DEFENCE R&D DÉFENSE

Distributed Network Management

David Kidston
Communications Research Centre Canada

Communications Research Centre
Centre de recherches sur les communications

The work described in this document was sponsored by the Department of National Defence under Work Unit 5CB12

Defence R&D Canada
DEFENCE RESEARCH ESTABLISHMENT OTTAWA

TECHNICAL MEMORANDUM
DREO TM 2000-109
CRC Report No. 2000-010
October 2000

National Defence Défense nationale

Canada

DEFENCE R&D DÉFENSE

Distributed Network Management

David Kidston
Radio Network Systems
Network Technologies

Communications Research Centre
Centre de recherches sur les communications

The work described in this document was sponsored by the Department of National Defence under Work Unit 5CB12

DEFENCE RESEARCH ESTABLISHMENT OTTAWA

TECHNICAL MEMORANDUM
DREO TM 2000-109
CRC Report No. 2000-010
October 2000

Abstract

This report begins with an introduction of network management and an outline of the problems facing the existing management systems for enterprise networks. Commercial and defence related issues in effective management of the next generation of application traffic are also discussed.

The following four main requirements for effective network management are extracted; management must be scalable, adaptable, useful, and collaborative. Scalability is required to handle the increasingly large sizes of contemporary networks. Also, network management systems should be able to adapt to the diversity of network components. New utility at all levels of network management is necessary to handle the traffic requirements of the new kinds of applications. For example, applications may require security and/or be QoS sensitive. Finally, for networks to co-exist and provide end-to-end management while maintaining local control, some kind of collaboration is needed.

It is postulated that a distributed management architecture is best suited to satisfy these requirements. Several network management architectures are reviewed. The report gives an overview of existing management related technologies. The report concludes with a comparison of these technologies, a coalition management proposal, and a discussion of several related issues.

Résumé

Le rapport que voici commence par une introduction à la gestion des réseaux et un exposé des problèmes que posent les systèmes de gestion actuels des réseaux d'entreprise. Il y est aussi question des difficultés associées à une gestion efficace du trafic que suscitera la prochaine génération d'applications dans les secteurs commercial et militaire.

Pour être efficace, la gestion d'un réseau doit reposer sur quatre grands éléments : l'échelonnabilité, l'adaptabilité, la fonctionnalité et la collaboration. Face à la taille grandissante des réseaux contemporains, le système de gestion doit être échelonnable. Il devrait aussi pouvoir s'adapter aux diverses composantes du réseau. Le système de gestion doit être fonctionnel à tous les paliers du réseau si l'on veut répondre aux exigences des nouvelles applications en matière de trafic. Certaines applications, par exemple, nécessiteront des mesures de sécurité et/ou seront sensibles à la qualité du service. Finalement, une forme de collaboration est essentielle pour que les réseaux coexistent et qu'on puisse gérer le tout sans perdre la maîtrise du réseau local.

Une architecture de gestion répartie satisferait sans doute le mieux les contraintes qui précèdent. Le rapport examine plusieurs types d'architecture. On y trouvera une vue d'ensemble des technologies existantes associées à la gestion des réseaux. Le rapport se termine par une comparaison des technologies en question, une proposition pour la gestion en coalition et une analyse de plusieurs points connexes.

Executive Summary

With the increasing size and complexity of modern computer networks, it is becoming obvious that current network management systems are incapable of making the required transition. The current centralised systems were designed for simple networks with a small number of devices located in close physical proximity. A more distributed approach is required for the global enterprise networks of today.

Distributed network management provides a possible solution to a variety of problems related to large heterogeneous networks. Due to its distributed nature, manager-agent communication can be localised in a portion of the complete network. This minimises enterprise wide congestion from management traffic, which in turn reduces lost packets and retransmissions throughout the network.

Distribution can also help when dealing with heterogeneity of devices. Since devices of similar configuration are likely to be physically and logically close together in the network, specialisation can also be grouped. This allows the network system to isolate the global variability into its local managers that individually must deal with much less complexity.

There are many issues for network management in both a commercial and military setting. One of the main findings of this report are four requirements network management systems should satisfy. Such systems should be scalable, adaptable (able to handle different types of devices and networking environments), utile (have the necessary functionality to be useable/useful) and collaborative (able to share information and co-ordinate management with external network service providers).

This report discusses three different architectures related to distributed network management systems, and some enabling technologies for each. Peer-to-peer NM systems can be based on existing management architectures. Such systems distribute their utility while maintaining existing protocols as much as possible through the use of multiple intermediate managers that communicate between each other to co-ordinate management between their various sub-domains. The IETF DISMAN working group is extending information models so that management tasks can be accomplished on remote sites. The DMTF is extending web protocols and languages to extend them into the management realm with WBEM.

Distributed object NM systems make use of object-based technology in an attempt to objectify network devices and management. By modelling devices as objects with internal state and methods, object request brokers can spread management services throughout the network. An advantage of this scheme is that it can be combined with existing protocols making it possible to implement this method now, as proposed by the JIDM. Under this framework, CORBA provides the distributed middleware on top of

which services can be written in a language independent manner. Access to network devices is accomplished by the use of a CORBA-SNMP or CORBA-CMIP gateway.

Dispersed NM systems are based on AI techniques for the dissolution of management tasks to semi-autonomous pieces of code. These agents communicate amongst themselves and interact with devices though special agent environments. Though promising, few agent standards exist. Several competing agent languages do exist, showing that the technology is maturing.

A distributed object based network management system based on the JIDM specification is proposed. This system uses a five-layer architecture using SNMP to contact network devices, CORBA middleware to provide distributed functionality, distributed management services that span the network, and on top of which a user interface can operate from any location. An initial prototype has been developed which can access SNMP-based device status. Other services that may be developed in this framework include auto-discovery of new devices, auto-configuration, and probabilistic auto-management.

Finally, several additional issues are dealt with in the final section of the report. The advantages of agent based notifications over server directed polling is investigated. The problems related to multi-service provider SLAs (Service Level Agreements) are touched upon, as are the advantages of agent-based management systems. The problems of vendor-specific MIBs and the alternate application-level network management model are also investigated.

DREO TM 2000-109, CRC-RP-2000-10, *Distributed Network Management,* David Kidston, CRC, VPNT-RNS, Ottawa

Sommaire

La taille et la complexité des réseaux informatiques ne cessant de croître, il est de plus en plus évident que les systèmes de gestion actuels ne pourront effectuer la transition. Les systèmes centralisés existants ont été conçus pour des réseaux simples, constitués d'un petit nombre d'appareils situés à proximité les uns des autres. Les réseaux des entreprises internationales qu'on connaît aujourd'hui nécessitent des systèmes de gestion répartis.

La gestion de réseau répartie pourrait résoudre divers problèmes associés à l'exploitation des grands réseaux hétérogènes. À cause de la nature répartie du système, la communication entre gestionnaire et agent peut s'effectuer à tel ou tel endroit du réseau. De cette façon, les risques que le trafic congestionne l'entreprise entière sont réduits au minimum, ce qui, par voie de conséquence, diminue le nombre de paquets perdus et de retransmissions.

La répartition a également son utilité quand les appareils sont hétérogènes. Puisque les appareils configurés de la même façon sont physiquement et logiquement regroupés dans le réseau, on peut aussi procéder à un regroupement par spécialité. Le système d'exploitation cantonnera la variabilité de l'ensemble au niveau des gestionnaires locaux, qui seront aux prises avec des problèmes beaucoup moins complexes.

La gestion d'un réseau soulève maintes difficultés dans un milieu commercial ou militaire. Une des principales constatations du présent rapport est qu'un système de gestion devrait satisfaire quatre critères : l'échelonnabilité, l'adaptabilité (il devrait gérer des situations et des appareils différents), la fonctionnalité (son utilité dépend des fonctions qu'il autorise) et la collaboration (il devrait partager l'information et coordonner la gestion avec les fournisseurs de services extérieurs).

Le présent rapport examine trois architectures réparties de systèmes de gestion de réseau et quelques technologies habilitantes pour chacune d'elles. Les systèmes de gestion entre homologues s'appuient sur les architectures existantes. Pareils systèmes répartissent les fonctions du réseau en préservant le plus possible les protocoles en usage par le truchement de nombreux gestionnaires intermédiaires qui communiquent entre eux afin de coordonner la gestion des sous-domaines. Le groupe de travail DISMAN de l'IETF (groupe de travail sur les technologies Internet) tente d'élargir les modèles d'information pour que les tâches de gestion puissent s'effectuer à distance. Le DMTF (groupe de travail sur la gestion répartie) perfectionne les protocoles et les langages Web pour les adapter à la gestion d'entreprise par le Web (WBEM).

Les systèmes de gestion de réseau répartis à objets font appel à la technologie des objets, l'idée étant de transformer appareils et fonctions de gestion du réseau en objets. En modélisant les appareils comme des objets, à l'état et aux procédés finis, les systèmes de courtage des demandes d'objet parviennent à répartir les services de gestion sur la totalité du réseau. Une telle solution présente l'avantage de pouvoir être combinée aux protocoles existants, ce qui en permet l'adoption immédiate, comme le propose le JIDM (groupe pour la gestion conjointe des domaines). Avec une architecture de ce genre, CORBA servirait d'intergiciel de répartition au-dessus duquel se trouveraient des services rédigés dans un autre langage. On accéderait aux appareils du réseau grâce à un portail CORBA-SNMB ou CORBA-CMIP.

Les systèmes de gestion de réseau dispersés recourent aux techniques d'intelligence artificielle pour diviser les tâches de gestion en codes semi-autonomes plus petits. Les agents communiquent entre eux et interagissent avec les appareils dans des environnements spéciaux. Quoique cette solution semble prometteuse, peu de normes s'appliquent aux agents. Plusieurs langages concurrents coexistent, signe que la technologie n'est pas encore parvenue à maturité.

On préconise un système de gestion de réseau réparti à objets épousant la spécification du JIDM. Ce système repose sur une architecture à cinq niveaux et se sert d'un protocole de gestion de réseau simple (SNMP) pour assurer la communication entre les appareils du réseau, de l'intergiciel CORBA pour maintenir la fonctionnalité après répartition et de services de gestion répartis sur l'ensemble du réseau avec, au sommet, une interface-utilisateur capable de fonctionner n'importe où. On a mis au point un premier prototype qui accède aux appareils SNMP. D'autres services pourraient être développés, notamment la recherche automatique de nouveaux appareils, la configuration automatique et une gestion probabiliste automatique.

La dernière partie du rapport aborde plusieurs autres questions. On y compare les avantages des avis en mode agent à ceux des interrogations par le serveur. On survole les problèmes posés par les fournisseurs multiservices qui ont conclu un accord sur les niveaux de service et les avantages des systèmes de gestion en mode agent. Le rapport se termine par un examen des difficultés que soulèvent les bases d'information de gestion propres au distributeur et d'un autre modèle de gestion de réseau au niveau des applications.

DREO TM 2000-109, CRC-RP-2000-10, *La gestion de réseau répartie (en anglais),* David Kidston, CRC, VPNT-RNS, Ottawa

TABLE OF CONTENTS

1. INTRODUCTION

Since the earliest days of computer networking, there has been a need to diagnose problems with the underlying infrastructure. From the hardware that supports the propagation of bits, up the OSI stack to application data integrity, there is a wide range of possible causes for a “network fault”. It is the role of the Network Management System (NMS, or just NM) to identify and help solve computer network problems as they arise.

Networks today typically consist of a large number of devices from a variety of vendors used together to service a single organisational entity. These networks can stretch over large distances, and support a wide variety of services. They often interconnect through a shared internetworking infrastructure that cannot be managed by a single network manager. Such networks are often called Enterprise Networks.

In the military environment coalition deployments are becoming more common. When the forces of several nations are deployed for group operations, communications between the various parties are vitally important. Creating a common wide area computer network should be a priority in modern warfare. However, nationalist interests suggest that each participant will want to retain control over their own communication infrastructure. Management systems are needed that can provide such federated control.

Increases in network size, proliferation of heterogeneous components, and the growing popularity of applications that are sensitive to the Quality of Service (QoS) available from the network have led to problems for contemporary management systems. Increases in network size have begun to over-tax existing management solutions due to the volume of traffic generated and concentrated by centralised architectures. Heterogeneity of network equipment also taxes centralised managers which must be able to distinguish and understand each devices management syntax. Real-time traffic is becoming predominant, and the utilities for managing individual real-time streams do not yet exist.

Existing technologies do not provide the scalability necessary to develop enterprise-wide management systems. The common centralised control architecture is swamped by large processing and traffic overhead. In this case, a single manager must contact a large number of possibly different types of devices directly. Distributed system techniques hold promise for dealing with increases in network size and complexity. By processing management information “close” to sub-network elements, inter-network management traffic can be minimised while heterogeneity is more simply accommodated. A more distributed architecture can also ease the implementation of more complex management utility, including federated control of coalition networks.

The heterogeneity of enterprise networks makes it difficult for a single management system to interface with all network devices. The Simple Network Management Protocol (SNMP) [1, 2] provides the *de facto* standardised management interface to network devices. Unfortunately, manufacturers create device specific Management Information Bases (MIBs) to describe the operation of their particular device. This makes it difficult to interpret the device's condition without the manufacturer's specially designed management tool. Further standardisation of a common and expressive device-management interface is required to handle this heterogeneity.

The basic functions of network management (fault, configuration, accounting, performance and security management) are well known from the OSI management model [3]. Existing management systems have focused on fault and accounting capabilities. Networks are now being used for more sensitive applications that require additional management capabilities. The new types of traffic are often more performance and security sensitive. Witness the interest in voice-over-IP, for example. Systems now need the utility to deal with issues of security, performance, and advanced configuration, the three additional roles set out in the OSI management model.

The deregulation of Internet provisioning has led to a variety of Service Providers (SPs) which provide near-global connectivity through their co-operative efforts. This has increased the frequency at which enterprise networks are connected through second-party equipment. Service Level Agreements (SLAs) promise connectivity of a specified bandwidth and/or availability, but connections cannot be managed dynamically or on a per-stream basis. Similarly, in the military environment, multi-national coalition deployments that include a shared computer networking infrastructure are becoming more common. The network infrastructure could be controlled piecemeal by the national owners, but end-to-end management of traffic is preferable. The lack of targeted and global control for all network assets has led to proposals for policy-based network management technologies. Policies can be used to limit control and/or visibility, can be enforced in real-time, and can involve bi- or even multi-lateral management agreements.

Considering these many complexities, it is imperative that integrated management technologies for large multi-owner heterogeneous computer networks be developed. Distributed technologies provide one potential solution. By distributing management utility throughout the network, device heterogeneity is more easily accommodated, scalability is more easily achieved, specialised services can be easily adopted, and ISP management policy negotiation is easily accommodated. Increased usage of computing devices for communication from different locations (desktop, laptop, palmtop, pager) means that management utility should likewise be widely dispersed. It is important to ensure connectivity between a wide variety of components with different capabilities and functions. The title given to this class of solutions is Distributed Network Management (DNM).

There are a number of possible realisations of the DNM concept. Individual devices could run management agents that are contacted from a set of peer managers using special manager-agent and manager-manager messaging and control architectures. Network elements could be modelled as abstract objects that are queried and controlled by remote invocation. Another proposal is for a system of wandering management agents that move from device to device, performing management tasks as they go. This report reviews these three solutions under the titles peer-to-peer, distributed-object, and agent-based management, respectively.

This report begins with a discussion of the history of network management. Section 2 provides a brief historical perspective of commercial network management, and a description of military coalition network management as it occurred in the Joint Warrior Interoperability Demonstration of last year (JWID99-R). Section 3 continues with a summary of issues important to commercial and military network management. Section 5 continues on this theme with a taxonomy of network management architectures, from the simplest centralised solution to the most complex distributed one. Section 6 describes the network management and accompanying technologies that are available today, as well as some under developed. The report concludes with a comparison of these technologies, an overview of the research project concluding here at CRC, and a few additional issues that should be addressed by a report of this scope.

2. BACKGROUND

The current trend in network management has been to view all connected communication equipment as part of a single enterprise-wide network, no matter how diverse or distant the networking devices. This has the advantage that problems in one part of the network can be correlated with their root causes in another part. Its disadvantage is that network management systems must deal with potentially very large and very heterogeneous networks.

Current management protocols were designed in the distant past (in computing terms) for small homogeneous networks. They were temporary *ad-hoc* management protocols implemented as a stop gap measure until something better could be designed. They have since become the *de-facto* solution. These protocols were based on point-to-point connectionless communication making them simple to design and implement. One central management station was sufficient for all your management needs. It is now believed that more distributed solutions are required for the more complex network environments of today.

Networks that support coalition deployments must also deal with the limited interoperability and scalability typical of COTS management systems. Coalition exercises during the most recent Joint Warrior Interoperability Demonstration (JWID99-R) provide a reference for the technical requirements of coalition network management. The Distributed Network Management (DNM) paradigm provides a potential solution by distributing the monitoring and control utility throughout the network to provide improved flexibility, scalability, utility, and federated control.

The following sections present some of the troubled history of network management in general, and the AUSCANNZUKUS maritime network management contribution to the JWID99-R Coalition Wide Area Network in particular. The lessons of these histories can be better understood as context for the drive towards more sustainable network management technologies.

2.1 History of Network Management

The lack of a common management framework amongst network devices was not a problem for network administrators when the first data networks were created. In the early ARPANET, if a connection seemed abnormally slow, a 'ping' of the affected system and the related routers allowed the administrator to determine the location of the problem. This required in-depth knowledge of the whole network and the possible interactions that might occur. Since networks were small enough, and managers were knowledgeable enough, simple tools were sufficient to manage these networks.

Networks then grew to a size where even those who knew every aspect of the network problems that could arise could not administer each individual node separately. In order to deal with this a temporary management protocol, the Simple Network Management Protocol (SNMP) [1, 2], was introduced to support managers while the richer OSI communication protocols were standardised and implemented. This suite of protocols was expected to include enough network management utility to replace SNMP.

This temporary measure has since become the *de facto* standard deployed in almost all network devices. The swift growth of the ARPANET, and later the Internet, left little time or incentive to implement the late and complicated OSI standards. As TCP/IP [4] came into prominence so did its associated management protocol, SNMP.

The client-server nature of SNMP has led most NM systems to collect data from a single management location in order to centralise data and processing. This approach provides the advantage of simplicity in design. Localising all management tasks in a single location means that algorithms to detect and correct network faults remain simple to develop and implement. Over the years this simplicity has led to the near exclusivity of centralised SNMP-based management architectures in existing COTS network management solutions.

The unexpected explosive growth of the Internet and the reliance on computer networks for routine communications has strained existing management systems. The increases in network size mean that a larger number of SNMP messages travel longer distances thereby adding to the increased traffic. Also, the increased market for network devices has meant an increase in the types of devices and quirks a management application must be able to deal with. The legacy of stateless management agents has left us with centralised management systems that are ideal for small and simple networks, but do not scale to the large, heterogeneous, enterprise-wide networks now found in government and industry.

2.2 Network Management in JWID99-R, A Defense Case Study

The Joint Warrior Interoperability Demonstration (JWID) is a yearly demonstration of current and emerging command and control technologies. JWID is a United States exercise with the participation of invited alliance partners. It provides a forum for the nations to expose military personnel to leading edge technologies and for researchers to receive early feedback on their designs. The AUSCANNZUKUS Maritime participation in JWID `99-Revised (JWID99-R) included the provision of a Coalition Wide Area Network (CWAN) which linked real and simulated ships, shore stations and marine forces into a Multi-national Naval Task Group (MNTG) [5].

The CWAN provided connections for the US, UK, Canadian (CA), and New Zealand (NZ) participants to a naval Task Group Area Network (TGAN), as well as connections to a Multinational Marine Force (MMF) network (Figure 1). The CWAN supported several applications including messaging, distributed collaborative planning, tactical operations picture, and some simple web services.

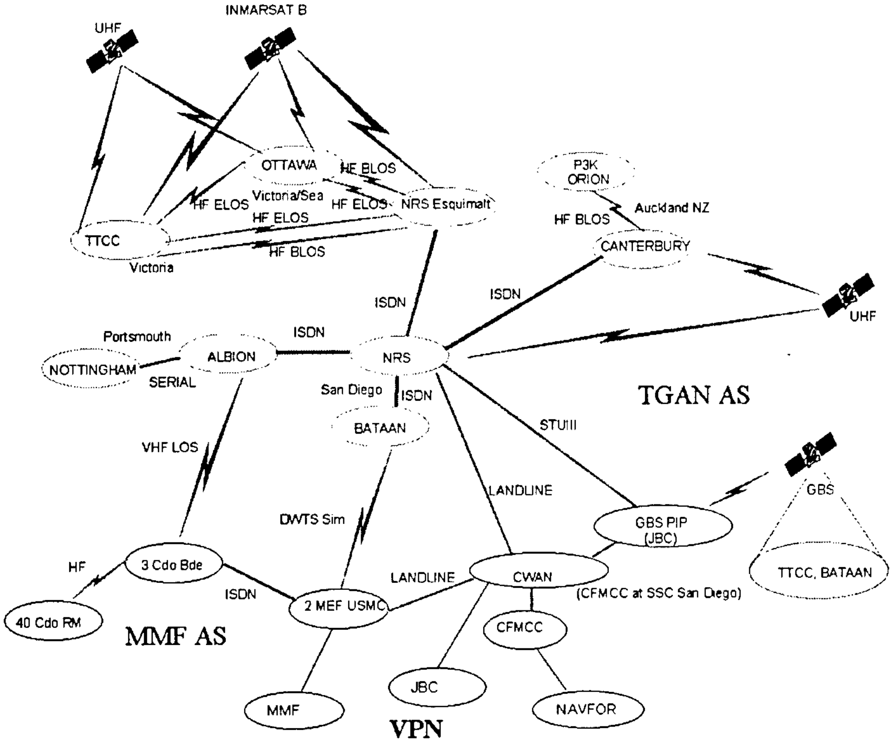


Figure 1. JWID-99R CWAN Connectivity Diagram

The network was managed using three different SNMP and ping based applications. An SNMP-based COTS product was installed on two nodes. It was used only to access local devices due to concerns of congestive collapse from excessive ICMP traffic generated during auto-discovery and monitoring. The product was deemed to be difficult to set up and difficult to use. `ws_watch`, a freeware program, was used to ping a set of assigned devices to determine connectivity. Configuration again proved to be a problem since IP addresses needed to be determined in advance. Finally the Network Control Tool (NCT), contributed by Canada was installed on several nodes. NCT uses SNMP to provide configuration and monitoring access to local devices. It also provides limited sub-network statistics. Though limited in its capabilities, the NCT was thought to be a valid first attempt at a management tool for such coalition networks.

Several requirements were identified from this exercise. Ease of use is important, especially since network management in military operations may have time critical components. Security of management is also a concern for systems where the CWAN is more globally accessible. Furthermore, of great concern was the ability to manage very low bandwidth sub-networks. SNMP was believed to saturate such links. Distribution of network management information was recommended.

3. ISSUES

There are several key issues for network management from both a commercial and defence perspective. With the increased size and heterogeneity of contemporary networks it is necessary to find scalable and adaptable network management systems. With the increased use of specialised applications with stringent communications requirements it is necessary to manage the network to ensure the desired conditions. Defence requirements such as those found in the JWID exercise also point to requirements for collaboration between different network service providers.

This section provides an overview of these and other issues that must be addressed by network management systems today. First, academic and commercial issues are expressed. Second, defence specific issues are investigated. Finally, the ISO management Framework is described in more detail, and four requirements of network management systems are expressed and explained.

The pressures of changing network conditions and military requirements suggest a switch to distributed network management. Networks are getting larger and require more scalable management solutions. There are also specialised requirements for certain network users, such as the military. In order to satisfy these requirements an understanding of the issues involved in contemporary network management is necessary.

3.1 Commercial Issues

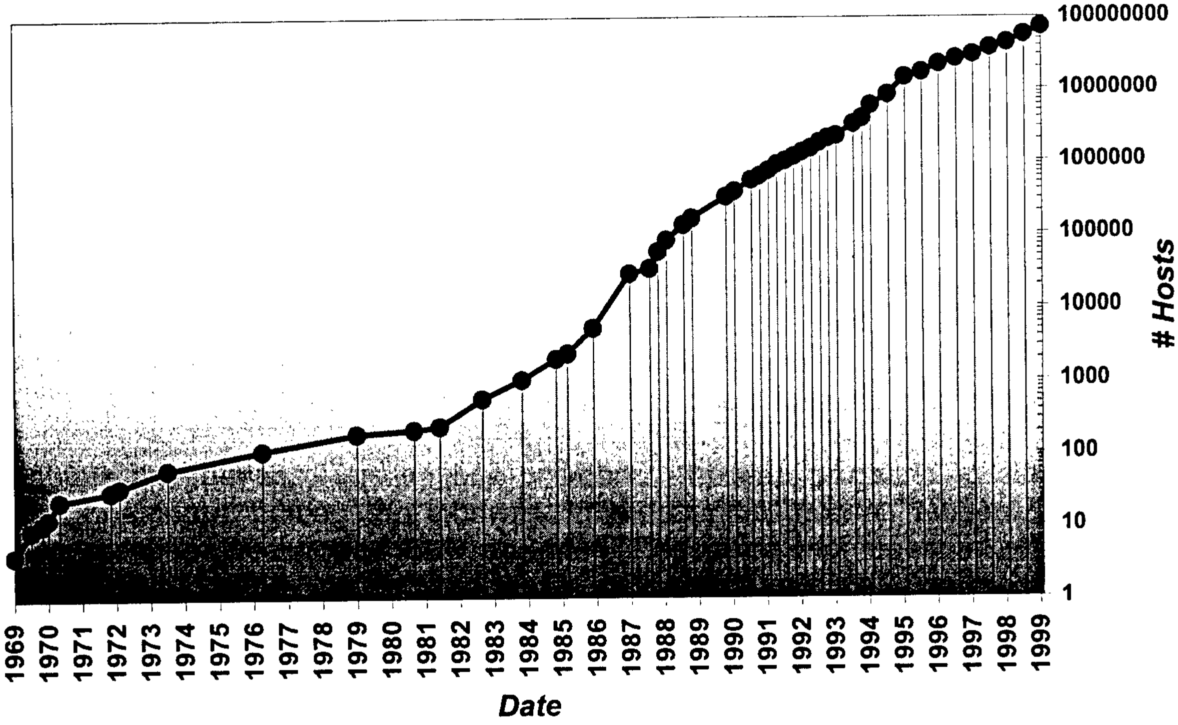
Management schemes have not developed in pace with the explosive growth in size and complexity of contemporary networks (See Figure 2). Networks are large, heterogeneous, and need to support increasingly complex traffic requirements. This has led to problems of scalability and adaptability. Current centralised systems cannot handle the traffic and processing burden of large and widespread networks. While networks were once mainly used for bulk data transfer, they are now becoming bearers of QoS sensitive data streams. Furthermore, the increase in network device vendors has made integrated management across devices difficult. Finally, in the commercial world of co-operating SPs and in the military world of coalition deployments, methods for federated management of shared infrastructure are needed.

3.1.1 Scalability

Reliance on computer aided automation for industrial tasks, and the switch from paper to electronic document creation and storage have all led to an increase in the number of computing devices. It has been the trend in the last decade to increase access to these devices and the information stored in them by connecting them into a network.

Consider the growth of the Internet shown in Figure 2*. The number of hosts connected to this worldwide network has increased exponentially for the past thirty years [6].

Figure 2: Internet Growth 1969-1999



Increases in connectivity have meant that the connectivity itself must be managed. Increases in the number of network devices managed have led to increases in communication and processing requirements. When control is centralised, the messages must travel to and from the central management station, focusing the traffic in the central Network Operation Centre (NOC) sub-network. As the number of devices being managed increases, the NOC sub-network reaches a point where it is overburdened with management messages. This may lead to saturation and failure of the network management system, and perhaps the network itself. Even if the management subnet can handle the traffic, the management system must still process the messages, integrate and analyse the management data, and then send out more monitoring and control messages. In other word, the processing power of the management machine must be taken into account as well.

Management systems must be able to handle increasing numbers of network devices in a **scalable** manner.

* This data was collected by Hobbes' Internet Timeline [6].

3.1.2 Adaptability

Increasing in connectivity has also created a growing market for communication and networking devices. Though dominated by one main network manufacturer (Cisco), there are many brands and types of devices, and the heterogeneity of equipment is likely to increase.

When network equipment manufacturers release their devices specialised management tools are released with them. In some cases, these tools are the only ones available to provide useful management tasks for that brand of equipment. Since standardised management tools cannot provide a common environment for problem correlation, managers move from one tool to another in order to accomplish complex tasks. Where integrated tools would find remote problems quickly, a human manager switching between various diagnostic tools could take significantly more time to locate the same problems.

One potential solution is to have management modules that translate between the proprietary management agents of some devices and a more standardised approach. Another would be to design a management system that could be extended to understand and integrate the proprietary management information of such devices into its operation.

Management systems must be able to **adapt** to an increasingly large variety of network devices, all in a scalable manner.

3.1.3 Utility

Related to both scalability and heterogeneity is the need for network management utility. As networks become larger and more diverse, they are also used for more numerous and diverse tasks. Furthermore, with the increased availability and stability of computer networks the way in which networks are used is changing. Management systems must handle networks that are used for real-time video (need for sufficient quality and bandwidth), and banking (need for increased security).

Currently emerging and future distributed multimedia public networks will carry traffic such as video, audio, and computer data with a broad range of QoS requirements, usually in terms of delay, jitter, and error rate. Meeting QoS guarantees is fundamentally an end-to-end issue, since QoS is most visible at the application level. To achieve this, end-to-end admission testing and resource reservation has to be done before flow of media information commences, along with active monitoring and maintenance of the delivered QoS while the flow is pending. Network management systems must have the utility to manage the network's capabilities in order to provide the **traffic characteristics** required by new applications

Another area of increasing concern is the security of management. As networks become the communication channels of choice for sensitive personal and financial information it becomes more critical that operations continue without failure or compromised data. Networks will become the focus for potential attack by those who would make use of such information, or benefit from the denial of service if such information were not transmitted correctly. Network management thus becomes a critical link. If network management is compromised, all manners of misdirection, compromise, and denial of service are possible. For these reasons, **security** is vital to network management.

Network management systems should increase utility in areas such as **QoS** management and **security** in order to deal with changing application demands.

3.1.4 Integration and Collaboration

Networks, including the Internet, have become so large that they are often subdivided into separately administered domains. Mechanisms for integrating management tasks across domain boundaries are needed. As corporate and military computer networks become more pervasive, they usually consist of local sub-networks that are connected through the networking services provided by another communication bearer to provide the illusion of a single wide-area network. As discussed in the QoS section of 3.1.3, there needs to be a way of managing flows end-to-end across possibly multiple Service Providers (SPs). This is especially true to enable the collaboration of management tasks across the various bearers to facilitate end-to-end QoS.

The integration of network management has been a priority for both defence and commercial bearers for some time. So far, management tools have been designed to handle either individual geographic areas, or individual areas of control, or logical division due to security considerations. Areas can also be divided by the types of equipment present. Network domains may not even be adjacent to each other, such as in the case of a large multi-national corporation whose network spans several continents and is interconnected by autonomous SPs. **Integration** is needed to co-ordinate management across domains.

Network management systems should **collaborate** with the management systems of adjacent networks in order to provide end-to-end QoS.

3.2 Defence Issues

Existing commercial tools were not designed to meet defence requirements for network management. Key issues include the ability to have both security of management and management of security, the ability to adapt to a dynamic networking environment, and the ability to operate in resource poor environments.

3.2.1 Security and Coalition Deployments

Security is an area of special concern to defence related network management systems. Security is important for defence in two main areas: security of management, and management of security. The former deals with the ability to provide management securely and the latter with methods for adding and removing the security aspects of resources; be they personnel, software or hardware. The latter is not discussed further in this report.

Historically the security of network management is monolithic. A single password provides access to the critical control mechanisms – a single management application has total control over the managed network. In a defence setting it becomes important to mirror the hierarchical nature of the command structure within the management security scheme. In this manner, higher level entities can override or modify the commands of lower level entities.

This becomes extremely complex in the case of coalition deployments. It is important to each nation that they retain control over their own assets, while collaborating (see previous section) with the other nations in the federated force.

Security is required so that parallel entities of each nation can share management information securely and provide a more integrated management environment.

3.2.2 Reliability and Resource Poor Environments

In defence networks both network elements and network topology are likely to change. It is, therefore, important to provide network management systems that are both robust and reliable.

The performance aspects of network control and management are of paramount importance in the military environment. Network management functions should be efficient and promptly performed both in the case of low/medium loads as well as during overloads. The amount of control/management information should have a minimal affect on the bandwidth available to users. This is particularly important in bandwidth-constrained environments.

Defence networks include a variety of networking resources that have characteristics different from that found in a typical commercial environment. In the wireless environment, relatively low bandwidth and high error rates imply the need for management systems that use a minimal amount of communication. As witnessed by the JWID-R exercise, bandwidth constrained management is becoming a defence research priority.

Network management systems should be able to operate, if at a reduced level, in **resource poor** environments.

3.2.3 Prioritisation and Guaranteed QoS

In military deployments, delivering end-to-end service guarantees is crucial. In stressed (overload) situations, the "best effort" policy (e.g. as currently used in the Internet) may lead to unacceptable delays, jeopardising whole missions. Simply adding transmission capacities to links and more processing power to network nodes to cope with the load does not solve the problem. The network needs to be able to efficiently perform and maintain QoS in stressed situations where only a part of all resources may be available. The transmission capacity of the currently impoverished satellite links used between strategic and tactical networks will be much lower than that available to ground based networks. Therefore, the delivery of QoS to satellite networks is of great importance.

In military networks, when insufficient capacity is available to support all traffic requests, messages carrying mission critical information should have a priority higher than less important ones. In addition, it is preferable to "step down" the QoS of less important military flows in overloaded networks than to release them. This capability is often referred to as graceful degradation.

Prioritization and management of per-stream priority should be included as a network management utility.

3.3 ISO Management Framework

Many of these problems were foreseen by the ISO in the early days of networked computing. The OSI management framework document [3] was under development at the same time as new protocols were being designed as an alternative to TCP/IP. The framework concentrates on five functional areas of management:

- *Fault Management,* which deals with the identification and correction of faults,
- *Accounting Management,* which deals with associating costs with resources consumed by a user,
- *Configuration Management,* which deals with initialising, updating and changing the configuration of devices,
- *Performance Management,* which deals with gathering statistical information and altering the system for performance reasons, *and*
- *Security Management,* which deals with security-related services.

Current management systems have focused on fault and configuration management. The ability to locate the source of network failures and to configure devices remotely provided sufficient utility for smaller networks that were under a single administrative domain.

Accounting management has never been of great importance. With the growth of multiple independent network Service Providers (SPs) there has been a growing movement to charge network traffic on a per-packet basis. This would require some kind of network accounting capability. The need for detailed accounting is also related to the desire of SPs to provide, and charge for, differentiated services. SPs would like to charge people more for an improved level of service, for example to charge extra for the QoS necessary for real time video.

Performance management is perhaps the most neglected of the functional areas, and one that is becoming increasingly important. Issues related to both defense and civilian network management utility involves questions of how to manage network performance.

Security functions have also become important as sensitive personal or financial information travels through the network.

3.4 Network Management Requirements

Four general requirements for an effective network management system for the systems described are proposed to address the issues outlined above. Such systems should be:

1. *Scalable,* able to handle large and increasing numbers of network devices in a scalable manner;
2. *Adaptable,* able to handle large and increasing varieties of network devices in a scalable manner;
3. *Utile,* have the utility to handle the granularity of control necessary to provide the QoS and other requirements of application level traffic; *and*
4. *Collaborative,* able to inter-operate with other network management systems to solve common or end-to-end problems.

These requirements will be referenced in the technology section.

The following section takes a look at the various network management architectures as a first step to designing a next generation network management system.

4. MANAGEMENT ARCHITECTURES

In order to design a network management system that will satisfy the proposed network management requirements, a suitable architecture must be selected from those available.

Management architectures have been divided into five categories: *Centralised, Hierarchical, Peer-to-Peer, Distributed,* and *Dispersed. Centralised* architectures rely on a central management station to collect, process, and act on management information. *Hierarchical* architectures are similar, but make use of sub-managers to delegate some management tasks. A *Peer-to-peer* architecture is similar to the *hierarchical* architecture except it has no central manager – sub-managers communicate directly to accomplish network wide tasks. *Distributed* architectures provide the next level of abstraction, where, instead of individual managers accomplishing all management tasks, management services are distributed throughout the network and can be accessed by management applications anywhere in the network. Finally, *dispersed* architectures contain no discernible management components – instead small management agents move about the network and use inter-agent communication and group intelligence to optimise network performance

Network management stations to date have consisted of three main components [7]: network managers, management agents, and a user interface. Network Managers (NMs) collect and process information from a set of software agents running on devices throughout the network. In order to gather that information, two things must be agreed upon by the NM and the agent; the management protocol (e.g. SNMP, CMIP), and a definition of the structure of the information being sent known as a Management Information Base (MIB). The information gathered from the various network devices' Management Agents (MA) by the NMs is stored in a management database (DB). The DB consists of all information gathered from management agent MIBs throughout the network. This gives NMs a consistent source of management data to work on. Along with the NMs and the agents, a User Interface (UI) is required to communicate with the human operator of the management system.

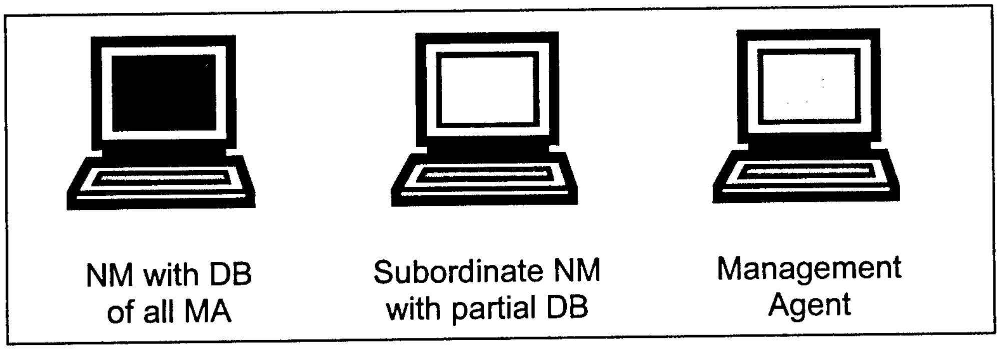


Figure 3: Management Architecture Components

Network management architectures differ in the location and task partitioning of NMs and associated DB. The prevalent architectures are shown in the following set of figures and described below. The associated classes of network management are described in the following sections.

4.1 Centralised Architectures

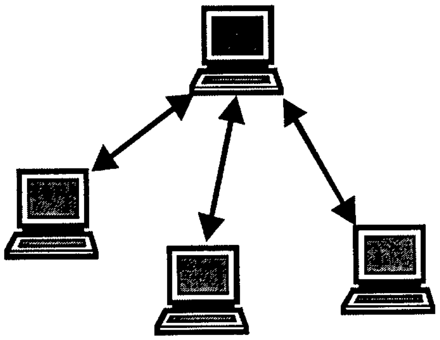


Figure 4: Centralized Arch

Centralised network management provides a single NM station with all management utility. The device agents are contacted directly from the centralised manager. The information gathered from the agents' MIBs is stored in the single DB local to the NM station.

This architecture is that most commonly associated with the ubiquitous Simple Network Management Protocol (SNMP). The simple architecture leads to an implementation that is both easy to implement and to use. Unfortunately, with the increases in the number of devices to be monitored, the NM station becomes a network traffic and computational resource bottleneck.

Most COTS network management systems are built using a centralised architecture. By gathering all network information at single point, deterministic algorithms can be used to determine the location and type of fault, even if the symptoms are in a completely different part of the network. This type of model is appropriate for networks that generate small amounts of network traffic relative to the available network bandwidth; networks in which network information can traverse the network quickly (in the order of a few seconds) and centralised processing can provide the necessary utility.

Examples of centralised COTS systems include the original versions of HP Openview, Sun Solstice and Tivoli's Netview. Some of these systems have been enhanced to make use of distributed techniques such as Remote Monitoring (RMON), an extension to SNMP explained further in Section 5.1.2.

4.2 Hierarchical Architectures

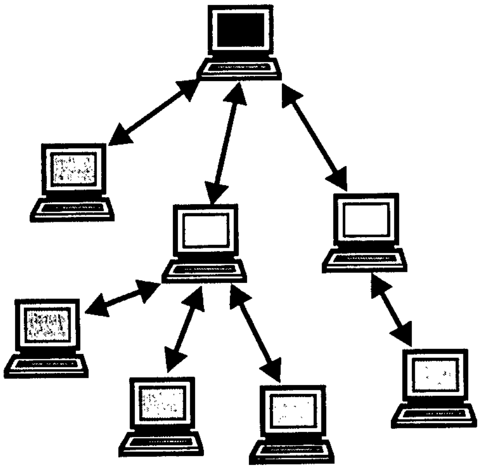


Figure 5: Heirarchical Arch

In the case of a hierarchical architecture, a single NM station maintains all information storage and processing functions. The central NM is, however, aided by a set of subordinate NMs. A single DB is maintained on the central NM while the subordinate NMs are relied upon for communication with most network devices.

This architecture provides some resolution to the problem of network traffic overload. Centralised processing still causes problems for very large and heterogeneous systems. The IETF's DISMAN group advocates this architecture, such that some management tasks are performed on remote devices.

A more distributed solution is required in cases where the network is more widely spread, includes devices with high management overhead, has portions with limited bandwidth, or has orthogonal management needs. A simple enhancement that can be made to centralised architectures is to add "mid-level management assistants" that aid in the processing or dissemination of management information.

An often-cited example in academe is the work of Goldzmidt and Yemini on Management by Delegation [8]. In their work, enhanced "elastic" management servers are placed on network hosts so that delegation agents can be moved to the device, reducing network latency and bandwidth utilisation to effectively nothing. These delegation agents implement management services and can be written in any arbitrary language. Thus the centralised management server can control management services on individual devices without actually needing to retrieve all the network information stored in the devices Management Information Base (MIB). For instance, a delegation agent could evaluate a health function at the device. If the function falls below a critical level; the central manager could be informed and further investigation follow.

4.3 Peer-to-Peer Architectures

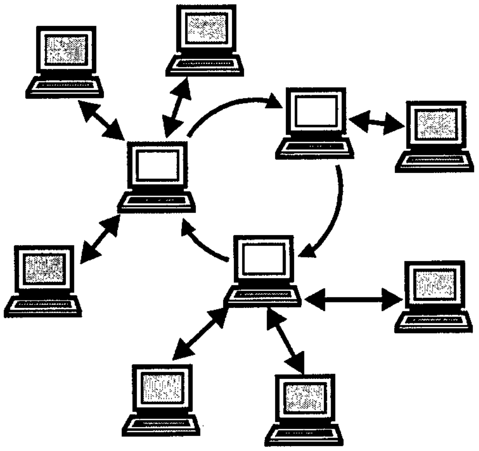


Figure 6: Peer-to-Peer Arch

For Peer-to-Peer architectures, a static set of co-operating NMs manage separate areas of the complete network. They communicate amongst themselves for the purpose of co-ordinated management between their various domains. These management domains are pre-configured to suit the managed network. NM stations communicate with the device agents in their specified domains. The DB of agent MIBs is partitioned amongst the various NMs, and may be partially replicated amongst NM stations in order to help with inter-station co-operation.

Peer-to-peer provides the simplest management scheme where multiple hierarchical architectures are chained together so that the once centralised managers communicate amongst themselves to co-operate in management tasks. There are currently no characteristic implementations of this architecture, though facilities exist in the latest versions of SNMP for inter-manager communication. The IETF DISMAN working group is also working in this direction (see section 5.1).

An alternative to hierarchical architectures, peer-to-peer is nearly identical except that the central manager has been removed, and end-to-end management is achieved through the co-operation of peer management stations. These stations are connected to sub-networks at strategic locations so that management traffic is reduced, and management tasks can continue even if a portion of the network is isolated.

An example of peer-to-peer NM in a COTS product comes from the Netrix "Distributed SNMP Management" product [9]. By distributing NETRIX network exchanges throughout the network, latency and bandwidth utilisation are reduced through local polling. SNMP polling only occurs between the Network Exchange switch and the local devices. Netrix contends that this allows more frequent polling which in turn leads to faster and richer data collection. Management information is shared between Exchanges. Thus any Netrix NMS console, no matter where in the network, can manage any SNMP device by contacting its local Exchange.

4.4 Distributed Architectures

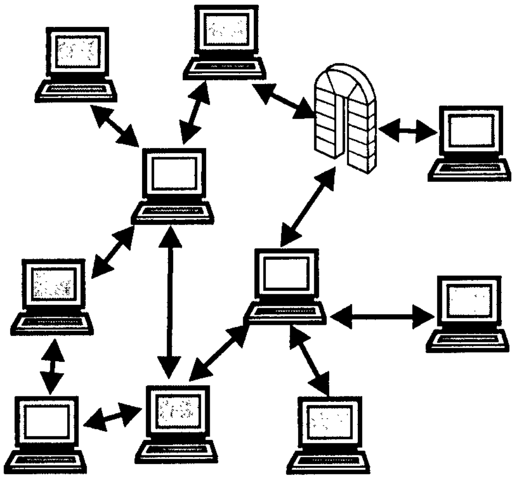


Figure 7: Distributed Arch

In a distributed network management architecture, a dynamic set of NMs manages sub-domains of the network, once again communicating amongst themselves for cross-regional management. The difference from peer-to-peer architectures is that the number and domains of NM stations is not set in advance. In this case, devices and subdomains may be added to and removed from NM regions. The DB is highly partitioned and highly replicated in this case. NMs communicate with their assigned devices and amongst themselves.

The distributed architecture is best suited to dynamic and heterogeneous systems where dedicated network management machines don't necessarily exist. This architecture is an example of that proposed by the JIDM. Devices are modelled objects that are accessed by a distributed management application. Devices are accessed dynamically using referral services so that their actual location and existence is not necessarily known before hand.

Distributed architectures is the next step in distribution for management. There are no longer any peers, simply various distributed management services that can be accessed from any point in the network. These services are robust enough to continue working even if the network partitions and can operate across opaque network: regions which connect individual portions of a larger network but which cannot be directly controlled. This was described earlier in section 3.1.3.

Much work has been done on using distributed middleware such as CORBA [10], but there are few implementations. In these works, the distributed object computing services of CORBA are used to access devices that are modelled as objects themselves. In order to deal with legacy protocols such as SNMP, a protocol gateway is often introduced to convert between a CORBA object model and the SNMP MIB representation. However the concept of independent distributed management services has yet to be explored. This proposal is investigated in more detail in section 7.

Another potential realisation of the distributed paradigm comes from the Jini connection technology [28]. Originally envisaged as plug-and-play for network devices, it also has potential applications for network management. As described in section 5.4, Jini-like systems could provide a simple type of automated management.

4.5 Dispersed Architectures

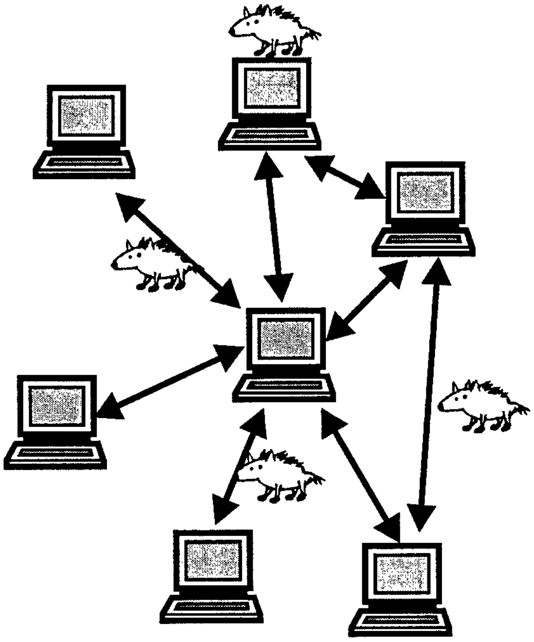


Figure 8: Dispersed Arch

For a dispersed architecture, a large number of mobile NMs manage the network on a per-device basis. No central DB exists; it is distributed amongst NMs so that local information and some remote information is stored in each NM's DBs. Individual NMs may take on or give up management of devices at will. NMs communicate with other NMs to accomplish management tasks.

This architecture takes management to the most distributed arrangement possible, where the network management functions are dispersed amongst a large number of dynamic NM stations. Agent-based network management best represents this architecture. AI techniques provide us with intelligent and possible mobile management agents that control and monitor the network under the guidance of a group intelligence. Agent technologies are unfortunately immature enough to provide significant implementations at this time.

For dispersed architectures, instead of distributed management services, you have independent pieces of management intelligence that collaborate to maintain peak network operation. The concept of a management service no longer has meaning, though the agents released into the network may have detailed and exact goals to accomplish.

An example of an agent based network management system can be found in the work of Bell Labs and Carleton University [11,12]. They use small pieces of mobile code that implement AI techniques to do various network management tasks. For instance, "deglets", or delegation agents can be used for network modelling and discovery. The deglet travels the network collecting device identifiers, and depending on the constraints placed, the network model created can be specialised for various management tasks. This methods improves on existing methods such as pings *or* SNMP gets *since the agent can be designed with multiple fallback procedures. Agents based management is discussed in more detail in section 5.5.*

5. ENABLING TECHNOLOGIES

This section of the report deals with the technologies available to satisfy the network management requirements outlined in Section 3. It is argued that in order to satisfy the first two of the requirements, scalability and adaptability, one of the distributed architectures described in Section 4 is required. Centralised architectures form the basis of most COTS offerings. Such monolithic solutions are unlikely to be scalable to the size of the network they manage, or adaptable to sensitive portions of the networks, such as low bandwidth wireless links. For this reason, this section reviews the operation and standardisation of several existing and proposed Distributed Network Management (DNM) solutions.

We divide the available distributed management technologies into three categories. *Peer-to-peer* technologies provide extensions to existing management solutions such that the network can be divided into neat sub-networks. Each is managed separately but can communicate with peer sub-network managers for common tasks. *Distributed* technologies make use of some distributed middleware, such as the Common Object Request Broker Architecture (CORBA) [13], to distribute management services throughout the network. Finally, *Agent* technologies make use of ideas from the field of artificial intelligence so small pieces of code may travel from network device to network device managing the network using a group intelligence.

Peer-to-peer management is based on an extension of the existing management frameworks. The original client-server interactions of centralised solutions are made distributed with the help of replicated managers and manager-to-manager communications. Management information is gathered and processed at a set of management stations throughout the network. The locations of these stations must be configured globally, and their functions made known to other stations.

Technologies based on the peer-to-peer architecture are the subject of a range of standardisation activities and implementations. One of the advantages of peer-to-peer management is that existing management methodologies can usually be replicated across the network. The difficulties come from integrating the replicated pieces into a coherent management system. The two main technologies that can be used to achieve peer-to-peer DNM are the next generation of **SNMP** and **WBEM**.

Distributed-object-based technologies make use of a well-understood middleware base, on top of which the DNM utility can be built. Using a management interface protocol such as SNMP can accommodate device heterogeneity.

Distributed object based technologies are not as mature as Peer-To-Peer management methods. There is however one main standards group which is working with CORBA to come up with a management framework, the **JIDM**. Also, the **JINI** connection technology provides a distributed object plug-and-play paradigm that is useful for device configuration.

A third category of DNM system is based on active networks. In this paradigm, devices do not support a single management agent, but provide an environment to run specialised intelligent and perhaps mobile agents from the network. Based on artificial intelligence techniques, these agents wander between and control devices as dictated by a group intelligence designed to optimally manage the network.

Although there are currently no agent-based network management systems, it is a lively area of research. Three main types of agent system are described in this section of the report. **Java Agglets** provides language extensions to Java that allows "agentification" of code. **Concordia** is one of several proprietary language extensions designed specifically for creating software agents. Finally, **Grasshopper** is an academic agent system that is based on the few scattered standards on network management agents.

One area not addressed by these distributed technologies relates to the final of the four network management requirements of 3.4, that of collaboration. The distribution and specialisation of management operations throughout the network can to differing degrees satisfy the requirements of scalability, adaptability and utility, but does not inherently satisfy the last requirement. One technology that potentially provides a means to this end is so-called policy based management.

In order to inter-operate with separate management systems, some kind of collaborative technology is required. Policies provide mechanisms for automating management to satisfy network operational policies. The technology often ascribed to this goal is **Policy Based** or **Directory Enabled** management and is covered in more detail in Section 7.2.

5.1 SNMPng

The Internet Engineering Task Force (IETF) [14] developed the first versions of SNMP [1, 2] in the early 1980s. SNMP sends control and monitoring Protocol Data Units (PDUs) to management agents located on SNMP-compliant devices. These agents gather statistics (variables) about the state of the device and store them in their local Management Information Base (MIB), a repository for information on the configuration and internal state of the device. When an agent receives an information (`get`) request PDU from an SNMP manager, it parses the PDU and responds with the appropriate 'variable' value from its MIB. When it receives a control (`set`) PDU, it will change the corresponding value in the local MIB to the requested value. Application daemons on the device watch for these changes and respond by an appropriate reconfiguration of the device.

This simple send – reply protocol has its limitations. Since messages are point to point, a centralised architecture is the simplest and most commonly implemented. This message pattern and architecture does not scale to large or heterogeneous networks. Also, the utility provided by SNMP is very primitive and in need of updating.

In response to these limitations the IETF has been, and is still in the process of, developing the next generation of SNMP. Several enhancements to SNMP have been made. The architecture has been redesigned with the addition of strong security features in SNMPv3. Enhanced capabilities are being developed by the RMON, AgentX, and DISMAN working groups.

With the input of these working groups a more distributed network management architecture is being developed. Remote monitoring (RMON) and remote script (DISMAN) based management provides a kind of primitive peer-to-peer distributed architecture. Since these extensions are consistent with the existing *de facto* standard there is great potential for this colossal effort. These enhancements can be tied together to fashion a distributed network management system based on the peer-to-peer architecture. The advantage of this system, and its greatest drawback is its reliance on the wide distribution of the old SNMPv1 agents. The knowledge and expertise behind SNMP is considerable, but it remains to be seen if it can be extended quickly to make use of the new DNM enhancements, and so allow for a scalable and secure system.

5.1.1 SNMPv3

The third version of the SNMP protocol, SNMPv3, was released in April 1999 as a standard track Request For Comments (RFC). One of the drawbacks of SNMPv1 is its lack of strong security and administration components. The third version enhances the protocol by adding the strong authentication and security modules necessary to support management of sensitive networks.

5.1.2 RMON

In order to deal with the problems of accessing distant devices, the IETF's RMON working group defined a set of managed objects for remote monitoring of networks. These objects provide the ability to monitor multiple network layers of traffic in remote networks, providing remote fault and configuration management while retaining consistency with the SNMP framework and standards.

5.1.3 AgentX

Another enhancement to SNMP comes from the IETF's AgentX Working Group. The Agent Extensibility (AgentX) protocol provides a platform-independent protocol that supports inter-agent communication within a device or local area network. This provides the ability of device agents to maintain co-operative management tasks throughout the local area, a useful DNM feature.

5.1.4 DISMAN

Most interesting from a distributed network management point of view is the IETF's DIStributed MANagement (DISMAN) Working Group. This group limits itself to distributed network management applications based on SNMP. The main goal of the group is to define objects that are consistent with the SNMP framework but extend its utility through the use of special script, schedule, and threshold monitoring MIBs. The script MIB is designed to allow management instructions to be evaluated at a distant management station. The schedule MIB allows management tasks to be executed at particular times, and the threshold monitoring MIB can notify managers when a threshold is passed, removing the need for application directed polling.

5.1.5 Case Study: Global Network Management System

One example of SNMP extensions comes from attempts to bring remote monitoring and control to individual radio receivers and antenna arrays that were not originally designed for such control. SNMP based COTS management systems were used to control devices such as individual radios, which may have well defined interfaces but no SNMP agent, or to manage devices such as antenna control systems which have interfaces that do not easily match with the SNMP management model. The distribution and heterogeneity of the systems would appear to call for a distributed solution.

In the work by Aicklen and Main [15], specialised SNMP MIBs and intelligent proxy agents (IPAs) were designed for each component. Remote monitoring and control functions were then integrated with standard COTS network management software such as HP Openview. Using this method, several key design considerations were met; utility, reliability, security and cost.

Management *utility* was provided by a special user interface developed to provide an interface to the newly connected devices. For instance, the AN/WSC-6 satellite terminal consists of several up-converters, down-converters, antennas, and other interfaces. In order to provide access to this utility the user interface was designed to provide logical instrumentation for all aspects of the antenna control unit. Management *reliability* proved to be a problem due to the unreliable packet based nature of the SNMP protocol. This problem was addressed by adding acknowledgements to ensure that traps (asynchronous messages from agents to the manager) are received. A second improvement was to make SNMP traps a trigger for the polling of the SNMP device. *Security* was provided by functions in the then prevalent SNMPv2 security systems, and *cost* was thought to be well dealt with by the efficiencies found in extending existing protocols and network management systems.

Thus, the use of standard protocols reduced time and cost while maintaining interoperability with existing management systems. It was recognised that many legacy systems could benefit from network control. While significantly different from the routers and bridges of standard networks, the communication devices were found to be manageable through the use of the SNMP protocol extended with specialised software.

5.2 WBEM

Web Based Enterprise Management (WBEM) [16] from the Distributed Management Task Force (DMTF) [17] initiative is "based on a set of management and Internet standard technologies developed to unify the management of enterprise environments". It includes a Common Information Model (CIM) [18] for describing management data. This is different from both the OSI and IETF MIB model, and is based on the eXtensible Markup Language (XML) [19].

"The CIM specification is the language and methodology for describing management data. The CIM schema includes models for Systems, Applications, Networks (LAN) and Devices. The CIM schema will enable applications from different developers on different platforms to describe management data in a standard format so that it can be shared among a variety of management applications. The xmlCIM Encoding Specification defines XML elements, written in Document Type Definition (DTD), which can be used to represent CIM classes and instances. The CIM Operations over HTTP specification defines a mapping of CIM operations onto HTTP that allows implementations of CIM to interoperate in an open, standardised manner and completes the technologies that support WBEM." [16]

In effect, WBEM places management servers on every device. Device variables are accessed from centralised management stations. The increased per-device overhead does not scale and introduces limitations on devices for resource-poor environments. It is however an excellent example of the management by delegation paradigm proposed by Goldszmidt and Yemini [8]. By introducing the per-device web-server as a location to process management information, a more hierarchical architecture is achieved.

WBEM is envisaged as managing more that networks. It is argued that both application and higher-level network services can also be managed used using the WBEM framework.

Though not yet widely deployed, the industrial backing by such companies as Intel, Cisco, Sun, and Microsoft may help this standard, regardless of its technical merits. Microsoft has produced a reference implementation of WBEM called Windows™ Management Instrumentation (WMI) [20] which is currently available on most versions of the Windows operating system. Several WMI based applications have been created, but as of yet, all are based on centralised servers accessing the devices directly.

5.2.1 Case Study: Tivoli Systems WBEM-Solution

Tivoli has used the WBEM standards to create a management backplane on which existing Tivoli products can be used [21]. The main gain for Tivoli is the standardisation of management information structure through the use of CIM.

Tivoli's WBEM architecture is divided into three main layers. The model and measurement layer is comprised of the managed objects that make up the CIM schema. The Knowledge layer reacts to changes in the model and measurement and conveys the relevant information to the decision layer. While the knowledge layer is meant to be proactive by reacting to and solving evolving problems in the lower layer, the decision layer is meant to be a tool to aid operators in their management of the system as a whole. CIM and WBEM thus provide a mechanism to provide the model and measurement layer while existing Tivoli products, appropriately modified, are to be used for the implementation of the knowledge and decision layers.

5.3 JIDM (CORBA)

Distributed-object-based management is commonly associated with the Common Object Request Broker Architecture (CORBA) [13] standardised by the OMG. CORBA provides the services necessary for invoking remote object methods through the use of an Object Request Broker (ORB). An ORB matches object invocations with the possibly remote object on which to apply it. While local objects are invoked as usual, remote objects are referred most commonly through the Internet Inter-ORB Protocol (IIOP). Management utility can then be distributed by objectifying and distributing it. However, some form of access to the devices themselves is required. This can be accomplished either by developing an object execution environment on the network elements, or by using existing interfaces like SNMP.

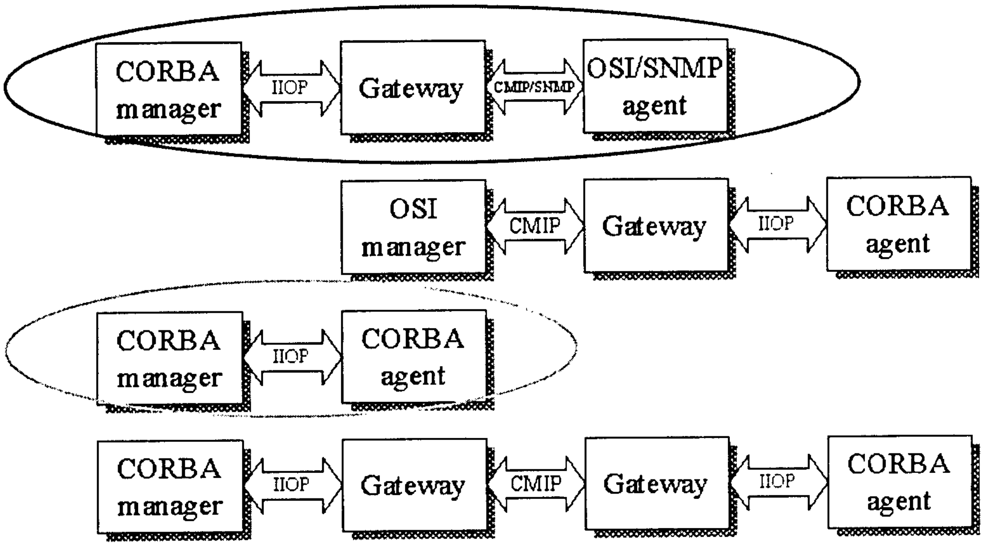
An alternative to CORBA that has since fallen from favour is Microsoft's DCOM [22]. Based on the Component Object Model (COM), Distributed COM (DCOM) is based on the Open Software Foundation's DCE-RPC specification. It was designed to support communication among objects no matter where they may be located throughout the network. The main goals of DCOM include scalability, security, support for fault-tolerance, and platform independence.

The Joint Inter-Domain Management (JIDM) [23] project is supported jointly by the TeleManagement Forum, the OMG, and the Open Group. The accompanying standards define how network management components based on OSI and SNMP can interoperate with CORBA objects. The specifications are divided into Specification Translation [24], which defines how to translate between different information models using IDL, and Interaction Translation [25], which defines how to perform management services from CORBA using a Gateway.

JIDM envisages four scenarios as shown in Figure 8. This CORBA / Management interoperability is defined by the relationship between Telecommunication Management (TMN) agents and managers where at least one of them is built on CORBA. Thus both CORBA manager / OSI-SNMP agent and OSI manager / CORBA agent relationships are considered.

The scenario of greatest applicability to existing networks is circled in red. In this scenario a CORBA manager communicates the a JIDM gateway using the CORBA Internet Inter Orb Protocol, the current TCP/IP standard protocol for CORBA objects to communicate with remote objects. The Gateway translates management-based requests from the CORBA manager into SNMP protocol data units (PDUs) and sends these packets to the appropriate device. In the reverse direction, communication from the SNMP agent on a device is captured by the JIDM gateway and translated into the appropriate object response or invocation to return to the CORBA manager.

Figure 8 – JIDM Scenarios



Besides network management based on SNMP, the JIDM gateway can also be used to manage OSI based telecom devices that use the CMIP protocol. Corba based Telecom Management is not new [26], however the JIDM approach allows a more global scope for management of devices that use different management protocols.

JIDM is also seen as a transitional technology towards the third scenario above (circled in green). In this case, CORBA managers communicate directly with devices that use CORBA objects as internal representations of management state. In this way, the intermediate gateway is removed and network managers can communicate directly with device agents.

The JIDM's CORBA-SNMP approach is the foundation for the proposed network management solution and is outlined in Section 6.2.

5.3.1 Case Study: Nokia's Distributed Computing Platform

This work by Rahkila and Stenberg [27] of the Nokia Research Centre outlines the Distributed Computing Platform (DCP) developed to provide tools, compilers and gateways to support both OSI and SNMP management through a CORBA infrastructure. Using CORBA as a base, DCP handles network management by adding managed-object models and protocols. It provides mechanisms that allow communication between CMIP-based objects and a gateway for SNMP-based systems. The prototype also allows users to access network information via HTML Web-browsers and Java.

The prototype developed by Nokia provides a proof of concept for CORBA based distributed network management. CORBA was a good integrator of technologies and provided a useable distributed platform. Java and HTML provided a good development framework on which to base network management applications and interfaces.

5.4 Jini

Jini connection technology [28] was not designed for network management, but the ideas on which it is based could be very useful for configuration utility in DNM. Jini is an extension of the Java object technologies where networks are defined as a pool of resources. As new devices are added to a network, they register with a central database that automatically configures the device and makes its resources available to the rest of the network, while also making the existing network resources available to the new device. This type of automatic configuration would be a great help to network management if it could be scaled to an entire network.

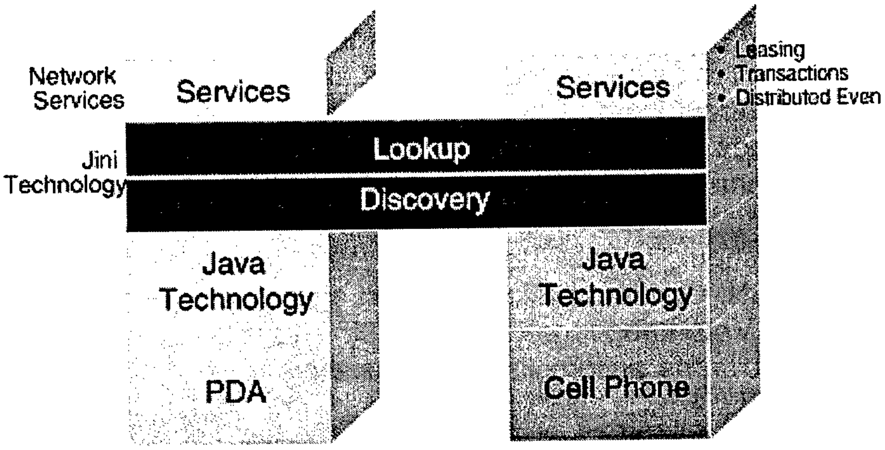


Figure 9: Jini Stack

Jini connection technology consists of an infrastructure and a programming model, which address the fundamental issues of how devices connect with each other to form an impromptu community. Devices such as PDAs and Cell Phones use discovery techniques to register themselves with the lookup service. The lookup service stores pointers to devices and their associated code and available resources. For instance when a printer connects to the network, it can publish its "printing" service, printing specifications, and all drivers necessary for use.

Since Jini is not a network management technology, no case study is given.

5.5 Agents

A final category of DNM technology is based on the idea of active networks. Distributed technologies must provide extra functionality at the network device, and active networks do provide a solution to software mobility, autonomy, and intelligence for the implementation of flexible service solutions and adaptive computing systems.

In this paradigm, devices do not support a single management agent, but provide an environment to run specialised intelligent, and perhaps mobile, agents from elsewhere in the network. Based on artificial intelligence techniques, agents wander and control devices as dictated by group intelligence designed to optimally manage the network.

The question of what exactly an agent is remains a hot topic in agent research. Some view agents as simple pieces of code which respond in a pre-defined way to input from a remote management application. This is the standard view of SNMPv1, which defines rigid communication protocols and management information structures.

For the Artificial Intelligence (AI) community, an agent is a software program that automates a series of computations on behalf of a user even when the user is not connected to a network. An agent performs its assigned work as defined by the developer (and authorised by a user), has a measure of autonomy, of critical thinking, and of methods for communicating with other agents. Agent autonomy means that agents will act according to their own internal state, and not necessarily deterministically as viewed from the external word. A common view is to attribute goals and desires to agents, with the methods of achieving them not being defined in advance. Agents are also thought to have some measure of problem solving ability. Agents should be able to do some critical thinking to determine the best actions to take in order to realise their goals. Finally, agents should be able to communicate both with the outside world as well as with other agents. Communication amongst agents is often accomplished using the Knowledge Query Meta-Language (KQML) [29]. This allows agents to share goals and thus collaborate towards accomplishing tasks they could not have accomplished alone.

One final attribute that is often debated in the AI community is whether agents should be mobile. Mobility adds to an agent's autonomy since they can operate at multiple locations. It also allows agents to reduce the amount of communication across the network since the agent can move to the location it wishes to communicate with and interact locally. A trade off must be made to accommodate the bandwidth required to move the agent and its state.

Though some agent standardisation effort has been going on in the Object Management Group (OMG), only the Foundation for Intelligent Physical Agents (FIPA) has done standardisation work related to network management [30]. FIPA is an international agent standardisation body whose main focus is on the definition of a generic Agent Communication Language (ACL) to allow agents of different vendors to interact. The FIPA `97 specification includes a section on network management and provisioning which includes an Agent Management System (AMS) and inter-agent communication [31]. There has also been a recent proposal for collaborative work between FIPA and the OMG [32]. This liaison work may lead to CORBA services that support object mobility and the enhanced object lifecycle support necessary for the "agentification" of CORBA objects. Agent standards and their implementation are discussed in more detail in the Grasshopper case study in Section 5.5.3.

Agents can be developed in three styles of agent technology. Java Aglets are Java applets with agent extensions added to support mobility and other agent related actions. Aglets are thus classified as a **language extension** to Java. Another type of agent technology comes from specialised agent languages. These languages, though perhaps similar to common programming languages, have been designed to support the implementation of agents. Two agent programming languages are reviewed, one commercial and one academic. Concordia is a **proprietary agent language** designed by Mitsubishi's Horizon Systems Laboratory. Grasshopper is a **standards based agent language** under research by the German research lab IKV++ GmbH. Both are now based on Java.

Though a very promising field for the future, several problems, including security, monitoring and control of the agents themselves, must be addressed before agent technology can become widely accepted as a DNM solution. There are currently no widely accepted agent based network management solutions, though several prototypes are under development.

The following case studies showcase the three main types of agent technologies. Java Aglets provide a language centred approach to the programming of agents. Concordia represents a commercial model of language extensions based on the Java programming language. Grasshopper is also based on Java, but is a more standards oriented solution.

5.5.1 Case Study: Java Aglets

Aglets [33] are a Java language extension developed by IBM's Tokyo Research Laboratory. Aglets are Java objects that can move from one host on the Internet to another. An aglet executing on one host can suddenly halt execution, dispatch itself to a remote host, and resume execution there. When an aglet moves, it takes along its program code as well as its data.

The main contribution of the IBM Tokyo lab is the open source distribution of an Aglet Software Development Kit [34]. This kit includes a user interface named Tahiti that can be used to monitor, create, dispatch, and dispose of agents. It can also set the agent access privileges for the agent server.

5.5.2 Case Study: Concordia

Like Java Agglets, Concordia [35, 36] is a framework for the development and management of mobile agent applications written in Java. Unlike Agglets, Concordia is a commercial product, which has been developed by Mitsubishi Electric's Horizon Systems Laboratory. An evaluation kit is available but does not include any security or reliability components.

In Concordia, Agents are Java objects that provide mobility, security, persistence, collaboration, and disconnected operation. An Agent Tool Library provides development tools including all APIs and agent classes needed to develop Concordia mobile agents.

Concordia is made up of several components. An *Agent Manager* provides the communications infrastructure to allow agents to travel. A *Security Manager* protects resources and ensures the security and integrity of mobile agents and their data. A *Persistence Manager* maintains the state of mobile agents and objects in transit. An *Inter-Agent Communication Manager* handles the registration, posting and notification of events to and from mobile agents. A *Queue Manager* schedules and guarantees delivery of mobile agents between Concordia servers. A *Directory Manager* provides the name service for applications and agents. Finally, an *Administration Manager* provides remote administration of Concordia servers.

Concordia has been a commercial product for some time, but only minor changes have been made since its last major release in 1998. For this reason it has only moderate system requirements (486, Java 1.1), and does not conform to any of the agent standards mentioned previously.

5.5.3 Case Study: Grasshopper

Grasshopper [37] is a relatively new standards based mobile agent development platform released by IKV++ in 1998. It was developed to be as flexible and open as possible. It is written completely in Java and is compliant with both MASIF and FIPA international agent standards discussed previously. It can also provide integration with distributed object platforms such as CORBA and DCOM. Though highly academic in its inception, Grasshopper is currently under continued development and is available for commercial use. Free downloads of the complete system are available for personal and non-commercial use.

The Grasshopper platform architecture is structured into three major parts. It includes a core system which provides the necessary capabilities for developing and running Grasshopper agents, and two optional open source extensions which provide the OMG MASIF and FIPA standard interfaces for agent/platform interoperability.

The Grasshopper core system is a pure Java-based mobile agent platform, providing all of the functional capabilities necessary to develop and run agent applications. The

Grasshopper system environment consists of several Grasshopper agent systems (agencies), grouped within a domain (region), such as an intranet. An agency is a Java process that enables and controls the Grasshopper agents. It includes services such as security, agent registration, persistence, agent management, agent transport, and communication.

The MASIF and FIPA add-ons are open source (GNU General Public License) Java class libraries. The MASIF add-on implements two specific CORBA Interfaces, the MAFFinder and the MAFAgentSystem. These interfaces and their associated standard data types are required for cross platform mobile agent applications. The FIPA add-on implements the FIPA Agent Platform (AP) which comprises the Agent Management System (AMS), the Directory Facilitator (DF), and the Agent Communication Channel (ACC). These provide the mechanisms necessary to provide agent management and an Agent Communication Language (ACL).

There is currently no network management functionality inherent in the Grasshopper system, though their web site [38] claims that SNMP interfaces will be made available within a year. Their future plans include web-based control of agents and agencies, and a Grasshopper-Media Framework such that agents could deliver content via phone, e-mail, fax, etc.

6. SOLUTIONS

Now that the main problems and technologies related to network management have been discussed, this section provides a brief comparison of the relevant technologies and their application to different network management situations. A proposal for further research is made involving previous studies involving the University of Quebec at Montreal (UQAM) and a Corba-SNMP gateway.

6.1 Comparison of Technologies

From the original discussion of network management issues, a matrix of management technology versus issue can be constructed as shown in Table 1. This table shows the extent to which a technology supports a solution for the associated issue.

Table 1 Management Technology vs. Issue

	Scalability	Adaptability	Utility	Collaboration
SNMP	Limited	Limited	Limited	No
WBEM	Limited	Limited	Extensible	No
JIDM	Distributed	Fully	Extensible	Possibly
JINI	Distributed	Fully	Extensible	Possibly
Agents	Unknown	Fully	Extensible	unknown

6.1.1 SNMP

Though several techniques for scalability are being explored by the various working groups in the IETF, as was discussed in Section 5.1, the SNMP architecture cannot support a truly distributed network management application. An application distributed throughout the network would require a complete SNMP protocol stack in all of its communication nodes, and would need to use internal data storage to store and correlate management data. Thus, scalability is possible, especially the hierarchical model, but is not well supported by this technology. Adaptability is similarly limited.

The services offered by SNMP do not meet the utility requirement previously described. Also, the functionality necessary to support collaboration between network management systems does not exist within the SNMP architecture.

6.1.2 WBEM

WBEM is designed along similar lines to SNMP and thus much that has been said for SNMP can be said again for WBEM. The WBEM architecture is not inherently a distributed one and does not fully support distributed network management applications. The ability of WBEM to support a large variety of network devices is not inherent in the WBEM architecture.

One of the main differences when compared to SNMP is the extensibility of WBEM. Whereas SNMP defines a small rigid set of operations, WBEM provides an operation environment in which the mini web-servers that run on devices can provide whatever functionality required. There is no mechanism internal to WBEM that provides the ability for collaboration.

6.1.3 JIDM

Since the JIDM gateway is simply an extension to allow CORBA applications to access SNMP enabled devices, little is inherently different between SNMPng and JIDM. However, the addition of CORBA as a backplane provides several advantages in resolving the issues presented here. Since CORBA provides a distributed execution environment, JIDM technology supports a scalable distributed network management. The technology can thus adapt to the heterogeneous network environment by localising the changes necessary for the manager to communicate with the device agents to separate parts of the distributed management application.

CORBA also provides the ability to add arbitrary functionality to the management application, allowing full utility. Collaboration between JIDM based management systems is not inherently supported, but could be added if an external interface were defined.

6.1.4 JINI

Jini works along similar lines to the CORBA portion of JIDM and thus the issues addressed are almost identical. The distributed nature of Jini provides both scalability and adaptability.

Management extensions to Jini could also provide the utility required without changes to its operation or architecture. Once again, collaboration amongst management domains would require a common interface.

6.1.5 Agents

Agents, as a dispersed architecture, are by definition distributed. However, there is little known about their operation or support for the various issues. It could be argued that the number of agents would increase linearly with the number of devices, but much would depend on the "personality" of the agents and the way in which they interact. Since agent domains on various devices would be standardised, the type of devices should not matter to the operation of the agent-based network management system.

Only the type and number of agents inserted into the network limit the utility of such a system. The ability of agents to collaborate between management domains remains unexplored.

6.2 A Distributed Corba-SNMP Services Proposal

Our network management system uses CORBA as the distributed object middleware, and SNMP for management functionality on the network elements. SNMP was chosen due to its ubiquity. SNMP, the *de facto* management standard with agents implemented on most network devices, is a mature yet evolving standard. CORBA, which provides the distributed middleware for management services, is similarly based on stable but evolving standards and has several mature implementations. CORBA provides a way of invoking methods on remote objects, such as JIDM gateways and through the gateway of contacting SNMP enabled devices, without necessarily knowing in advance the location of those objects, their exact functionality, or even the language / architecture in which they are implemented.

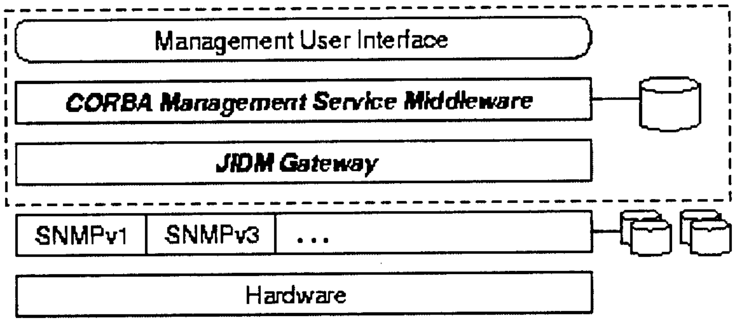

Management User Interface
CORBA Management Service Middleware
JIDM Gateway
SNMPv1
SNMPv3
...
Hardware

Figure 10: Management Architecture

6.2.1 Architecture

The architecture is composed of five layers (Figure 10). Network hardware is monitored and controlled using SNMP at the second layer. The JIDM gateway provides a mapping between CORBA method invocations and the SNMP protocol operations at the third layer. The fourth layer is comprised of the distributed CORBA-based management services. These services support the various management functions defined by the OSI Management Framework [3]. A user interface, at the fifth layer, allows human control of the management services from any location in the network.

This distributed architecture provides a number of advantages for managing coalition networks. Consider the generic coalition network configuration shown in Figure 11. A number of national subnetworks are connected by various types of links, in one case through an autonomous (privately managed) network. Managed subnetworks contain SNMP enabled devices, CORBA management-service objects and at least one management gateway (for redundancy). A user interface may be present at any location. The distributed management objects communicate across network boundaries using CORBA's Internet Inter-Orb Protocol (IIOP).

6.2.2 Operation

For management of local subnetwork elements, requests from the user are passed, via the user interface, to the CORBA management-service objects. Invocations on the local JIDM gateway by these objects are translated into SNMP packets sent to the appropriate device(s) in the local subnetwork. In this case, the advantage of a distributed architecture is the simplification of the management services on the local network. These services can be tailored for the specific environment, and do not need to deal with the details of equipment outside of the local area.

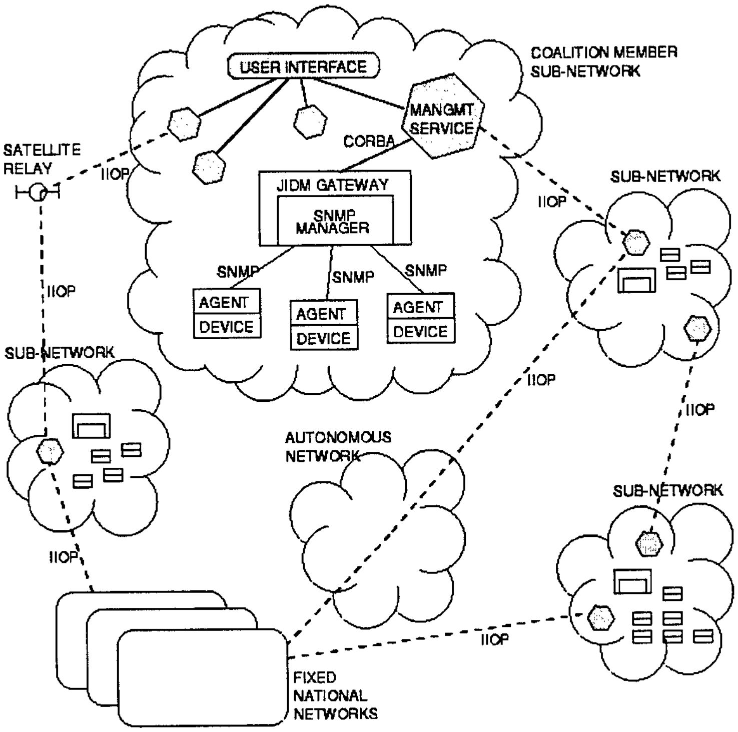


Figure 11: A generic Coalition Network

To manage remote subnetworks, management-service objects communicate using IIOP with objects or gateways in those remote subnetworks. Invocations may travel through intermediate autonomous networks. The exact location of these objects need not be known in advance. The remote objects may in turn contact other more remote management objects, and so on. Local invocations are serviced as before by the JIDM gateway, perhaps requiring special policy considerations if national boundaries or autonomous networks are involved. The ability to treat all connected subnetworks as a single distributed whole while isolating the heterogeneity and SNMP traffic to a single subnetwork is one of the significant advantages of this design.

Note that management functionality does not have to be user initiated. In this architecture, management services could run in real-time. By using any decision system available, the distributed management application may make adjustments to the devices as problems develop.

6.2.3 Evaluation

Though preliminary, this model provides potential solutions for all network management requirements proposed in section 3.4. The distributed network management service model on top of CORBA provides the scalability required for large networks. The ability to use management services tailored to particular environments when appropriate provides adaptability. The exact nature and design of such services will depend on the heterogeneous nature of the region to be managed, and may require the creation of a completely new service. It is likely that the automation of service creation will be a priority in later stages of this project.

The utility of the proposed management system is limited by the services created within the architecture. With the use of JINI-like service brokering mechanisms, it is believed that auto-discovery of new devices, auto-configuration, and heuristic-based auto-management will be possible within the architecture.

The problem of how to collaborate with external network management systems has yet to be resolved. Without standard interfaces and communication protocols no collaboration is possible. The path ahead for this area may lie in policy based management. As this field becomes more mainstream, a policy service could be added to the architecture described here to interface with the policy servers of other network management systems. By dynamically aligning the policies of the management systems involved, collaborative management should be possible.

6.2.4 Current State

A prototype based on this design is currently being developed. A simple client interface provides monitoring of local and remote networks. Presently, distributed applications are under development to provide simplified OSI based functionality. APIs for clients to access these Services are under development. The first services are envisaged to automatically handle faults and reconfigure devices for improved network performance.

To summarize, this architecture provides a distributed and device independent network management system. SNMP provides ubiquitous access to monitor and control the network devices. The JIDM gateway provides the necessary IDL conversion from CORBA method invocations to SNMP protocol operations so that management applications can transparently access local and remote SNMP agents. The CORBA system is responsible for coordinating the distribution and caching of management information and potentially the management policies for intermediate service providers. Management interfaces may be built on top of the CORBA services, which provides an abstracted view of the underlying network.

7. OTHER ISSUES

Several additional issues related to distributed network management should be mentioned. This section reviews three issues that were not appropriate elsewhere in this report. The first issue relates to data acquisition by management systems. Adaptive polling is one of several schemes used to balance the needs of low network traffic and the timeliness of device information. The second issue provides an overview of inter-agent communication. While peer-to-peer management systems use manager-to-manager protocols and distributed object systems use an ORB, agents have no standardised method of exchanging information. Finally, while the original version of SNMP enjoys high popularity several problems, including what can be called holes, underline the need for new standardised solutions.

7.1 Polling vs. Trap Based Systems

When determining how to gather statistics about the network, it is necessary to determine the method of communication that will occur between the devices and the NMs. There are two possibilities. Either the device's agent informs their managers of their state, or the NM makes a similar request of the device.

The agent-based or "active" approach has several problems associated with it. The manager must be located in advance of any communication, for one. Problems also arise in what to communicate with the manager. Sending the complete state at regular intervals may cause the network or NM to become overloaded. The advantage of the active approach is that managers can receive timely information about critical events. Unfortunately a failure of the device is a critical event in which a message can not be sent.

Polling of the devices by NMs is by far the more common approach. The NM can discover the location of devices using ICMP messages and then query devices directly. This solves the location problem. The load on the network and the NM can be reduced since the manager can determine what information it needs at a specific time, and request only that data. The disadvantage is that it may take some time for critical information to reach the NM as it takes on average one half of the polling interval to retrieve an update from a device. This can be especially problematic if a device has failed, since a missing response could also indicate a network fault. The question of when and how often to poll is still a subject of research.

The most common solution is to poll a device again if the number of outstanding polls does not exceed *N*, typically a number between 3 to 5. This is known as COP-N retransmission [39]. When the first poll is not acknowledged within 10 seconds, the poll is retransmitted. Subsequent polls follow at double the previous interval until the *Nth* poll is sent. At that point, the NM considers the device to be inoperable.

The choice of the number of outstanding polls *N* can have a grave impact on the operation of the management system. A high number may cause burstiness as the NM continuously attempts to reach devices. A low number can prevent this congestion, but may result in incorrect assumptions of inoperability.

A second alternative is to allow the polling interval to adapt to the current state of the network and the level of interest the NM has in the data being requested. As proposed in [40], a specific polling rate can be maintained by adapting to congestion and timeout values from the network. Rate Adaptive Polling (RAP) detects congestion making it easier to tell if a poll has been lost on the network or the device is down.

7.2 WAN Management

As discussed in Section 3.1.4, the Wide Area Networks (WANs) of an organisation may span large distances and be connected by one or more independent Service Providers (SPs). The organisation negotiatea with the SPs to come up with Service Level Agreements (SLAs) that describe the quality of connectivity to be provided. These agreements can be multi-lateral and quite complex since many SPs may be needed to connect the various subnetwork units, and the organisations backbone network may in turn be providing connectivity for other clients.

Though the concept of SLAs is widely used, the measurement of the interactions between, and the enforcement of, SLAs are not well understood. It is, however, vital for the success of WAN management to do all three. Intermediate SPs must react in well-defined ways, especially in the case of prioritised or QoS sensitive communication streams.

There are many issues involved in creating an appropriate management model for networks that include independent SPs. How much information can you expect from the SP? In a federated military deployment you may expect almost complete operational information without operational control. In a commercial environment, operational details may be considered proprietary and almost no information will be available. In the former case, information may be limited by the access rights of the user. In the latter case, information may be something that is bought. In any case, security and access is something that must be closely monitored.

One of the big questions is how to collaborate with SPs to solve common problems. How does an operational entity solve end-to-end problems when it has no control over portions of the interconnected networks, and, at worst, only partial monitoring capabilities? One possibility is the use of dynamic policies, where SPs are left to solve their own problems and need only provide an interface at which guarantees can be negotiated.

Policy-based systems in the simplest sense are action-reaction pairs. If some condition arises, the following action must be carried out. Policy systems are becoming more popular in network management as a method to automate repetitive tasks. Recycle the network logs at 2AM, do a complete network mapping at 3AM, if the main DNS server goes down, start up the clone, etc.

Policy systems can become quite complex as resolution systems must decide which conflicting operation (if any) is to be carried out. Similarly, chains of operations may lead to failure conditions that are not obvious from when the policies were initially entered. Changes to existing policy systems can also be fraught with danger.

For WAN management, policies can be used to co-ordinate resources for best use during times of partial failure. Consider a high priority connection that is traversing a single SP between two organisation subnetworks. It is determined that the connection is not getting the level of service specified in the SLA. Through communication between the various policy systems of the organisation and the SP, it is determined that the SP is experiencing a problem that does not allow it to provide the required service at that time. The organisation can then re-route the priority stream to another SP if one is available, or attempt to negotiate a temporary SLA with the failing SP to use as much connectivity as is currently available. In any case, the failure to provide the QoS defined by the SLA should have pre-defined consequences for the SP.

7.3 Agent Communication

One of the main advantages that agent-based management systems have over more static systems is their ability to co-operate and exchange information. The question, however, is how to achieve the expressiveness required to communicate the goals of one agent to another in a compact format. The more complex the agent system, the more complex the language required. For truly intelligent agents, it is possible to talk about the expression of desires, fears, and other less tangible directives on which other agents may or may not act.

In order to support this communication, some method for transferring information is required. The most common approach used by several systems, for example [41], is to create a common blackboard upon which agents may place and retrieve information. In this way, agents can retrieve and post the completion of tasks, and otherwise express the state of the network.

Depending on the particular implementation of the blackboard system, several problems may arise. Blackboards are notoriously wasteful of resources. In order to store, search, and modify potentially large data spaces, agent environments become even more unwieldy as a consequence of all the additional services that are required. Also, since no standard system exists, agents from one system will not understand the messages from another system.

An alternate method is to use direct agent-to-agent communication, possibly routed via facilitators that match messages with those who may be interested. KQML [42] is an abstract specification that comes from the AI community. It includes typed messages that facilitate interpretation of the messages themselves. A typed message can include information about how to reply, and what it wants done (*performative*). These performatives can be used to query (e.g. ask-if) or reply to (e.g. tell) an agent. There are also methods for information exchange (e.g. deny), information transfer (e.g. insert), and task matching (e.g. advertise).

7.4 SNMP "Holes"

SNMP has often been sited in this report as a well designed ubiquitous protocol. It is widely implemented, deployed, and well understood (at least in its original form as SNMPv1.) Since the protocol is well defined, management applications are built to use the information SNMP provides.

There is however a fairly serious problem which makes SNMP less effective than it could otherwise be. The problem arises initially from the lack of a set of standard metrics for all network devices. Since networking itself is not well understood, no common set of metrics can be decided upon for inclusion in any device agent. Should an agent be monitoring the number of packet errors per second, per thousand messages, or the total errors over its latest operational cycle? Or could all these metrics be useful in different instances?

The lack of consensus means that the definition of network metrics to remain open for interpretation. Unfortunately, vendors use this ambiguity to defend "proprietary" metrics. When a new device is created a new SNMP MIB for that particular device and implementation is created along with a special SNMP management application which uses those metrics.

These MIB-extensions cause problems for SNMP management applications that attempt to be more global in scope. Such applications either need to know about all the existing (and future) devices and their MIB extensions, or an information gap will exist.

7.5 Self Managing Applications

A tacit assumption in network management is that applications demand a certain QoS from the underlying network that does not change. What happens when that QoS is not met is not well defined, and can range from complete failure to slight decreases in utility. This assumption is not strictly correct. Just as applications can vary in their consumption of other resources, an application can, if properly informed, vary their network consumption to make best use of the QoS that is currently available.

The work of Kunz and Black [43] proposes the use of a proxy which acts as an intermediary between a high QoS and low and variable QoS portion of the network such as that provided by wireless connections. By placing part of the application logic on the proxy, the communication stream between the proxy and the part of the application on the wireless portion of the network can be tuned to make use of the QoS available. In times of sufficient QoS, the communication between the client and sever portions of the application pass through the proxy unchanged. In times of low QoS, the proxy can intercept the communication stream and alter the data in such a way that the client side can still make use of the reduced communication with minimal operational degradation.

Applications themselves can thus alter their operation to make use of the network resources available, raising network management to the application level.

8. CONCLUSIONS

The current state of the art in distributed network management is an un-standardised mix of proprietary software that supports a single manager with auxiliary managers spread throughout the network. The first version of SNMP is used almost exclusively to contact devices and gather performance statistics, as well as to provide for some limited control of their operation.

This corresponds with the precursor of the peer-to-peer model previously described. Despite its distributed title, all information is processed at a single management station for presentation to the operator. In order to cut down on the network and processing bottleneck, services such as those being developed by the DISMAN working group need to be standardised. Without the scalability and security such distributed systems can provide, there will be little market for this direction in the future.

Distributed object computing is enjoying a rise in popularity, and one of the domains it's being applied to is DNM. While there are still many issues to be worked out, there is concerted effort being put into a DOC solution. Network management lends itself well to objectification, where devices all have similar interfaces, but very different internal implementations. By using CORBA services and facilities, distribution of management utility becomes inherent in the solution.

While agents provide an interesting abstraction for network management, much work needs to be done on the basic infrastructure, and on understanding of agent systems before it can adopted as the solution of choice. Issues of non-determinism and security will need to be addressed. It is predicted that agent technology will mature in ten to fifteen years, at which point network management will become an appropriate domain in which agents could be used.

REFERENCES

[1] William Stallings, SNMP, SNMPv2, SNMPv3, and RMON 1 and 2. Addison-Wesley Press, 1999.

[2] Jűrgen Schőenwälder and Aiko Pras, *SNMP Standards Summary.* Simple Times <http://www.simple-times.org/> Vol. 7, Num. 1, March 1999.

[3] ISO/IEC 7498-4:1989, *Information Processing Systems - Open Systems Interconnection - Basic Reference Model - Part 4: Management Framework.*

[4] W. R. Stevens, *TCP/IP Illustrated Volume 1.* Addison-Wesley Publishing, 1994.

[5] AUSCANNZUKUS Naval C4 JWID AdHoc Working Group, *Multi-national Naval Task Group (MNTG) Final Report JWID99-R.* September 1999.

[6] Hobbes' Internet Timeline *Website* <http://www.isoc.org/guest/zakon/Internet/History/>

[7] Akhil Sahai and Christine Morin, *Towards Distributed and Dynamic Network Management.* Proceedings of the 1998 IEEE Network Operations and Management Symposium, Volume 2. February 1998, Pages 455-464.

[8] G. Goldszmidt and Yechiam Yemini, *Distributed Management by Delegation.* Proceedings of the 15th International Conference on Distributed Computing Systems, June 1995.

[9] Netrix Corporation, *Distributed SNMP Management White Paper.* <http://www.netrix.com/products/wpprs/snmp/snmp.htm>

[10] M. Leppinen, P. Pulkkinen and A. Rautiainen, *Java- and CORBA-Based Network Management.* IEEE Computer, Vol. 30, No. 6, June 1997. Pgs 83-87

[11] A. Bieszczad, B. Pagurek and T. White, *Mobile Agents for Network Management.* IEEE Communication Surveys <http://www.comsoc.org/pubs/surveys/>, Fourth Quarter 1998.

[12] G. Susilo, A. Bieszczad, and B. Pagurek, *Infrastructure for Advanced Network Management based on Mobile Code.* Proceedings of the Network Operations and Management Symposium (NOMS '98), Boston, MA, April 1998. Pgs 322-333.

[13] OMG, *The Common Object Request Broker; Architecture and Specification (v2.3.1).* OMG document formal/99-10-07, Oct 1999.

[14] IETF *Web Presence*: <http://www.ietf.org/>

[15] G.H. Aicklen and P.M. Main, *Remote Control of Diverse Network Elements using SNMP.* Proceedings of MILCOM 1995 Vol 2, San Diego, Nov 6-8, 1995. Pgs 673-677.

[16] WBEM *Web Presence*: <http://www.dmtf.org/wbem/>

[17] DMTF *Web Presence*: <http://www.dmtf.org/>

[18] DMTF, *Common Information Model (CIM); Core Model White Paper.* Aug 1998.

[19] DMTF, *XML as a Representation for Management Information; A White Paper,* <http://www.dmtf.org/spec/wbem.html> Oct 1998.

[20] BMC, *Making WMI Work for You*. From the 1999 DMTF Conference Presentations, <http://www.dmtf.org/educ/conf1999/pres.html>, June 1999.

[21] Tivoli, *WBEM-Based Solution for NT*. From the 1999 DMTF Conference Presentations, <http://www.dmtf.org/educ/conf1999/pres.html>, June 1999.

[22] Charlie Kindel, *Distributed Component Object Model Protocol -- DCOM/1.0*. INTERNET-DRAFT <http://www.microsoft.com/com/tech/DCOM.asp>, Jan 1998.

[23] JIDM *Web Presence*: <http://www.jidm.org/>

[24] The Open Group, *Inter-Domain Management: Specification Translation*. TeleManagement CS342, Feb 1997.

[25] Object Management Group, *CORBA/TNM Internetworking: JIDM Interaction Translation*. White Paper (telecom/98-10-10), Nov 1998

[26] Object Management Group Telecom Task Force, *CORBA-Based Telecommunication Network Management System*. White Paper (Revision 96-07-01), May 1996.

[27] S. Rahkila and S. Stenberg. *Experiences on Integration of Network Management and a Distributed Computing Platform*. Proceedings of the 30th Hawaii International Conference on System Sciences (HICSS-30) Vol. 1, Maui, HI, January 7-10, 1997.

[28] Sun Microsystems, *The Jini™ Architecture Specification Version 1.0.1*. <http://www.sun.com/jini/specs/>, Nov 1999.

[29] Tim Finin, Jay Weber et al, DRAFT Specification of the KQML Agent-Communication Language. DARPA Knowledge Sharing Effort <http://www.cs.umbc.edu/kqml/>, June 1993.

[30] FIPA *Web Presence*: <http://www.fipa.org/>

[31] FIPA, *FIPA 97 Specification – Part 7 – Network Management and Provisioning*. <http://www.fipa.org/spec/FIPA97.html>, Geneva, Switzerland, 1997.

[32] OMG Document ec/99-03-12, *OMG - FIPA Liaison*. <http://www.objs.com/isig/omg-fipa-liaison-4.html>, March 1999.

[33] Mitsuru Oshima, Guenter Karjoth, and Kouichi Ono, *Aglets Specification 1.1 Draft*. <http://www.trl.ibm.co.jp/aglets/documentation.html> Sep 1996.

[34] Aglets Open Source, *Web Presence* <http://www.aglets.org/>

[35] Mitsubishi Electric ITA - Horizon Systems Laboratory, *Mobile Agent Computing*. A White Paper, January 1998.

[36] Alberto Castillo, Masataka Kawaguchi, Noemi Paciorek, and David Wong, *Concordia™ as Enabling Technology for Cooperative Information Gathering*. Proceedings of the Japanese Society for Artificial Intelligence Conference, Tokyo, Japan June 17-18, 1998.

[37] IKV++ GmbH, *Grasshopper – A Platform for Mobile Software Agents*. A White Paper, 1999

[38] Grasshopper *Web Presence* <http://www.grasshopper.de/>

[39] A. B. Bondi, *A Nonblocking Mechanism for Regulating the Transmission of Network Management Polls.* Proceedings of Integrated Network Management V, May 1997, Pages 565-580.

[40] Pratyush Mogheé and Michael H. Evangelista, *RAP - Rate Adaptive Polling for Network Management Applications*. Proceedings of the 1998 IEEE Network Operations and Management Symposium, Volume 2, February 1998, Pages 395-399.

[41] Gatot Susilo, Andrzej Bieszczad and Bernard Pagurek, *Infrastructure for Advanced Network Management based on Mobile Code*. Proceedings of the 1998 IEEE Network Operations and Management Symposium, Volume 2, February 1998, Pages 322-333.

[42] Tim Finin, Jay Weber et al, *DRAFT Specification of the KQML Agent-Communication Language*. The DARPA Knowledge Sharing Initiative External Interfaces Working Group, June 1993.

[43] T. Kunz and J. Black, *An Architecture for Adaptive Mobile Applications.* Proceedings of the 11th International Conference on Wireless Communications, Calgary, Alberta, Canada, July 1999. Pages 27-38

DOCUMENT CONTROL DATA

(Security classification of title, body of abstract and indexing annotation must be entered when the overall document is classified)

1. ORIGINATOR (the name and address of the organization preparing the document. Organizations for whom the document was prepared, e.g. Establishment sponsoring a contractor's report, or tasking agency, are entered in section 8.)

COMMUNICATION RESEARCH CENTRE CANADA
3701 CARLING AVENUE, PO BOX 11490 STATION H
OTTAWA, ONTARIO, CANADA K2H 8S2

2. SECURITY CLASSIFICATION (overall security classification of the document, including special warning terms if applicable)

UNCLASSIFIED

3. TITLE (the complete document title as indicated on the title page. Its classification should be indicated by the appropriate abbreviation (S,C or U) in parentheses after the title.)

DISTRIBUTED NETWORK MANAGEMENT (U)

4. AUTHORS (Last name, first name, middle initial)

KIDSTON, DAVID

5. DATE OF PUBLICATION (month and year of publication of document)

OCTOBER 2000

6a. NO. OF PAGES (total containing information. Include Annexes, Appendices, etc.)

x+45

6b. NO. OF REFS (total cited in document)

43

7. DESCRIPTIVE NOTES (the category of the document, e.g. technical report, technical note or memorandum. If appropriate, enter the type of report, e.g. interim, progress, summary, annual or final. Give the inclusive dates when a specific reporting period is covered.)

TECHNICAL REPORT

8. SPONSORING ACTIVITY (the name of the department project office or laboratory sponsoring the research and development. Include the address.)

DEFENCE RESEARCH ESTABLISHMENT OTTAWA (DREO)
NATIONAL DEFENCE HEADQUARTERS, OTTAWA, ONTARIO, CANADA
K2H 8S2

9a. PROJECT OR GRANT NO. (if appropriate, the applicable research and development project or grant number under which the document was written. Please specify whether project or grant)

5cb12

9b. CONTRACT NO. (if appropriate, the applicable number under which the document was written)

10a. ORIGINATOR'S DOCUMENT NUMBER (the official document number by which the document is identified by the originating activity. This number must be unique to this document.)

CRC-RP-2000-10

10b. OTHER DOCUMENT NOS. (Any other numbers which may be assigned this document either by the originator or by the sponsor)

DREO TM 2000-109

11. DOCUMENT AVAILABILITY (any limitations on further dissemination of the document, other than those imposed by security classification)

(X) Unlimited distribution
( ) Distribution limited to defence departments and defence contractors; further distribution only as approved
( ) Distribution limited to defence departments and Canadian defence contractors; further distribution only as approved
( ) Distribution limited to government departments and agencies; further distribution only as approved
( ) Distribution limited to defence departments; further distribution only as approved
( ) Other (please specify):

12. DOCUMENT ANNOUNCEMENT (any limitation to the bibliographic announcement of this document. This will normally correspond to the Document Availability (11). However, where further distribution (beyond the audience specified in 11) is possible, a wider announcement audience may be selected.)

UNLIMITED

13. ABSTRACT (a brief and factual summary of the document. It may also appear elsewhere in the body of the document itself. It is highly desirable that the abstract of classified documents be unclassified. Each paragraph of the abstract shall begin with an indication of the security classification of the information in the paragraph (unless the document itself is unclassified) represented as (S), (C), or (U). It is not necessary to include here abstracts in both official languages unless the text is bilingual).

This report begins with an introduction of network management and an outline of the problems facing the existing management systems for enterprise networks. Commercial and defence related issues in effective management of the next generation of application traffic are also discussed.

The following four main requirements for effective network management are extracted; management must be scalable, adaptable, useful, and collaborative. Scalability is required to handle the increasingly large sizes of contemporary networks. Also, network management systems should be able to adapt to the diversity of network components. New utility at all levels of network management is necessary to handle the traffic requirements of the new kinds of applications. For example, applications may require security and/or be QoS sensitive. Finally, for networks to co-exist and provide end-to-end management while maintaining local control, some kind of collaboration is needed.

It is postulated that a distributed management architecture is best suited to satisfy these requirements. Several network management architectures are reviewed. The report gives an overview of existing network management and related technologies. The report concludes with a comparison of these technologies, a coalition network management proposal, and a discussion of several related issues.

14. KEYWORDS, DESCRIPTORS or IDENTIFIERS (technically meaningful terms or short phrases that characterize a document and could be helpful in cataloguing the document. They should be selected so that no security classification is required. Identifiers such as equipment model designation, trade name, military project code name, geographic location may also be included. If possible keywords should be selected from a published thesaurus. e.g. Thesaurus of Engineering and Scientific Terms (TEST) and that thesaurus-identified. If it is not possible to select indexing terms which are Unclassified, the classification of each should be indicated as with the title.)

Network Management
Enterprise Management
Distributed Systems
JWID99-R
SNMP
RMON
DISMAN
TCP/IP
WBEM
JIDM
CORBA
JINI
Intelligent Agents