex implementation, network elements which run the protocol require significant processing power and memory. This might be acceptable on a well-configured workstation, but on the average router or hub, free system resources are scarce.

For this reason, CMIP has failed to be become widely adopted. While SNMP was criticised for being too simple, CMIP is being slated for having too many features. This doesn't make life any easier for programmers either. Programming CMIP variables requires more skill, experience and time than a similar task with SNMP.

CMISE

Because of its OSI background, CMIP is not intended to be programmed in the same way as SNMP. As with all OSI applications, it is recommended that the application service elements are used as interfaces to lower level network services.

The Common Management Information Service Element (CMISE) is one such element that resides in layer 7 of the OSI reference model. CMISE's function is to support OSI applications in transferring network management information from one system to another. CMISE maps its operations onto CMIP PDUs.

Thence, it is actually CMISE, and not CMIP, which programmers must learn to use in order to build OSI-based network management applications [31].

3.5 Summary

A common set of concepts and terms underlie network management. Once these are understood, it is not difficult to compare and contrast available network management protocols.

The advantages and disadvantages of the two protocols discussed greatly reflect on the two organisations (and respective standardisation processes) responsible for them. ISO took time to create CMIP, an excellent protocol, but not entirely practical. The IETF, on the other hand, encouraged the widespread use of SNMP, a simple but effective protocol.

Given the lack of available implementations of CMIP, it is no surprise that it is not a popular choice in the enterprise networking world. SNMP is showing its age, and SNMPv2 is currently in trouble. The network management community had best not look to the protocols for salvation.

Chapter 4

Applications

4.1 Introduction

There are three basic classes of network management applications which we will examine in this chapter. These are:

- Enterprise Wide — applications such as SunNet manager or HP OpenView fall into this class. They are large scale applications used by the majority of enterprise network managers to get an overall view of how the network is configured and is performing.
- Custom applications — these are lower level applications which are designed to manage a specific device. The interface used between the application and the device is often proprietary, leaving the user with no choice but to purchase this single application from the vendor.
- Research tools — this is a much wider category than pure research tools. It covers any freely distributable applications which can be used to help manage a network. Very often these tools are written by universities, and

have a particular strength or focus, such as SNMP or LAN performance monitoring.

All three of these categories are important, as each covers the management of a network at a different level. Each takes a unique view of what a network is composed of and how a network device should perform. Without combining the use of all three, it is impossible to fully analyze an enterprise network.

4.2 Enterprise Wide applications

There are a number of applications in the marketplace which fall into the Network Management System (NMS) category. These include:

- Cabletron SPECTRUM
- HP OpenView
- NetLabs DiMONS
- SunConnect SunNet Manager

The University of Michigan Future Computing Environment Monitoring Team conducted a study on how each of these platforms compared [3]. Their results are summarised in the subsections below.

4.2.1 Cabletron SPECTRUM

Currently, only Cabletron SPECTRUM has dependency support. Dependency support refers to a NMS product's ability to understand the relationship between all the devices on a network so that when one of them fails, any other devices which stop responding to NMS queries are not treated as failing because the NMS knows that they are dependent on the failing device for network connectivity to the NMS. Systems which don't understand dependencies have a tendency to set off many alarms when there is a failure of only a single device in a complex networking environment.

4.2.2 HP OpenView

HP OpenView is clearly the platform most widely chosen by third-party application developers. There are likely many reasons for this, including third-party vendor perception of OpenView's current and future market acceptance, ease of developing applications for OpenView, the need to compete with other products already available for OpenView, etc. This means that it is possible to obtain a wider range of add-on products for OpenView. SunNet Manager is in second place, but not growing much. IBM NetView/6000 is close behind Sun, but there are many fewer third-party applications available for either SPECTRUM or DiMONS.

4.2.3 SunConnect SunNet Manager

It appears that SunConnect is no longer actively developing SunNet Manager. Sun has licensed the DiMONS technology from NetLabs and it appears that Sun is now concentrating their development effort on a future product called Encompass which they are basing on DiMONS. In addition, SunNet Manager, more than any of the other products, is mainly a platform system, i.e., to make SunNet Manager very useful, it is necessary to buy third-party applications to use with it. Each of the other NMSs has more built-in capability without as much

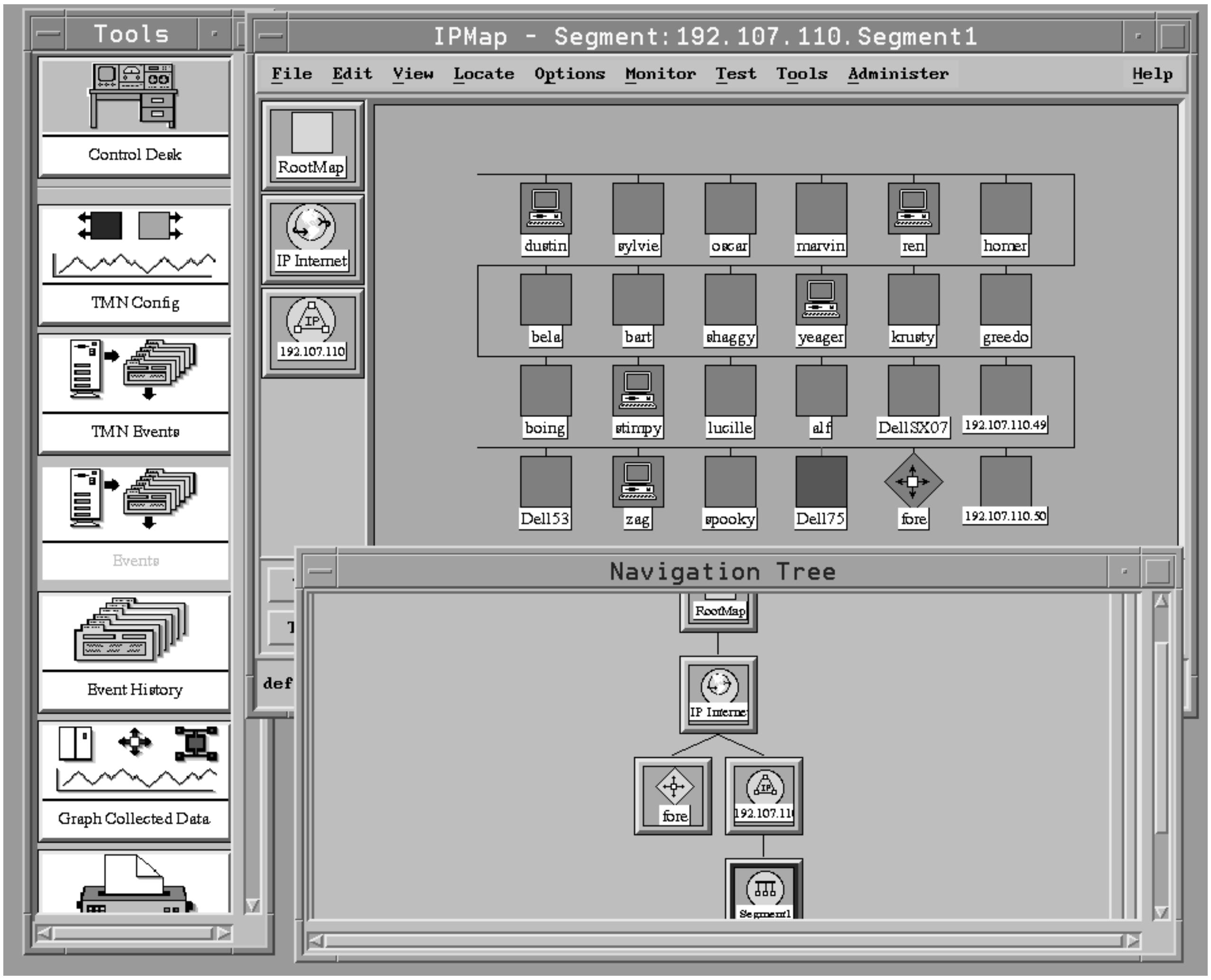

Figure 4.1: Typical Enterprise Network Management - IBM NetView/6000

need for third-party add-on products.

4.2.4 NetLabs DiMONS

NetLabs DiMONS appears to be a product which may not have much future as a platform. If recent press reports are to be believed, NetLabs is intending to concentrate their future development on their more sophisticated applications, particularly one called NerveCenter which provides a very interesting facility allowing a user to graphically build state models to diagnose network problems and to easily apply these diagnostic models to specified types of devices.

4.3 Custom and Proprietary applications

The application we will use as an example of this class of network management application is FORE System's ForeView 4.0 which is used to control their FORE-runner ASX-200BX ATM switch.

4.3.1 ForeView

ForeView is a complete network management system for FORE Systems' ATM products [12]. It can operate either as a standalone application or integrate with enterprise networking applications such as those discussed in the previous section.

Figure 4.2 shows the 'AsxView' module of the ForeView software. The graphical interface represents the current status of the ATM switch being managed. Rather than present a conceptual view of the device status, the UI shows live views of the status LEDs and ports. It is immediately apparent what the status of a port

Figure 4.2: ForeView 4.0 showing ATM switch status

on a remote switch is by simply looking at the virtual representation on screen.

In addition to viewing switch status, the application menu system allows many other management functions which can control almost any aspect of the ATM switch operation.

A common task for a network manager might be to create new virtual paths or circuits through an ATM network. Rather than configure this from the command-line interface of a device, the FORE product provides a graphical front-end where VP/VC values can be entered.

The path and channel tools make it easy to set up permanent virtual circuits (PVC) in a Fore ATM network. By pointing and clicking on the end switched in the network diagram, ForeView can automatically configure a PVC using the optimal route through the network. This eliminates the need to manually configure PVCs. This is illustrated in 4.3.

While products such as this are a major benefit to the network manager, they are essentially closed products. Proprietary interfaces may be used when communicating with the switch when remote configuration is performed.

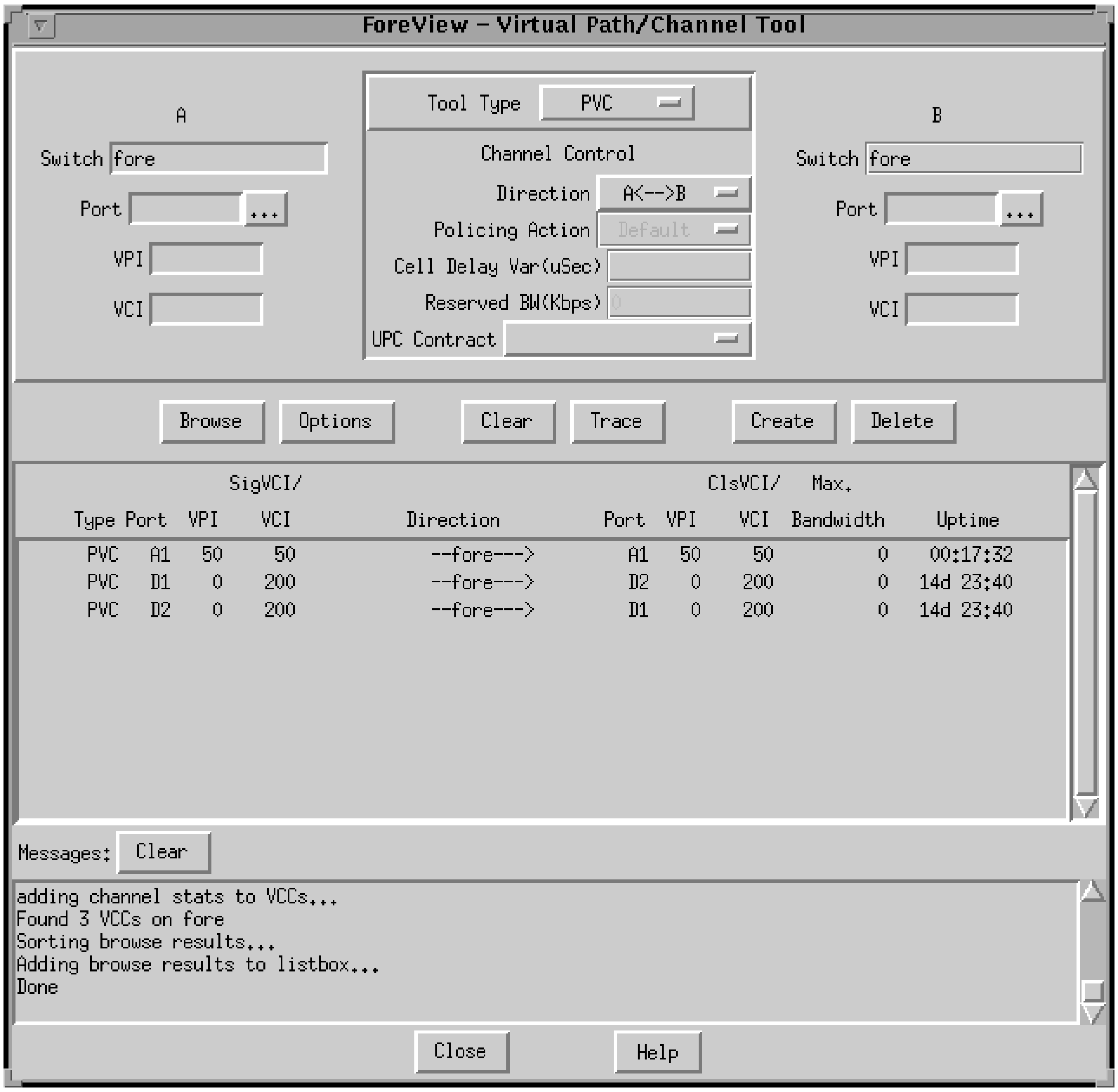

Figure 4.3: Creating virtual paths & circuits with ForeView 4.0

4.4 LANalysers and Research tools

The application we will use as an example of this class of network management application is a set of tools created by the computer science department at the Curtin University of Technology in Perth, Australia [34].

As part of their work they have created a suite of network management (or to be more precise, monitoring) tools which run on a variety of versions of the Unix operating system.

These tools are excellent examples of the tools which are becoming available on the Internet which are of very high quality and often include complete source code. Tools in this category are quite distinct from those discussed in the previous two sections. They tend to be monitoring, rather than management tools, and they are general based on open practices where possible. They are rarely used to operate on a single vendor-specific product.

The three tools developed by Curtin University that we will examine are: Etherman — dynamically monitors ethernet LAN traffic; Interman — dynamically monitors local internet traffic; Packetman — captures and analyses data packets from LAN.

The same research group have also developed a number of other similar graphical tools. 'loadman' analyses data packets collected from a LAN, and constructs a graphical representation of where traffic has been traveling, and uses combinatorial optimisation techniques to suggest ways in which the network can be segmented to minimise loading. The 'geotraceman' application is a graphical version of the popular Unix/IP tool 'traceroute'. A map of the world is shown on screen, and by consulting a database with a mapping of IP addresses to geometric coordinates, the path along which an IP packet travels can be rendered on screen.

4.4.1 Etherman

Etherman is a tool for monitoring all ethernet host connections in real-time. It can display the amount of traffic on each host pair link. The greater the amount of data being transferred between the machines, the larger the line drawn between them will be. See figure 4.4.

It is possible to scale nodes and links to allow a better picture of relative network communications, if the standard LAN view becomes too congested or is not entirely relevant.

Bandwidth consumption statistics are sampled once every second and displayed via a scrolling stripchart widget.

A textual dump of protocol summaries collected for each host and link is available. The information given is for the run-time of the program. These summaries are decoded for all ethernet frame id's and all IP protocols. The output format for each host pair indicates the amount of data exchanged in bytes and packets, and the protocol in use.

The 'loadman' tool mentioned above which can suggest how to change the topology of a heavily loaded network can take its input from the protocol data output of etherman.

Etherman can be used to detect a variety of network problems and shortcomings. Because of its graphics nature, problems such as excessive bandwidth consumption and broadcast storms become easy to identify. It is also very useful for finding unexpected transmissions or unknown devices.

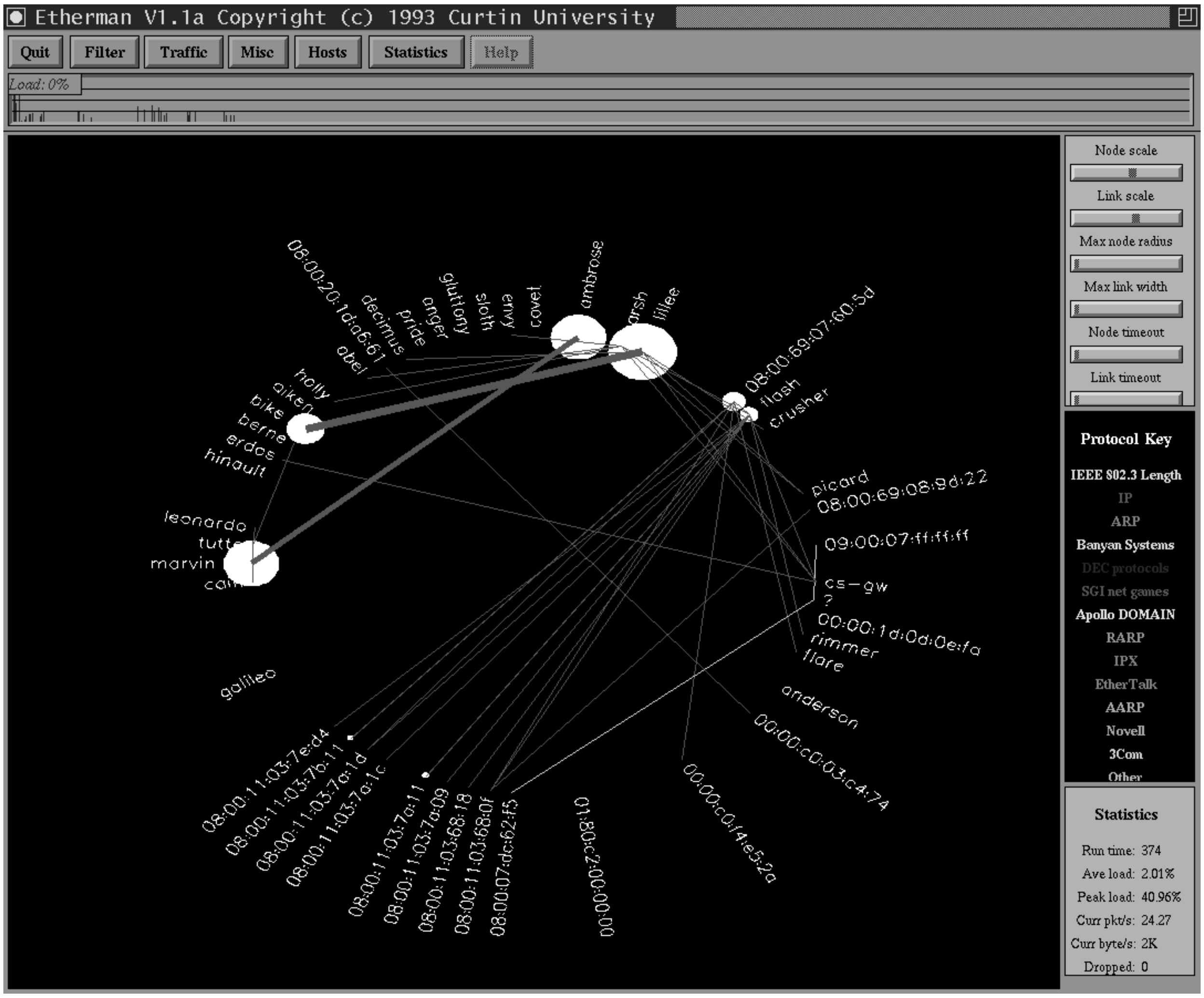

Figure 4.4: The Etherman Application

4.4.2 Interman

The Etherman application runs at a layer or level below the Interman application. Like Etherman, Interman can monitor and display traffic in real-time with update times varying because of traffic flow and monitoring station capabilities.

As can be seen from figure 4.5, different networks are shown as circles with hosts local to each network listed around their respective circumferences. Hosts are connected between two networks via a single line; the colour of the link denoting the dominant protocol. Hosts and networks appear on the diagram as communications are monitored, and disappear when a host or network has been idle for a configurable period of time.

Several operations that extend beyond the scope of the original work have been included. These include 'finger', 'telnet', 'ping', network compression, 'traceroute', a modified 'fping', basic SNMP information, and a screen refresh.

These tools become useful when it is necessary to probe for network or host information. Other miscellaneous options such as protocol summaries, filtering, postscript dumps, and varying node and link time-outs are available via menu options and scroll bar adjustments.

4.4.3 Packetman

Unlike the preceding Curtin applications, Packetman is a retrospective packet analyser. It can be used to decipher packet trains which are buffered, and optionally stored for future reference. A protocol analyser has the advantage of being able to decompose headers and examine protocol transactions in great depth. This implementation provides a good base from which to provide comprehensive protocol analysis.

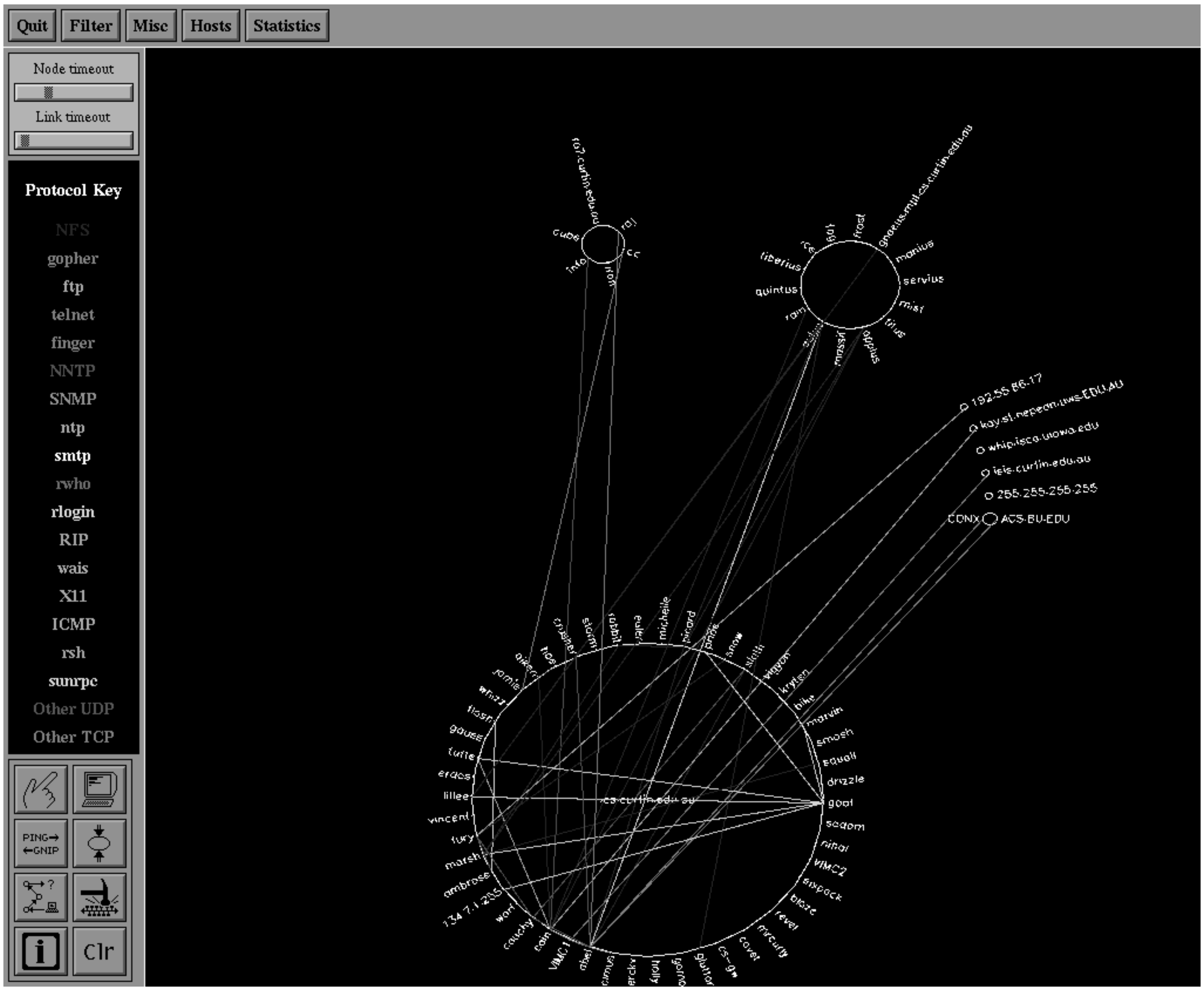

Figure 4.5: The Interman Application

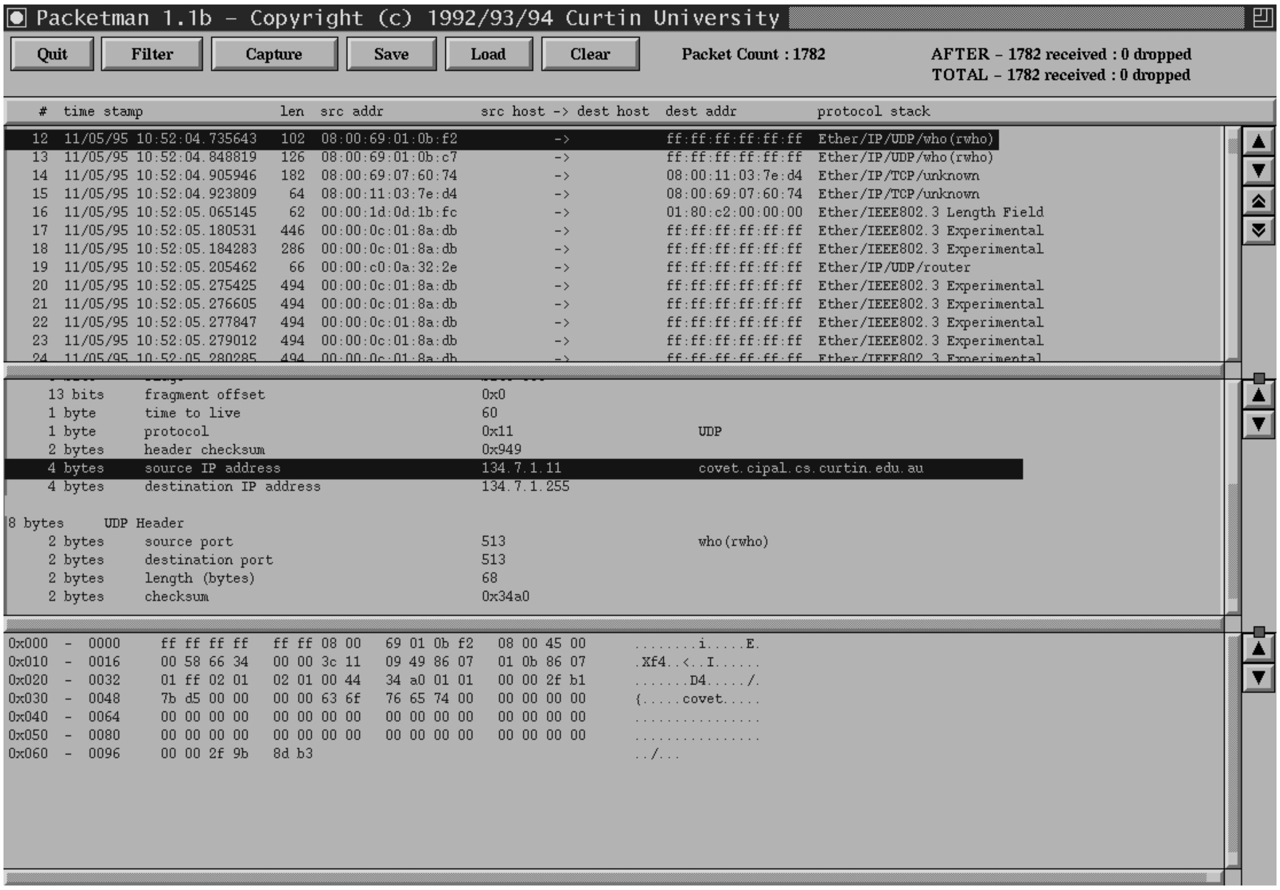

Figure 4.6: The Packetman Application

As can be seen from figure 4.6, the display is segmented into three windows each providing a different view of captured network data. The top window is a sequential trace of captured data. Note that a filter may be applied before commencing a trace to allow the user to focus on transactions of interest. Each packet has an associated sequence number, timestamp, source/destination part, and a brief overview of the protocol contained within.

A textual description of all decodable protocol fields within the selected packet is displayed in the middle window. Decomposition for Ethernet frames types, selected IP/UDP, IP/TCP, ARP and ICMP protocols is available. The lower windows gives a simple hexadecimal and ASCII dump of the entire packet.

4.5 Summary

There are a small number of enterprise-wide network management applications. Each has strengths and weaknesses, but on the whole, they all perform within narrow bounds of each other.

In general, these NMS applications lack the level of detail or granularity to probe a machine for in-depth information on its status and configuration. This, however, is more due to limitations of underlying protocol implementation such as SNMP and CMIP.

Because of this lack of detailed control in an open manner, custom applications are required to control and configure complex devices. An ATM switch is an excellent example of such a device, and the ForeView product gives a good indication of the functionality expected by the network manager.

Freely available research tools provide the network administrator with an immediate and intuitive understanding of network utilisation. They can be used to diagnose and correct a variety of faults on several networks. Such tools were not designed to replace existing network management systems, but rather to supplement them. By adding the information provided by research tools to the facilities provided by HP OpenView etc, the network manager can receive a more complete and powerful management system.

The bottom line for network managers however, is that there is just far too little integration between these classes of applications. Much more standardisation and cooperation between vendors is required. Truly open interfaces and guidelines must be laid down and adhered to. The next chapter explores this in more detail.

Chapter 5

Organisations

5.1 Introduction

As mentioned briefly before, there are several organisations who play important roles in a number of areas in the field of enterprise network management.

Standards are the key to successful enterprise network management. Unfortunately, many of the standards initiatives to date have had rather lofty goals and focussed on issues which are somewhat removed from the problem a network manager faces when a remote PC fails to operate correctly.

This chapter examines the work of one organisation who is trying to bring together vendors from all the computing and networking related fields. Rather than focus on standardising one particular protocol or interface, they are attempting to build an overall model of how components in a fully open and heterogeneous enterprise network should manage and be managed.

5.2 Background

The computer industry has been growing at a phenomenal rate. Much of this growth can be attributed to the falling cost of the desktop PC, and the ease with which new machines can be made to communicate and internetwork.

While this explosion in network computing power might offer great benefits to the end-users, the network manager is unfortunately faced with the prospect of a network of 'black boxes' which have historically been virtually unmanageable.

PC users today face systems that are difficult to configure, service, and even difficult to know what exact software and hardware resources their machines are equipped with. Basic PC management functions remain elusive to many of today's PC users.

At the corporate level, network managers have found it nearly impossible to manage PC resources, or easily configure or troubleshoot systems, especially if these systems are remotely located.

Management functionality has always been added as an 'after-thought' in the PC world. With such product diversity in the marketplace, the lack of even the most simple de-facto network management standards, and cut-throat competition between vendors, there has simply been no place whatsoever for pooling resources to build interoperable manageable products.

What was needed was a large scale collaborative initiative which could start afresh, and plan for the needs of future desktop machines and networks.

Building this enabling technology for smart, managed PCs and deploying it across the vendor community is the mission and goal of the Desktop Management Task Force – the DMTF [10].

5.3 The Desktop Management Task Force

The Desktop Management Task Force had its roots at the Interop 92 Spring trade show. The original vendors who gathered at the show to develop a cooperative strategy for managing desktop systems, Intel, Microsoft, Novell, SunSoft and SynOptics, were quickly joined by Hewlett-Packard, IBM and Digital Equipment Corporation. Since then, membership has broadened and its steering committee includes the above vendors in addition to new arrivals such as Apple, Compaq, Dell, and Novell.

A series of DMTF working committees have been established. These are responsible for areas such as Modems, Printers, Servers, etc. All of these working groups are populated by leading computer industry vendors.

The essential goal of the DMTF is to build a system whereby desktop computers, hardware and software products, and peripherals (be they standalone or networked) are allowed to be manageable and intelligent. The system allows them to communicate their system resource requirements to one another and coexist in a manageable PC system.

For the vendors, this technology adds ease of use, intelligence, management and interoperability to hardware, software and peripheral products that implement it. For users, the DMTF solution offers a number of benefits: systems and software that are simple and easy to use, plug-and-play installation, and real-time systems diagnostics and support.

In their own words, the DMTF claim their solution will provide a “flexible and sophisticated way to manage systems and products through an inherently open PC landscape” resulting in “an invigorated industry and a seedbed for the next generation of computing products”.

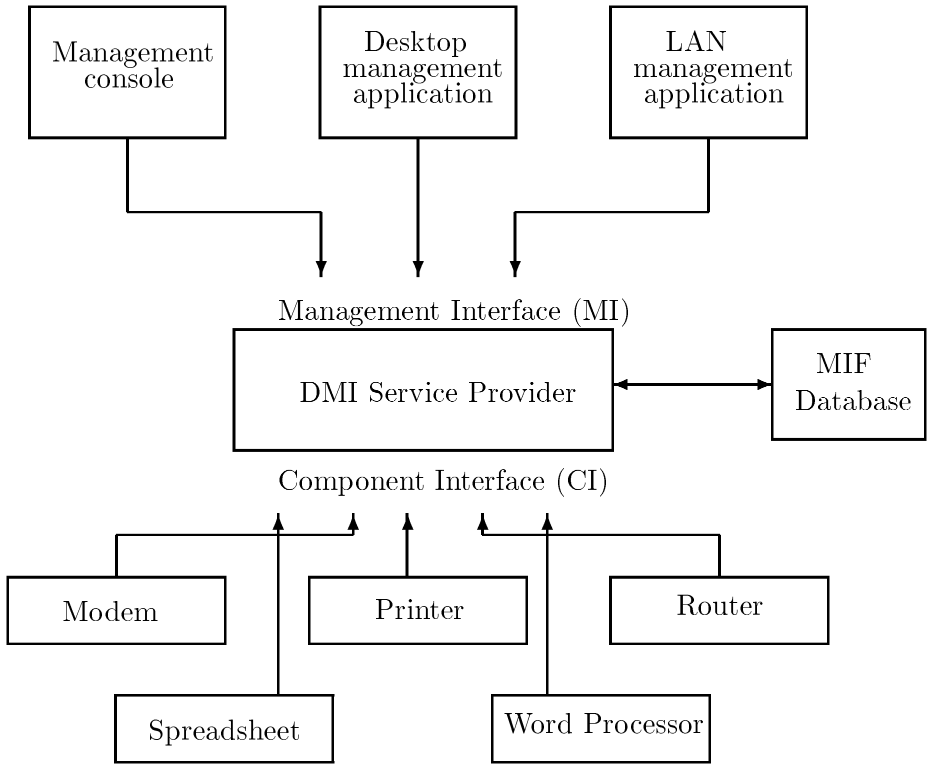

Figure 5.1: The DMI Architecture

5.3.1 DMTF architecture

The DMTF have created the Desktop Management Interface (DMI). This architecture is shown in figure 5.1.

The main component of the architecture is the DMI Service Provider. This is a local program which collects information from products, manages this information in the Management Information Format (MIF) database, and passes the information to management applications as requested. It controls communication between itself and management applications by means of the Management Interface and between itself and manageable products by means of the Component Interface.

Management applications are remote or local programs for changing, interrogating, controlling, tracing and listing the elements of a desktop system. A management applications can be a local diagnostics or installation program, or an agent which redirects information from the DMI over a network (see SNMP later in chapter).

Manageable products are hardware, software or peripherals that occupy or are attached to a desktop computer or network server. Each product provides information to the MIF database by means of a MIF file.

DMI Interfaces & Components

The two main interfaces to the DMI service provider are at the application and network device level. They are the Management Interface (MI) and Component Interface (CI):

- The MI allows DMI-enabled applications to access, manage and control desktop systems, components and peripherals. It shields management applications from the different mechanisms used to obtain information from products within a desktop system.
- The CI allows components to be seen and managed by numerous applications that call the DMI. It also shields components vendors from decisions about management applications, protocols and operating systems, and allows them to focus on providing competitive management for their products.

The Service Provider handles Get, Set, List and miscellaneous commands from management applications, retrieving the requested information from the MIF database or passing the requests on to manageable products as needed.

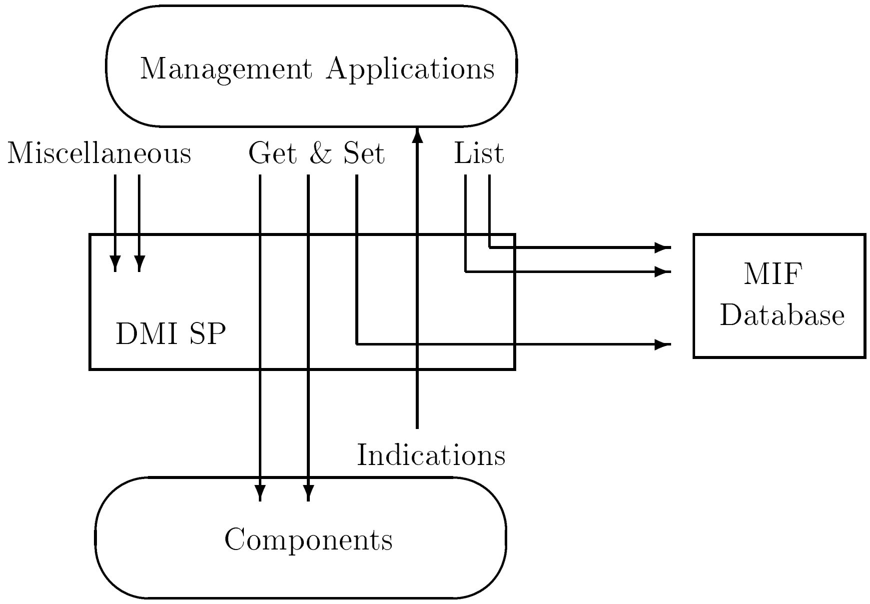

Figure 5.2: DMI Information Flow

It handles indications from manageable products and passes that information on to the management applications. This flow of information is show in figure 5.2.

The MI commands provide three types of operations to control manageable products: Get, Set and List. There are also commands for installation and registration.

Get allows a management application to get the current value of individual attributes or groups of attributes. Set allows writable attributes to be changed.

List commands allow management applications to read the MIF descriptions of manageable products, without having to retrieve the attribute values for that product. Thus, a management application can query a system and retrieve useful information about the contents of the systems, with no previous knowledge of that system.

The Component Interface handles communications between manageable products and the service provider. It communicates with manageable products for Get and

Set operations. It receives indications from manageable products and passes those to the MI.

The CI and MIF file shield vendors from the complexity of encoding styles and management registration information.

MIF files

The MIF database contains information about the products on the system. MIFs define the standard manageable attributes of PC products in categories including PC systems, servers, printers, LAN adapters, modems and software applications. The DMI's Service Provider manages the information in the MIF database, derived from MIF files provided with each manageable product.

The MIF file has a defined grammar and syntax. It is a simple ASCII text file describing a product's manageable attributes, grouped in ways that make sense.

When a manageable product is initially installed on a system, the information in its MIF file is added to the MIF database and is then available to the DMI Service Provider, and thus to management applications.

The simplest MIF file contains only the Component ID group, but MIFs can become as complex as needed for any given product. The value for each MIF attribute can either be found in the MIF database or provided by program code (called 'instrumentation') that runs when the value is requested.

Ideally, MIF values are provided dynamically by instrumentation; this avoids the problem of keeping the values up to date as products change.

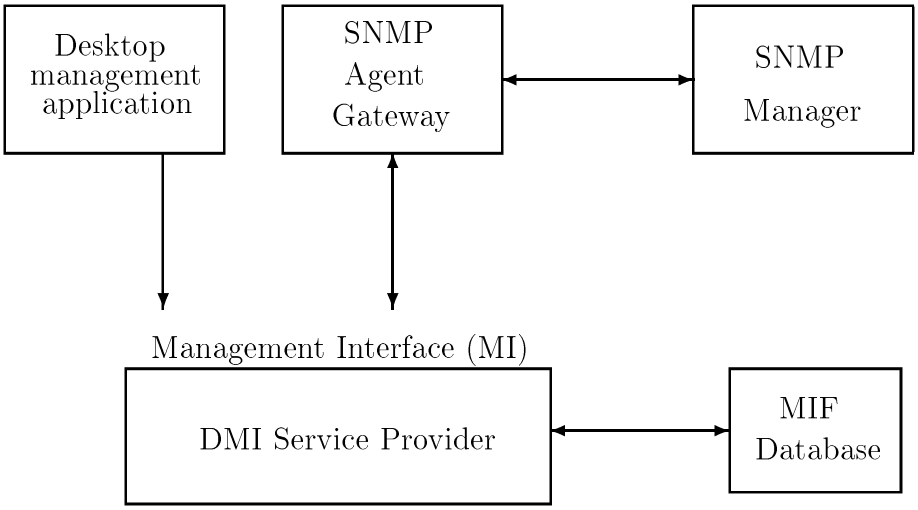

Figure 5.3: Integrating DMI and SNMP

5.3.2 SNMP & DMTF

The DMTF work is complementary to existing network management standards such as SNMP which already has a large installed base. DMI maps to SNMP-based consoles to access the desktop in a uniform manner. The DMI and SNMP can work together to provide the network manager with extensive LAN statistics and information.

By building an SNMP 'gateway' agent as a DMI management application, remote SNMP managers can gain full access to the DMTF MIF database. This is illustrated in figure 5.3.

Work is ongoing in the Internet Engineering Task Force to build an SNMP to DMI Mapping standard [5]. As noted above, this will enable systems instrumented to the DMI to be remotely, and uniformly, managed using SNMP. There are two facets to this work: enabling new SNMP applications to be written to manage DMI instrumented systems, and to enable DMI instrumented systems to support standard SNMP MIBs.

5.4 Summary

The DMI has gained strong support from numerous vendors and corporate users who support the DMTF's continuing effort of proliferating the DMI standard in all computing environments. The DMI promises an open standard, which will guarantee vendors a common strategy for designing supportability and manageability into their products. For users, the DMI promises ease-of-use and inherent intelligence in their computing and networking environments.

In previous chapters we have seen the increasing complexity of local and wide area networks, and we have seen how traditional network management protocols such as SNMP and CMIP are having difficulty in alleviating a network manager's problems.

By addressing these issues from an objective viewpoint, the DMTF has outlined the 'glue' which is needed to bring together vendors from all different areas of networking products, both software and hardware.

This type of high-level approach is often adopted by standards bodies aiming for perfection, and is often either not implemented at all or made superfluous by de-facto industry standards. Fortunately for the enterprise network manager community, the DMTF is not a traditional standards body, but an industry driven force, and hence its lofty goals are well within reach.

Chapter 6

Object Technology

6.1 Introduction

Distributed computing has made significant advances in the past few years, with the release of both the OSF's Distributed Computing Environment and the Object Management Group's OMA/CORBA specifications.

This chapter describes the OMG's Common Object Request Broker Architecture and examines how it could be used in the domain of enterprise network management.

Although this technology is still relatively young, it holds great promise for how future enterprise network management systems might be built. As discussed in previous chapters, management protocols and applications are not progressing at the same rate as the underlying networks.

While the Desktop Management Task Force are bringing together vendors, and providing a framework and specification within which they can cooperate, they do not rule out the adoption of new technologies in order to progress their overall

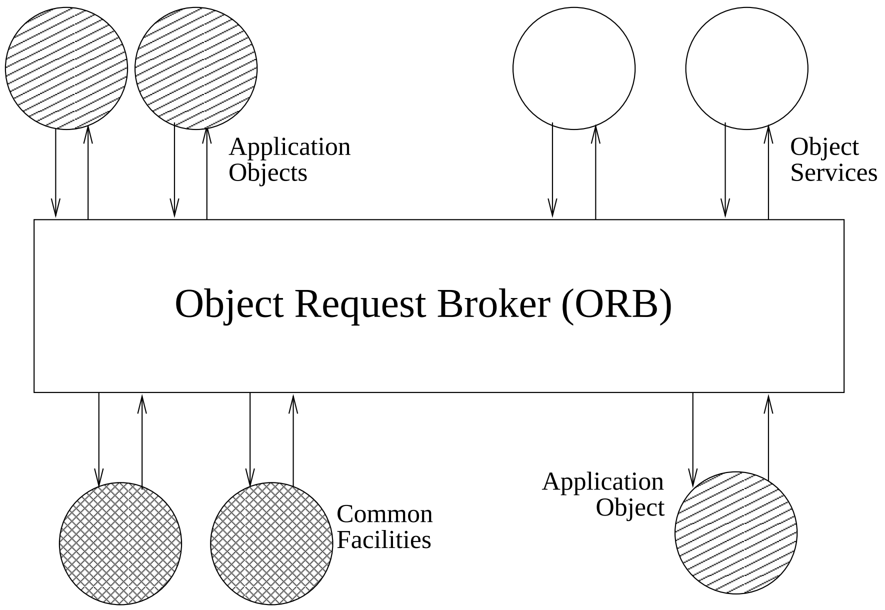

Figure 6.1: The Object Management Architecture

desktop management goals.

For some time, the software industry has been moving towards object-oriented techniques, and it is rare that new projects are started today which do not incorporate object technology in one form or another.

Given the inherent distributed nature of enterprise networks, it makes sense then to explore how techniques learned from distributed object computing can be applied to enterprise network management.

6.2 Object Management Architecture

Figure 6.1 shows the main components of the Object Management Architecture. The Object Request Broker or ORB is the central component: it provides all the other components with the ability to communicate.

Ideally, an ORB should make distribution completely transparent to an object which is communicating with other, potentially remote, objects. For example, it should not be apparent to an object where a remote object is located or what operating system or programming language has created that object.

Changes of characteristics in one object should normally not force the recompilation of other objects. This allows changes to be made dynamically to the implementation of an object without affecting other objects. Thus it is easy to introduce new implementations and replace existing services.

There are many possible ways to realise and implement such an ORB, but the OMG have standardised how they believe this should be done and this is the CORBA specification, which is discussed in the next section.

6.3 The Common Object Request Broker

This section discusses the OMG's CORBA specification [29]. The important features which are covered include: the object model, the interface definition language (IDL), the orb itself, IDL stubs and skeletons, the dynamic invocation interface, the interface repository, and the basic object adapter. See figure 6.2.

6.3.1 The Object Model

An object is an entity which encapsulates state and provides one or more operations on that state. Operations can be requested by clients and are described in the IDL language (see below). Specifications include the parameters required, result expected, exceptions, and operation execution semantics [11].

To invoke an operation, clients must identify the receiving object. Object ref-

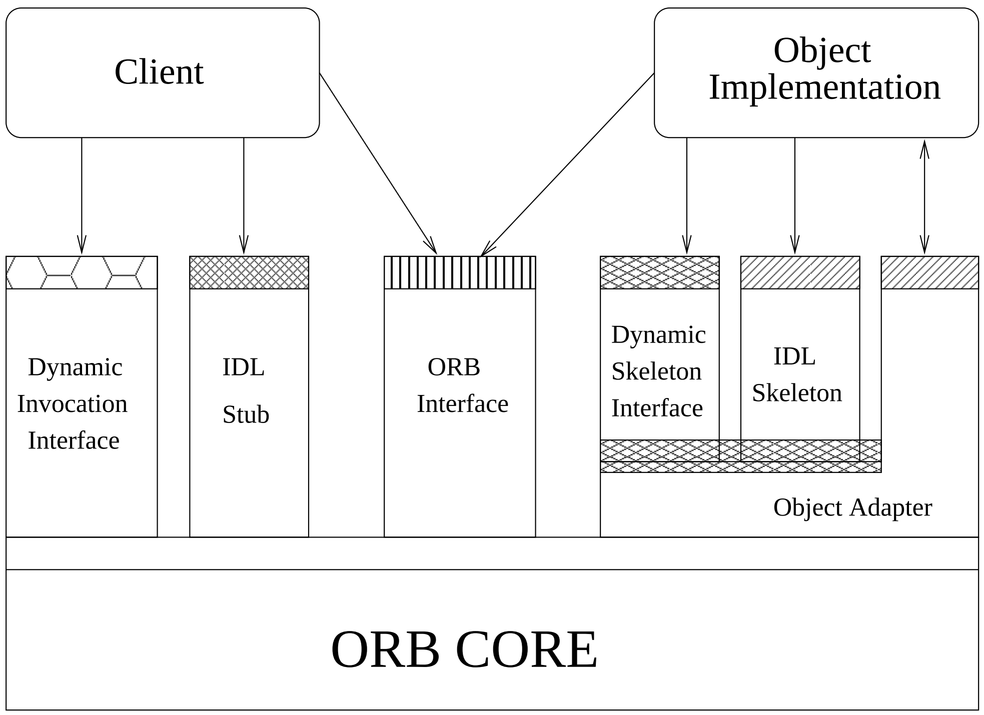

Figure 6.2: The Common Object Request Broker Architecture

erences are used for this. An object reference always identifies the same object. However, the same object may be identified by more than one object reference.

When an object receives an invocation from a client, it may invoke operations on other objects before returning a result to the client. The client has no way of telling if other objects were invoked.

An interface is a set of possible operations which a client can request of an object. Interfaces are specified in IDL. An object may support multiple interfaces. Given an object reference, a client may invoke any operation within any interface supported by the object.

Objects can be created and destroyed as the result of operations; the mechanisms for this are transparent to the client. The result it perceived by the client as an object reference which identifies the new object.

6.3.2 Interface Definition Language

The Interface Definition Language is used to specify the components of the ORB as well as services which objects make available to clients. IDL uses a C++ like syntax for defining the operations supported in an interface. It completely specifies the parameters, results and exceptions for each operation, but does not specify the semantics of that operation. This means that two implementations may satisfy the same IDL specification.

A language mapping defines how the CORBA IDL types map onto the type system of a target language. In addition, the mapping also specifies how certain ORB interfaces must appear to programmers using the language. A number of language mappings have been defined including: C, C++, Smalltalk, Ada, Cobol, and recently, Java.

6.3.3 Object Request Broker

The primary function of an ORB is to enable a client to invoke an operation on a potentially remote object. A client identifies the target object with an object reference.

The ORB is responsible for locating the object, preparing it to receive the request, and passing the data need for the request to the object.

Once the object has executed the operation identified by the request, if there is a reply needed, the ORB is responsible for communicating the reply back to the client.

6.3.4 IDL Stubs and Skeletons

The job of the stub and skeleton is to hide the details of the underlying ORB from the developer, making remote invocation look similar to local invocation. Stubs and skeletons are automatically generated from an IDL interface definition. A 'stub compiler' can take IDL and convert it into the native language which will be used for compilation e.g. C++, Java, etc.

6.3.5 Dynamic Invocation Interface

The DII provides clients with an alternative to using IDL stubs when invoking an object. The DII allows an invocation to be constructed manually, without static (compile-time) knowledge of the object being invoked. The dynamic API allows for decisions to be made much later than the static approach. This is an attractive feature for many applications. However, the DII API is considerably more complex than the static approach [40].

6.3.6 Interface Repository

The Interface Repository provides an alternative to IDL for specifying interfaces to objects. Applications building and invoking requests via the DII can obtain information at runtime from the repository.

In other words, with an interface repository, a client can locate an object unknown at compile time, enquire about its interface, and then use the DII to build a request to be forwarded through the ORB.

6.3.7 Basic Object Adapter

The object adapter or BOA provides the run-time environment for instantiating server objects, passing requests to them, and assigning object references to them. The BOA is the means by which object implementations access most ORB services, including generation and interpretation of object references, method invocation, activation of an object, mapping references to implementations, etc.

Just as interface definitions are stored in the Interface Repository, the BOA expects implementations to be stored in an Implementation Repository. A typical repository might store compiled programs or scripts to start and initialise objects.

6.4 Applying CORBA

In this section we will see how CORBA object technology is being used in industry today by public telecom operators and private commercial ventures.

Close parallels can be drawn between how these projects harness CORBA to provide seamless communication facilities between multiple distributed nodes and how enterprise networks could benefit by using CORBA as a means of sharing and accessing management information in a structured format.

6.4.1 Telecommunications

The initial use of CORBA has been as a distributed platform for Telco Operations Systems, enabling the integration and greater re-use of different management software components, encapsulated within well defined IDL interfaces [22].

CORBA is also being applied to inter-system communication. In this scenario, CORBA communication replaces the CMIP protocol stack. Instead, GDMO-based management information models (managed objects) are mapped to IDL and CMIS messages carried by the ORB messaging protocols. A joint task force between the OMG and NM Forum is, at present, addressing the mapping of GDMO to IDL and run-time mediation between CORBA and TMN/OSI [2].

This case in particular is very close to the enterprise networking model. Instead of a group of telephone switches exchanging TMN/GDMO information via CMIP, we have a distributed network of routers and other devices exchanging simple SNMP control messages.

In light of SNMP's recent problems, and CMIP's complexities, a viable alternative might be a CORBA based system. Network device characteristics would be modelled in objects, and lightweight orbs would run on the devices. These orbs would communicate in 'CORBA-space' with a central network management station. Since operations on remote objects are transparent using CORBA, emphasis could be placed on building management functionality into the system rather than worrying about the type of protocol which is being used (i.e. SNMP vs CMIP).

6.4.2 The Iridium Project

The Iridium system is one of the most ambitious telecommunications projects ever undertaken. When online, it will provide the first wireless communications network that spans the world.

Motorola are creating Iridium using CORBA as the software with which they will build and control the ground station segment for the Iridium Global Cellular Network program.

CORBA technology is being used to build the terrestrial software application to control sixty-six Iridium satellites. Three ground stations will contain all the intelligence for the network, from satellite navigation control to telecommunications switching.

Motorola say they chose CORBA as it allows their programmers to develop distributed, object-oriented applications quickly, yet follow a consistent and straight forward standard-based model. The implementation of CORBA they are using, Iona's Orbix, meets these needs and also claims to be lightweight in operation.

The Iridium system will provide global wireless hand-held telecommunications services including voice, facsimile, data and paging via 66 lightweight satellites in low-earth orbit.

The satellites will be placed in six orbital planes consisting of 11 satellites each, positioned approximately 420 nautical miles above the earth. The satellites will be interconnected through cross-links to provide complete coverage of the earth's surface.

The Iridium system will allow subscribers to use pocket-sized hand-held wireless telephones to communicate with virtually any other telephone in the world. Communications will be relayed via satellite and through the ground-based gateways, placed in key regions world-wide, that will interconnect with public telephone networks.

In areas where conventional terrestrial cellular service is available and compatible with Iridium dual-mode telephones, the subscribers' communications will transmit over the cellular network.

6.5 Conclusion

It should be apparent from the above, that the CORBA approach is highly scalable. As orb products mature, and provide extra functionality, yet remain lightweight in terms of performance overhead, it will be more and more appealing for network equipment vendors to consider how CORBA can be used to build an enterprise network management system.

In the Iridium system, end-user telephone handsets, satellites, base-stations, and regular telephone exchanges are all being modelled as objects. In a corporate network, there is no reason why printers, modems, routers and end-user PCs cannot be modelled in a similar fashion.

Most recently, orb vendors have introduced Java language bindings for their CORBA implementations. This means that applets written in the Java language, can now communicate with any CORBA compliant server, provided that an interface has been defined in IDL.

The next chapter discusses the Java language in more detail, and shows how it could be used to re-engineer how manager-agent network management takes place. If Java can be integrated into network equipment, then the argument for introducing CORBA to enterprise network management can only be strengthened.

Chapter 7

The Java Solution

7.1 Introduction

Intelligent agents, distributed computing, middleware, object request brokers, component software…

These are the buzzwords of today's computing and networking industries. They are all hot topics and are all under intense scrutiny in research labs to investigate how new products can be built or competitive advantages can be gained [19].

In this chapter we will look at a technology which appears to have something in common with all of the above: Sun Microsystems' new programming language, *Java* [14]. In Sun's own words:

> *"A simple, object-oriented, network-savvy, interpreted, robust, secure, architecture neutral, portable, high-performance, multithreaded, dynamic language."*

7.2 The Java language

By examining each of the attributes in the quote above in more detail, a clearer picture of the Java programming language emerges.

Simple — modern programming languages are getting larger, more feature-rich, and hence more complex and difficult to use. Java is small and simple to use. A self-contained Java microkernel can be built in less than 220 kilobytes.

Object-oriented — Take C++ and all the OO features it provides, apply the Java simplicity philosophy and remove some rarely used and complicated features, and you're left with a small OO language.

Network-savvy — Java provides libraries with TCP code for handling commonly used protocols such as HTTP and FTP. Java applications can open and access objects across the net via URLs with the same ease that programmers are used to when accessing a local file system.

Interpreted — Java source files are 'compiled' into *bytecodes*. To support Java, a virtual Java machine is run which can interpret these bytecodes. This approach helps Java in its architecture neutrality and portability.

Robust — Much effort has gone into early checking for potential problems and avoiding situations which are error prone. One example of this is Java's pointer model. Java has true arrays – there is no more pointer arithmetic. Programmers have less opportunity to corrupt memory, leading to increased application robustness.

Secure — Essential in a networked environment, several levels of protection in the Java virtual machine combined with public key encryption techniques helps Java's security and robustness.

Architecture neutral — To build a system which can be used to develop code which runs on multiple heterogeneous networked systems, a basic requirement is that it is architecture neutral.

Portable — Java code will run on any machine with a port of the Java virtual machine. The machine runs applets by interpreting the bytecodes created by the Java compiler, regardless of where the code was compiled.

High-performance — Despite the interpreted nature of Java, Sun have built the bytecode system so as to gain as high a performance run-time system as possible. Already, there are hardware Java chips available which run bytecodes natively.

Multithreaded — Support for threads has been built into Java to increase interactive responsiveness and real-time behaviour. In a world where many different events are happening simultaneously, no longer does the programmer have to link in third party threads packages to handle multiple simultaneous tasks.

Dynamic — Java is a dynamic language. It was designed to adapt to an evolving environment. By using a concept known as interfaces, and by binding software component objects together as late as possible, Java avoids a number of software reusability problems evident with languages such as C++, and makes the use of the OO paradigm much more of a reality.

7.3 A Java network device

In this section we will examine how Java could be used to aid in the management of enterprise networks by building a new breed of network device that makes use of Java technology. The principles introduced here could be applied to almost any existing network device such as a router, ethernet hub or ATM switch.

7.3.1 Current scenario

Some of today's enterprise network management tools were examined in Chapter 4. These offer many excellent features of use to the network manager. However, the basic technologies behind these applications are relatively dated when considering the speed with which advances are being made in the networks that these applications are attempting to manage.

Management of devices takes place by using simple protocols which are essentially static in nature. A manager application usually polls an agent on a device using a well known protocol and asks some basic questions.

The next section shows how the addition of Java technology to this scenario can offer considerable improvements.

7.3.2 Java solution

A complete Java-centric solution to this scenario can be achieved by implementing Java technology at many levels, both in software and hardware.

Figure 7.1 shows one such solution. On the manager side, a basic personal computer of any kind can be used. No particular operating system or hardware is preferred. The only requirement is that a Java-capable web browser is available.

On the agent side, we will consider three areas where the introduction of Java will have a significant impact:

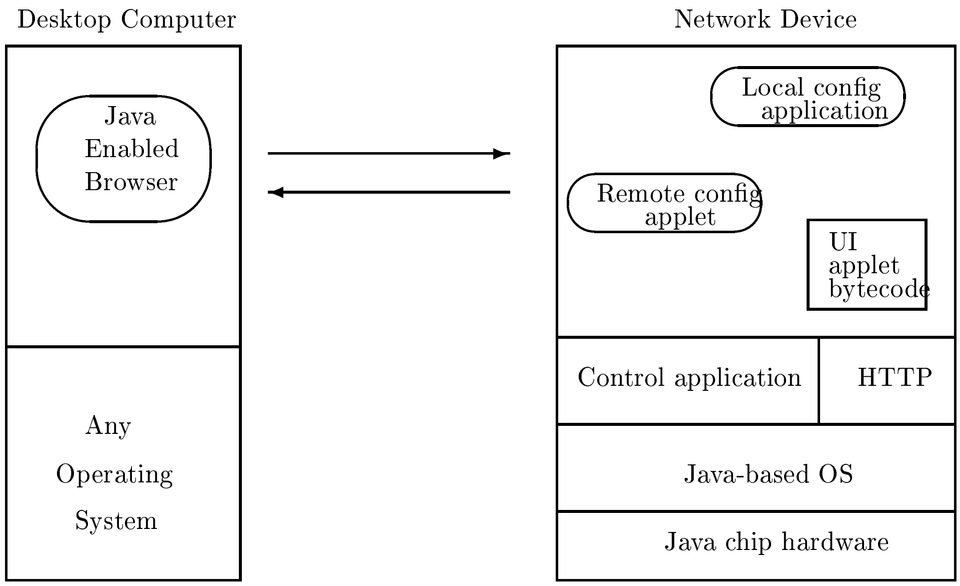

Figure 7.1: Java Network Device Architecture

- Java hardware for the network device
- Java based OS for the network device
- A networked object-oriented Java paradigm for the communications protocol in use between the network device and the manager

At a hardware level, Sun CPU chips which execute Java code natively can be used to build the network device. A conventional Motorola or Intel solution is no longer needed. The Java chips are available in a number of configurations to suit the size, speed or power requirements of the device in question. Where previously, it might be considered excessive to place management functionality in a particularly small device because of the expense and overhead of a suitable CPU, a low-end Java chip can be used.

Because the hardware has changed, the programming of the device must also change. In this case, because of native execution, assembly language or C can be replaced with Java and still run at close to the same speed. In addition to this, because Java is scalable, no matter how small or how large the network device is, the same Java code will execute on whichever Java chip is chosen for the device. Code developed for high-end network devices can easily be reused in low-end models.

Above the hardware and OS is the "control application" or functionality of the network device. This is complemented by HTTP (HyperText Transfer Protocol) code which allows the device to communicate with world wide web browsers. The top half of the agent in figure 7.1 contains three components. These function as follows:

- Local configuration application — this is local front-end configuration application which will be used by the network manager if a local terminal is attached to the device. It can be as simple as any conventional command-line interface.
- Remote configuration applet — this code runs on the network device and handles communications with a remote web browser or equivalent when the devices is being remotely managed by a custom applet and possibly a custom protocol.
- User Interface applet bytecode — this code is the user interface which will run on the management platform through a world wide web browser with Java capability. The applet bytecode is transferred to the web browser using HTTP and then communicates with the remote configuration applet described above.

These three components are naturally written in Java, and compiled into bytecodes. Because of the built-in HTTP functionality, each of these modules can be replaced at any time with new versions through simple software upgrades across the network.

This is a good example of the dynamic nature of the Java language impacting on how software is distributed and deployed. The other dynamic aspect of Java is how it handles communications protocols.

Dynamic Java

Not only has the chipset and programming language changed, but the whole methodology or paradigm for how network management takes place between manager and agent can now be re-evaluated. The dynamic nature of the Java language and how it handles new protocols can be used to great advantage in enterprise network management.

Figure 7.2 illustrates how building a network device with Java can lead to a much more advanced management system than is possible with conventional management protocols.

The network manager first makes contact with the network device by launching a web browser. By opening a normal web HTTP connection to the device, the device can act as a simple web server and return an opening screen via HTML (HyperText Markup Language) which may or may not contain Java applets.

At this very basic level, the network manager is able to monitor the status of the device by simply browsing web pages. These web pages which are returned to the browser by the device can all be generated on-the-fly and contain up-to-date information each time they are requested. This browsing though HTML capability comes through the ease with which HTML/HTTP functionality can be implemented using Java.

The next step up from this simple browsing is the inclusion of Java applets within the HTML pages. This is the "UI applet bytecode" shown in figure 7.1 and the

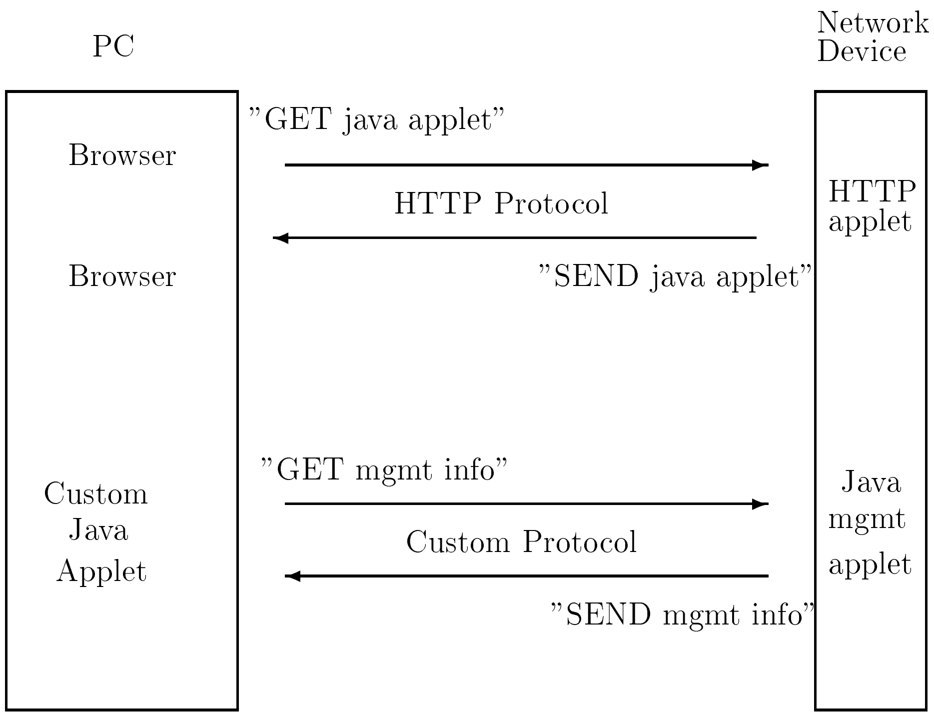

Figure 7.2: Dynamic protocols using Java

"java applet" in figure 7.2. This applet is the front-end application to the network device that the network manager can execute locally via the web browser.

Because of the ease with which Java can adapt to new protocols, the front-end Java applet need not be restricted to using any conventional network management or world wide web communications protocols. New custom protocols can be designed for manager-agent communications in this scenario.

7.4 Integrating Java

So where does this leave the conventional enterprise networking applications? How can Java be used to improve and add functionality to the already comprehensive set of management tools in widespread deployment?

Tools such as HP OpenView and SunNet Manager can easily be improved by introducing a Java Virtual Machine which can run applets in the same way as Java capable web browsers.

Traditional SNMP and CMIP protocols can be used to query a device and ascertain whether or not it has Java functionality. If it does, the network management application can spawn a subtask which presents the user with a custom interface and indeed even custom protocol which can communicate directly with the device, bypassing any limitations of SNMP or CMIP.

Already vendors are exploring these possibilities. The next section looks at some of the early initiatives which are taking place.

7.5 Java Initiatives

There has been a frenzy of industry activity since the Java language was released and its full potential made clear. Below are some of the early adopters of this technology and their efforts in establishing it in the network management marketplace.

Advent Network Management — have created Java class libraries for SNMP which developers can use to create cross-platform alternatives to monolithic SNMP applications [25, 1].

Advent hopes that developers will use the software to build SNMP applications such as Management Information Base (MIB) applets for Web browsers.

These SNMP applets will run in any Java-enabled browser and allow managers to perform the same tasks that are commonly the domain of more costly tools from vendors such as Hewlett-Packard and IBM.

Sahara Networks — plan to use Java applets and Web servers to manage large-scale broadband ATM networks. The company plans to combine its own JavaView Java-based agent technology with SNMP and file-transfer technology in its SaharaView management architecture [24, 15].

Each device will include an embedded Internet server consisting of a suite of Internet protocols to SNMP, file management and JavaView agents. SaharaView will integrate with other management platforms and report the status of network devices in realtime via Java applets.

UB Networks — plan to offer Java applets on their GeoLAN 500 hubs [35]. New products will roll out in three phases; the first release will be a Java-based real-time graphical application that will let network managers monitor device status, both remotely with Web browsers and onsite with traditional management platforms. Next, UB will embed Java applets into its existing EMPower management module which resides within the hub. Finally, UB will provide users with a software developer's kit to create custom applets to run on EMPower.

IBM — have created a new version of their NetFinity LAN management software called PC SystemView. This new release will give users the option of managing networks from a WWW browser and allow the management component to run on Windows NT servers [26].

IBM claims to be the first company to give users the option of using a Web browser to access data and initiate management tasks. The software will gather hardware or software data and package it according to the DMTF's MIF format, but it also available in MIB format for SNMP managers.

PC SystemView is Web enabled with a software stack that maps network data and management action capabilities to HTML and Java, so that any

Java enabled Web browser will support a remote PC SystemView management session.

7.6 Java Management Toolkits

As a language, Java provides a simple and elegant means of developing object-oriented code. In addition, it has great support for distributed application development.

Network management application development can benefit from this, as well as allow users to manage their networks from anywhere. Network managers no longer have to feel constrained by their systems as long as they have access to a web browser.

This section overviews four Java toolkits of particular relevance to enterprise network management developers:

- The Advent Java SNMP Package
- The JavaManagement API
- JaSCA - Java SNMP Control Agent
- IBM's Webbin CMIP

7.6.1 Advent Java SNMP Package

The Advent Java SNMP package is a group of Java class files that provides Java programmers with an API for developing network management applets and applications that use SNMPv1 [1]. Many different architectures are supported, such that SNMP can be used in the web browser, on the server, or even on managed elements that have a Java Virtual Machine.

Java applets that are loaded from the network are usually restricted by browsers from connecting to any system other than the applet host. The Advent package provides special support for applets loaded remotely via the network that run in browsers that have socket and file access restrictions. Via a Java program called the SNMP Applet Server (SAS) on the applet host (web server), applets can communicate with the SNMP managed devices and save and retrieve files on the applet host.

The implementation is based on Carnegie Mellon University SNMP 1.2U, with a new MIB parser that handles multiple MIB module files that can be dynamically loaded in any Java application or applet.

7.6.2 The JavaManagement API

In order to promote Java in as many areas as possible, Sun Microsystem's daughter company JavaSoft have developed the JavaManagement API [20]. The API is a set of extensible objects and methods which can be used in the development of system, network and service management solutions for heterogeneous networks.

The JavaManagement API provides the user with:

JavaMagement API User Interface Style Guide — a document which aids the developer in creating applications that conform to UI guidelines.

Admin View Module — the AVM is an extension of the Java Abstract Window Toolkit (AWT). AVM offers a set of UI components to build richer UIs, providing components such as tables, hierarchy browsers, charts and graphs.

Base Object Interfaces — these support the construction of objects that represent distributed resources. They allow the developer to define abstractions that contain distributed attributes and methods.

Managed Container Interfaces — these allow applications to perform actions on a single group of objects instead of each instance. This permits applications to scale upwards by allowing for multiple instances of managed objects to be treated as just one.

Managed Notification Interfaces — the basis from which more complex event management services can be built. The model provides asynchronous event notification between managed objects or management applications.

Managed Data Interfaces — support mapping classes and instances of Base Objects to a relational database. They support a number of commercially available RDB engines.

Managed Protocol Interfaces — provide the infrastructure to perform distributed operations securely.

SNMP Interfaces — extend the Managed Protocol Interfaces to allow extension of Base Objects to contain information obtained from existing SNMP agents.

Applet Integration Interfaces — allow developers to integrate Java applets into the JavaManagement API.

By utilising the JavaManagement API, developers can create consistent applications with a decreased learning curve and reduced development cycle.

7.6.3 JaSCA - Java SNMP Control Agent

Designed by Pekka Nikander and Petri Wessman at the Helsinki University of Technology, JaSCA is an extensible applet set and class library designed to integrate Java with SNMP [28].

JaSCA consists of a number of Java classes and packages, implementing e.g. generic, extensible ASN.1 encoding/decoding functionality, generic SNMP machinery including automatic poller, and a number of classes designed to create online animation.

A number of demonstration applets have been made available:

- JaSCA with an imaginary demo network
- JaSCA with a single real SNMP data source
- JaSCA with a more complex example using real SNMP data

The JaSCA package contains all the necessary software to allow a standard Java compatible web browser to monitor the state of SNMP attributes at the host where the applet is loaded from. By installing portions of JaSCA software at the local host, monitoring of other SNMP compatible hosts and networks devices is also possible.

7.6.4 IBM Webbin' CMIP

Webbin' CMIP is a research project aimed at simplifying the access to network management information and resources. The web browser has been used as a user interface for Webbin' CMIP, and a prototype was shown in operation at the Third WWW Conference in Darmstadt in April 1995 [9].

Webbin' CMIP includes two applications:

- GDMO/ASN.1 Search Engine — enables users to navigate OSI management documents and browse General Descriptions of Managed Objects. It is based on enhanced versions of IBM's GDMO and ASN.1 compilers, and is able to generate HTML code in addition to data structures used for BER encoding/decoding.
- Liaison — establishes a bridge between the web and the world of network management. It contains facilities for: automatic network resource discovery, error handling, simple ASN.1 representation, and access to metadata information.

External Java bindings are available which make it possible to access the services provided by Liaison from external applications or applets. This allows developers to write network management applications and Java applets in a few lines of code with a single and simple object model for both CMIP and SNMP. Since the protocol used for these bindings is just HTTP, management applets can communicate with peers and with Liaison even if they are behind a firewall.

7.7 Summary

Despite its youth, hardware vendors are already coming forward with plans to bring Java-based products to market.

The technologies discussed in the introduction to this chapter may play a large part in the future of network management. It is no coincidence that Java has an integral part to play in each of these technologies. Either they rely on Java for some of their own internal functionality, or they make sure that an external Java interface is available to communicate with the rest of the world.

Of all the emerging 'hot' technologies, Java is the most concrete in terms of implementation. Others make great promises of what the future holds when these technologies are embraced, but Java can deliver clear advantages today in how we build new software products.

On a more sobering note, it is important to remember that after all the hype, Java is just a programming language. It's how it is packaged and used that is important. Java itself won't change the enterprise networking world, but it can play a major role as a catalyst.

Chapter 8

Conclusion

In this final chapter, the report concludes with a summary of what has been discussed, the issues that need addressing in the industry, and an outlook on how the future might unfold.

8.1 Introduction

Enterprise network management covers many areas. It is almost impossible to bring all these areas together in one volume. Problems which arise for a network manager can happen at many levels. If one piece of equipment in a network of thousands of peripherals fails, troubleshooting the problem can be a low-level technician task or high-level network manager task.

Network managers' roles can vary greatly from company to company. This report has attempted to highlight the diversity of skills and expertise required in trying to manage enterprise networks. This was done by overviewing a wide cross-section of relevant topics in the enterprise networking domain.

8.2 Report summary

The report discussed:

1. Networks: today's LAN and WAN technologies.
2. Protocols: standardisation processes, SNMP, RMON and CMIP.
3. Applications: the most common types in use.
4. Organisations: the Desktop Management Task Force.
5. Object technology: the CORBA approach.
6. Java Solution: Sun Microsystem's Java language.

8.3 Issues

Having discussed these topics, it is apparent that there are a number of issues which need to be addressed if successful network management can take place in the future.

- Network standardisation — is something that can never be fully accomplished, but more work is needed in this area. Market forces and industry competition is fragmenting the marketplace in new networking solutions such as Fast Ethernet and ATM. Early adopters of these networks are finding themselves using proprietary implementations.
- Improved protocols — are needed if the conventional enterprise management applications are to proliferate. CMIP is obviously not the answer anymore, and the SNMP debacle has left the network manager with no choice but to use an outdated network management protocol.

- Application integration — is something which must be achieved at a much tighter level. Current applications fail to communicate with each other, and perform specific tasks which might solve one problem or give one representation of a network, but lack any information transfer or sharing capabilities with other comparable applications.
- Industry cooperation — above all is needed. As always, the great thing about standards is that there are so many of them to choose from. The DMTF is most definitely a step in the right direction, but more cooperation is needed across the board. There is always a place for competition. Standards don't impact on this, they simply set down some ground rules.

If these issues can be tackled then enterprise network management techniques as they stand today could be improved to place the network manager in a considerably better position than at present.

8.4 The future

If the 1980's was the decade of the LAN, then the 90's must be the decade of the WAN. The growth of the Internet, if nothing else, is testament to this.

For years, Sun Microsystems have used the slogan 'The network is the computer' when promoting their products. Initially, it was interpreted as nothing more than a simple marketing catchphrase, but in recent years it has become a reality.

The computing and communications fields have been converging for some time, and distributed computing is now an everyday fact of life. Key to the distribution of computing resources is software object technology. Everything can be treated as an object, be it a simple numeric calculation, or a complex sequence of instructions comprising a program task or thread. If software systems are designed in an object oriented manner, then technologies such as the Common Object Request Broker Architecture can be used to allow these objects to be distributed transparently over computer networks.

The Java programming language serves to reinforce this idea. It is inherently object oriented and comes ready to run with network libraries, and can be used to implement CORBA objects. With low-cost Java hardware chips already in production, it would appear to lend itself naturally to building and monitoring network components.

8.5 Conclusion

Managing large scale corporate networks today is not an easy task. Despite much effort from standard bodies and industry co-operatives, there is no clear candidate for an all-encompassing enterprise network management platform in sight.

This is due in no small part to the rate at which network technology has improved and been deployed in the last five to ten years. With this trend continuing it is difficult to see how today's initiatives will be able to keep pace and adapt to the massively complex terrabit networks of the years ahead.

At present, the great hope for network management has to be object technology. If objects become ubiquitous and network devices can be accurately modelled as objects with appropriate characteristics, then the lessons learned from the software industry in managing objects en-mass will hopefully be applicable to enterprise networking.

Bibliography

[1] ADVENT NETWORK MANAGEMENT, INC. (1996). Advent Network Management JAVA SNMP Package. *http://www.adventnet.com/*

[2] ASHFORD C. (1993). Comparison of the OMG and ISO/CCITT Object Models. *Report of joint NM Forum/OMG taskforce on Object Modelling*

[3] BEECHER, B. et al. (1994). Network Management Systems Product Review. *http://smurfland.cit.buffalo.edu/NetMan/ProdRevs/UnivMichReport.html*

[4] BLACK, U. (1992). Network Management Standards, The OSI, SNMP and CMOL Protocols. *McGraw-Hill*

[5] BOSTOCK, S. (1995). DMTF SNMP to DMI Mapping Standard. *IETF RFC-950713 Informational*

[6] CASE, J., FEDOR, M., SCHOFFSTALL, M., DAVIN, J. (1990). A Simple Network Management Protocol. *IETF RFC-1157*

[7] CASE, J., McCLOGHRIE, K., ROSE, M., WALDBUSSER, S. (1993). Introduction to version of the Internet-standard Network Management Framework. *IETF RFC-1441*

[8] DERI, Luca. (1996). Network Management for the 90s. *http://www.zurich.ibm.com/ lde/SimpleNM.html*

[9] DERI, Luca. (1996). Surfin' Network Resources Across the Web. *http://www.zurich.ibm.com/ lde/Surfin.html*

[10] DMTF. (1995). The Desktop Management Task Force Enabling the Next Generation of Intelligent and Manageable PCs. *http://www.dmtf.org/*

[11] EDWARDS N. (1995). CORBA - An industrial approach to Open Distributed Computing. *Draft for inclusion in Open Distributed Systems*

[12] FORE SYSTEMS. (1994). ForeThought and the ATM Internetwork. *http://www.fore.com/*

[13] FRONTIER SOFTWARE DEVELOPMENT. (1995). The Integration of RMON and SNMP Technologies for Managing Enterprise Networks. *http://www.frontier.com/whitepap/wp1/*

[14] GOSLING, J., McGILTON, H. (1996). The Java Language Environment. *http://java.sun.com/doc/language/environment/*

[15] GREEN, T. (1996). Sahara puts Java to management test. *Network World, Feb 12 1996 v13 n7 p1*

[16] HAUGDAHL, J. S. (1986). Inside the Token Ring. *Architecture Technology Corporation*

[17] HERMAN, J. (1996). The sorry state of enterprise management. *Distributed Computing Monitor, March 1996 v11 n3 p3*

[18] HOU, Y. T., TASSIULAS, L., CHAO, H.J. (1996). Overview of Implementing ATM-Based Enterprise Local Area Network for Desktop Multimedia Computing. *IEEE Communications, April 1996*

[19] HUNTINGTON-LEE, J. (1996). Finding solutions with staying power. *INTERNETWORK, Jan 1996 v7 n1 p11*

[20] JAVASOFT. (1996). The JavaManagement Application Programmer Interface. *http://java.sun.com/products/JavaManagement*

[21] JEACLE, K. (1995). ISDN – The Integrated Services Digital Network. *Just Amiga Monthly, Aug 1995, p28*

[22] KINANE B., SLEVIN M., ANDERSON G., DONNELLY W. (1995). Object technology and its impact on the telecommunications industry. *Communicate: The Broadcom Technical Journal*

[23] KOOIJMAN, R., VAN OORSCHOT, J. P. M., WISSE, D. M. (1994). RMON Implementation Issues for Managers and Agents. *http://dnpap.et.tudelft.nl/DNPAP/Research/Research.html*

[24] LACH, E. (1996). Sahara's hot new management plan. *Communications Week, Feb 19 1996 n597 p68*

[25] LEON, M. (1996). Advent creates Java tools for SNMP applications. *InfoWorld, March 25, 1996 v18 n13 p8*

[26] LEON, M. (1996). IBM LAN management app gives users control from Web. *InfoWorld, Feb 19 1996 v18 n8 p48*

[27] LUNDFALL, K. (1992). Frame Relay - for faster and more efficient data communications. *Ericsson Review, No. 1-2, 1992*

[28] NIKANDER, P., WESSMAN, P. (1996). Java SNMP Control Applet. *http://termiitti.akumiitti.fi/nixu/*

[29] OMG. (1995). The Common Object Request Broker: Architecture and Specification, Revision 2.0. *Object Management Group*

[30] PFEILER, R., RITTSTEIGER, G. (1991). High-speed communications the pulse of the economy. Deutsche Bundespost TELEKOM puts the MAN on

trial in Munich and Stuttgart. *Telcom Report International, July/Aug 1991 v14 n4 p38*

[31] ROSE, M. T. (1990). The Open Book, a practical perspective on OSI. *Prentice-Hall*

[32] ROSE, M. T. (1994). The Simple Book. *Prentice-Hall*

[33] SCHATT, S. (1991). Linking LANs: A Micro Manager's Guide. *Windcrest / McGraw-Hill*

[34] SCHULZE M., BENKO S., FARRELL C. (1993). Homebrew Network Monitoring: A Prelude to Network Management. *http://www.cs.curtin.edu.au/*

[35] SNELL, M. (1996). Java wakes up network management. *LAN Times, March 4 1996 v13 n5 p18*

[36] STALLINGS, W. (1993). SNMP, SNMPv2, and CMIP. *Addison-Wesley*

[37] STALLINGS, W. (1992). ISDN and Broadband ISDN. *MacMillan Publishing Company*

[38] TANENBAUM, A. S. (1981). Computer Networks. *Prentice-Hall*

[39] THOMAS-CONRAD CORP. (1995). High-Speed Networking White Paper. *Thomas-Conrad Corporation*

[40] YANG Y., DUDDY K. (1995). Distributed Object Computing with CORBA. *University of Queensland, DSTC Technical Report*