第 8 部

ネットワーク管理

第 1 章

Prologue

1.1 The NetMan working group

The NetMan working group has been active in the area of network management. The aim of this working group is to make significant contribution in the areas of network management in particular and the Internet in general.

1.1.1 Activities during the year 1992-93

In the year 1992-93 we have mainly concentrated on the issue of examining the required and available information components for network management. The requirement analysis is based on the perceived use of the information. Having identified the network infrastructure as an important information component we have proceeded to develop the framework to service this information requirement. Experimental protocols [98] [99] have been developed and experimentation is presently underway on an international scale. We have investigated areas in the Internet which require managerial attention. As a case study, we have examined the non-optimality of FTP-traffic in the Japanese portion of the Internet. The non-optimality detected does call for attention. We have investigated short-term and long term solutions. Finally, we have taken up the issue of managing network applications. **MIB**s have been developed [100] and implemented.

1.1.2 Planned activities for the year 1994-95

In the year 1993-94 we plan to work on the following

- deployment strategy for X.500 in the Internet. This involves populating the directory with information from the Internet with minimal human intervention.
- development a framework for *smart applications*. *Smart applications* are those which use network information to operate efficiently and optimally as far as network access and generated traffic is concerned.

- development of tools for cartography in the network. Tools that will (*semi-*) automatically generate network maps using information from the network.
- deployment of an overall management framework for a largescale network to evaluate the tools and algorithms that have been developed.

1.2 Outline of the document

In chapter 2 we have examined the information requirements of network management systems. We have enriched the information with configuration/ policy ... related details. Based on this information, we have designed algorithms that can make decisions for efficient network management.

For efficient network management it is essential that the directory services be used as a source of information for components that are static or quasi-static in nature. The possible uses of such information components are discussed and their potential uses are shown. The corresponding areas where work needs to be done are described in chapter 3.

File retrieval constitutes a major component of the traffic in the Internet. Several servers offer the same file. This redundancy gives rise to a lot of avoidable non-optimality. A study has been carried out to examine and estimate this non-optimality. The results are presented and methods of improved user support based on knowledge of the network topology for optimising the ftp-traffic are discussed in chapter 4.

To date, network management systems have focussed on the lower three layers [Interface, IP, Transport] of the Internet suite of Protocols. The standard **MIB**-II covers these layers well. Further, more specific **MIB**s have been developed for particular protocols in each of these layers. However, with the maturation of network management systems , the need for managing network applications is increasing. The development in this area has become more interesting with the fact that there are a host of OSI applications [X.400, X.500, ..] running over the TCP/IP stack. As a result, the application management protocols will need to address both OSI and Internet applications. Ongoing work in this area is described in chapter 5.

Chapter 6 details the work that has been done towards putting NIC information in the directory. We have worked on the JPNIC [Japan Network Information Center] and APNIC [Asia Pacific Network Information Center] databases. The experience and problems encountered are described.

第 2 章

Information, Intelligence & Algorithms for Efficient Network Management

2.1 Introduction

Any management basically involves gathering information about the subject that needs to be managed. The gathered information is then processed and analyzed using the knowledge about the subject, to make judgements about the status of the subject and to take decisions about the future course of action.

Traditional network management systems which have concentrated mainly on monitoring network traffic, performance and fault detection have focussed on the operations and functions of the network elements and their interfaces. In the Internet there are several **MIB**s that are proposed standards or are under development- almost all relate to protocol layers 1-4.

However for efficient network management, it is important to *know* about higher level information e.g. the functioning of the applications and the network configuration, administration and policy.

Given the appropriate information intelligent management applications can be created by developing algorithms that model an expert network managers approach to solving a particular problem.

In the rest of this chapter a *network* is the media for transmitting information. *network elements* are equipment with one or more *network interfaces* whereby it is possible to exchange information with the network. *Network elements* with multiple interfaces e.g. gateways/routers/bridges/repeaters... may be used to connect networks.

2.2 A Model for Intelligent Network Management

Most of the work in Network Management, so far, has been done in building the framework for collecting information from the network. Presently, there is an agreed management framework. Agents residing in network elements will provide the network manager

application with information about the network. Standards have been fixed for communicating management information between the manager and the agent. The objects which need to be managed, are listed up and described in terms of their properties in **MIB**s. In this framework, the human managers' view of the network is limited to objects in the Management Information Base(**MIB**). Of course extensive research has been carried to develop these **MIB**s and the development and refinement is an ongoing process.

In the following discussion we will use the example of the SNMP [101], [102], [103] network management framework; it is possibly the most widely deployed and experimented.

2.3 Information Components of a Network Management System

The Information Components of a network management system are as follows:

- Communication operations related
  - System
  - Lower Layers
  - Network Layer
  - Transport Layer
  - Application Layer
- Infrastructure related
- Administrative & Policy related

In the following we briefly describe the issues related to various sub-components.

2.4 System, Lower, Network & Transport

These components are possibly the most widely explored and discussed and deployed. The **MIB**-II [103] contains the core set of managed objects for the for the System, Lower, Network & Transport layers of the Internet suite of protocols. The Host Resources **MIB** [104] has been developed. It covers the system and process related details of network elements. There are several other **MIB**s which deal with a specific type of interface or transport medium or protocol. Together these **MIB**s allow the monitoring and management of the communication infrastraucture - the lines nodes and the communication protocol entities. They give an immediate idea of the functioning of the network and are widely used for network monitoring, traffic and performance estimation, fault-detection, etc. .

2.5 Application

With the basic management framework reasonably well established and with the growing spread of network applications like mail, news, DNS, Directory Services, ... , the issue of managing Network Service Applications[NSA] is gathering attention. NSAs include all applications that provide network services. The necessity of managing Network Service Applications is particularly critical for NSAs which have (widely) distributed components such as the Mail and Directory Services.

The targeted NSA management centers around general monitoring

- to detect a broad range of operational problems: broken connectivity, system failure, congestion, ..
- to monitor performance and utilization

Presently a generic NSA-MIB [105] which covers the general network service related aspects has been defined. It focusses on the NSA description, the operations and activity indications, the associations and association related details and two application specific **MIB**s for mail [106] and directory [100] applications which focus on the respective application-specific aspects have been defined to work in conjunction with the NSA-MIB. Detail discussion on the **MIB**s and their development follows in chapter 5.

2.6 Network Infrastraucture

The widening span of computer networking has highlighted the importance of holding and servicing information about the networking infrastructure itself. The growing and active interest in network management [107], is severely constrained by the lack of any organized pool of information about the network infrastructure itself. Some attempts have been made, on a piecemeal basis, to provide a larger view of some particular aspect of the network (WHOIS, DNS, .. in the case of the Internet; [108], [95]). Presently, there is a movement to explore the possibility of setting up a framework to hold and serve the infrastructural information of a network [109].

2.7 Infrastructural information requirements

Network operation and management requires information about the structure of the network, the nodes, links and their properties & functions. Further, with current networks extending literally beyond bounds, the scope of the information covers networks beyond the span of local domain of authority or administration. When the Network was relatively small and simple the map was already known to the knowledgable network administrator.

Based on this knowledge the course of the packets to different destinations would be charted. But presently the size of the Network is already beyond such usages, with the current growth being near explosive. This is giving rise to the urgent necessity of having infrastructural and service related information made accessible from all places and at all times in a reasonably efficient manner and with reasonable accuracy.

Network related information, referred to as 'network map' in the rest of this paper, should

1. Show the interconnection between the various network elements. This will basically represent the Network as a graph where vertices represent objects like gateways/workstations/ subnetworks and edges indicate the connections.
2. Show properties and functions of the various network elements and the interconnections. Attributes of vertices will represent various properties of the objects e.g. speed, charge, protocol, OS, etc. Functions include services offered by a network element.
3. Contain various name and address information of the networks and network elements.
4. Contain information about various administrative and management details related to the networks and network elements.
5. Contain the policy related information, part of which may be private while the other part may be made public.

Using this map the following services may be provided

1. Configuration management:
   - Display the physical configuration of a network, i.e. nodes and their physical interconnections
   - Display the logical configuration of a network, i.e. nodes and their logical interconnections.
2. Route management:
   - Find alternate routes by referring to the physical and logical configurations.
   - Generate routing tables considering local policy and policy of transit domains.
   - Check routing tables for routing loops, non- optimality, incorrect paths, etc.
3. Fault management: In case of network failures alternatives may be found and used to bypass the problem node or link.
4. Service management: Locate various services and servers in the Network.

5. Optimization: The information available can be used to carry out various optimizations, for example cost, traffic, response-time, etc.
6. Provide mappings between the various names and addresses of elements
7. Depict administrative/autonomous domains.
8. Network Administration and Management: References to people responsible for administering and technically maintaining a network will be useful.

Examples of such usages are described in [98], [110], [111].

2.8 Algorithms

As can be expected a whole range of applications can be developed based on the information components provided. Simple monitoring of the network for fault detection, generating *annotated network maps*, making routing decisions, finding alternate paths in cases of failures or congestion, finding the *cheapest* source of a document are just some examples.

In this section we examine the various algorithms that may be used in some of these applications for efficient management.

2.8.1 Monitoring Algorithm

One of the purposes of network monitoring is to detect any abnormality or fault. This is the simplest of algorithms which involves the periodic checking of the value a managed object or a set of managed objects. The detection of abnormality or fault depends on the type and definition of the managed object. For example, for a *Status* type object, if the value is a direct indication. For *Gauge* or *Counter* type objects there will be a reference value or threshold which decides whether there is an abnormality or not. E.g. the presence of error packets beyond a certain critical count would indicate a definite abnormality.

2.8.2 Network Maps

A map is very important for management purposes - irrespective of whether the manager is a human being or a application process. Given the configuration information is a straight forward generate the topological graph of the network where the vertices represent the components of the network and edges indicate the connections. For visual representation the graph may be translated to a more "physical" illustration.

Just as there are several maps of the same geographical domain (political, natural...) one can envisage several views of the same network and its components. A view (called

"image" in the remainder) could pertain to a particular protocol suite (IP/OSI/...), an administrative domain or purpose. Using images, several abstractions of the same object is possible. For more details refer to [99].

2.8.3 SoftPages[SPP]

It is desirable to query for files always on filestores that are nearest/cheapest to the users' site. For that purpose, it is necessary to evaluate distance/cost between the users' host and possible servers. Also, it is necessary to search the contents of the servers. For example, a user on host *A* looking for a certain document should under ideal situation carry out the search in the ordered list of fileservers *[C,B,D]*, the ordering being based on the potential retrieval costs.

The Soft Pages Project [110] which implements this algorithm started with the wish to reduce ftp traffic on crowded overseas links and national backbones. The algorithm makes use of the network infrastructure information concerning network configuration, location of servers and their contents, the links and their charges.

The algorithm makes use of the network infrastructure information concerning network configuration, location of servers and their contents, the links and their charges.

Parameter values are read from the network database and represent:

speed - the bandwidth (in kbps) theoretically available in a network

traffic - the average use (as percentage) of this network, thus telling something about congestion

charge - monetary units (as abstract integer number) to be paid for the transmission of one packet; this should be used to express relationship between several charges rather than absolut amount of money (not to speak of currency, exchange rates, etc.)

priority - sometimes system administrators want to keep certain connections free for important traffic like mails. They can do this by increasing the priority (integer value) of the network connection in comparison to another.

Cost is calculated as follows:

$$cost = f(speed, traffic, charge, priority)$$
$$cost = a * (1/speed) + b * traffic + c * charge + d * priority$$

whereas the weights a, b, c, d can be chosen freely by the administrator of a site. This way an individual evaluation is possible. For some sites speed has a higher weight than charge, for others it may be the other way around.

With the formula given above, a cost index can be calculated for one network connection. If traffic has to pass several networks, their cost indexes will be added to an overall cost index for the end-to-end connection. Thus, the number of network hops goes into the cost index, too.

2.8.4 Alternate Paths

Given the mesh-type connectivity of networks, it is very important to have a clear picture about the connections; for example, A-net has potential connectivity to D-net via multiple routes. The topological map of the network is useful; to find alternate paths in case of failure, to know the transit policy of the provider on the alternate path, to know the point of contact for the provider. Needless to say the map can be used to detect non-optimal routes and to evaluate the quality of the connectivity in terms of redundancy.

2.8.5 Fault Tolerant Management

In the present management scheme a manager monitors networks through the manager workstation. Now if the link from the manager workstation to the agent is down or still worse if the link to the managed network is down without alternate means there will be a collapse of network management. Among the alternate means are the following

- **Use Alternate paths for management traffic**

  This implies that the management application has control over the networking layer. It uses the network configuration information to decide the next alternative path/router and instructs the networking layer to route its requests/queries accordingly. The intelligent agent at the other end reads the request and sends the reply along the appropriate route.

- **Use Proxy Agents**

  This method is a more simpler method but involves the cooperation of a few more agents in the network. For example suppose the the link between management stattion A and Managed Station B is down. Now, if the link between A and some station C is up, and if there is a link between C and B, then if there is an agent on C which agrees to act as a proxy for the agent on B then the management station can redirect its query to the proxy agent on C.

2.9 Conclusion

As part of an effort to cope with the rapidly changing communications scene and the explosive growth in communication networks, the need for a framework to hold the infrastructural and service related information about communication networks has been emphasised. The use of such a *map* is manifold. A Pilot based on this idea, presently covering the Japanese Internet, is in operation. Future plans involve extending the pilot to the International arena to cover other countries/NICs. The network model adopted in the Pilot for representing a communication network with all its related details and descriptions in the Directory, is described. Experience with problems and strategies for bootstrapping, operation and maintenance of the Pilot are discussed. A new genre of applications based on this proposal are coming up and their current status is reviewed.

第 3 章

Using the X.500 Directory for Efficient Network Management

3.1 Introduction

Management basically involves gathering information, processing it in an appropriate manner to derive some conclusions and, carrying out appropriate action based on the inferences. Network managemnent is no exception. The nature of the information utilized by network management can be broadly classified as

- dynamic. The information that (potentially) changes frequently, e.g., the operational status of an interface, the cumulative number of error packets at a particular receiver etc..
- quasi-static. Information that changes - but not with high frequency - i.e. not on an hourly or daily basis. E.g. Network configuration.
- static. Information that changes very infrequently - for example the ip-address of a host, the manager of a network, ...

The information components required for network management have been outlined in chapter 2. The dynamic components are in general serviced by querying some agent which has access to the latest information *e.g.* link status or the number of active connections. Some of the static/quasi-static components and the problems involved in holding and servicing these components are discussed in the following.

- **Network Maps:**
  Like any other map, the utility of a map detailing the network configuration is tremendous. Yet, implementing and maintaining a detailed map of the network poses a serious problem. The scope of the map is global and the network itself is expanding. Some of the problems that are peculiar to the network map are listed below:

- The Network configuration is quasi-static. Nodes, links and networks are being added,updated and deleted someplace or the other.
- The Network is huge and geographically distributed.
- The Network spans several political and administrative areas. The related information is also controlled and maintained in a distributed fashion.

In short, global network configuration information is unwieldy and growing continuously. It is impossible to service such information in a centralized fashion. So, a distributed database system is necessary. In this context, the X.500 Directory system [112] is an appropriate candidate on which the network map may be implemented. The X.500 Directory is intended to be a very large and highly distributed database. It is structured hierarchically with entries arranged in the form of a tree in which each object corresponds to a node or an entry. Information is stored about an object as a set of attributes.

- **NIC-information:**
  The information services provided by the various NICs scattered all over the network is essentailly quasi-static. The information is in generally administered locally while the information may be sought and used from any part of the network. It is clear that a service based on X.500 will be much more powerful due to the global and distributed database providing a single window whois-type service.

It is beyond doubt that the X.500 Directory provides an excellent framework for storing and servicing information which is of static or quasi-static nature, in a distributed fashion. Pilot projects employing the directory are already in place. The PARADISE [113] project presently spans 35 countries including Japan and covers more than a million entries. Thus the directory services, if properly utilized, could be a repertory of useful information for network management.

Though there has been extensive work on the Directory system, it has mainly been concentrated on persons, organizations, ..., there has been little or no work on the framework for representing networks in the Directory. Only recently the situation has started changing as is described in the following.

3.2 Present Status of research and experimentation

A major challenge in employing the X.500 directory for novel applications and purposes is to design appropriate schemas. Schemas related to the applications mentioned above are listed below along with the present status.

- A model in which a communication network with all its related details and descriptions can be represented in the X.500 Directory is defined in [99]. Schemas of objects

and their attributes which may be used for this purpose are presented. The model envisages physical objects and several logical abstractions of the physical objects.

- The objects necessary to include information about IP networks and IP numbers in the X.500 Directory are defined in [98]. It extends the work "Charting networks in the X.500 Directory" [99] where a general framework is presented for representing networks in the Directory by applying it to IP networks. This application of the Directory is intended to support the work of IP network assigning authorities, NICs, as well as other applications looking for a mapping of IP numbers to data of related networks. Furthermore, Autonomous Systems and related routing policy information can be represented in the Directory along with their relationship to networks and organizations.

- A mapping of domains onto X.500 is proposed in [114]. It gives a range of new management and user facilities over and above those currently available. This specification proposes an experimental new mechanism to access and manage domain information on the Internet. This document is slated for revision in the light of the new experimental protocols [99] [98] that have come up.

3.3 Potential areas for research and experimentation

- Configuration management:
  - Display the physical configuration of a network, i.e. nodes and their physical interconnections
  - Display the logical configuration of a network, i.e. nodes and their logical interconnections.
- Route management:
  - Find alternate routes by referring to the physical and logical configurations.
  - Generate routing tables considering local policy and policy of transit domains.
  - Check routing tables for routing loops, non- optimality, incorrect paths, etc.
- Fault management: In case of network failures alternatives may be found and used to bypass the problem node or link.
- Service management: Locate various services and servers in the Network.
- Optimization: The information available can be used to carry out various optimizations, for example cost, traffic, response-time, etc.
- Provide mappings between the various names and addresses of elements

  - Depict administrative/autonomous domains.
  - Network Administration and Management: References to people responsible for administering and technically maintaining a network will be useful.

- Extensive deployment of the directory and network related info.
- WHOIS info in the Directory - multilingual scheme.
  Extensions to incorporate multilingualism supported by some NICs needs to be urgently designed.
- NIC info in the directory
- Use of network configuration information for smarter applications:
  - Smart Archie
  - SoftPages
  - WWFS
  - Other applications
- Assistance in configuring Applications
  Several applications need to have (*application-specific*) configuration info during operation/installation. Servicing this requirement by using the X.500 directory is a promising area. It basically envisages storing and servicing
  - NTP maps in the directory.
  - NNTP maps in the directory.
  - other maps in the directory.

3.4 Challenges

There are three major areas of challenge in realizing the proposal for using the directory for efficient network management :

1. Schema design and Naming Context: The issues that need to be taken care of in designing the schemas and naming contexts are as follows:
   - Scalability, Distribution of control & maintenance
   - Preservation of administrative boundaries/controls.
   - Minimize data duplication
   - Evolving technologies/problems
   - Security and access control

2. A framework for dissemination of Schema information: A methodology for using the Directory itself for disseminating schema informatio n.

3. Bootstrapping, Operation & Maintenance: The strategy for startup, operation and maintenance of the proposed directory services is a the most crucial aspect that will decide the success of the proposal.

3.5 The Contribution

The proposed framework will have far reaching consequences in the networking arena. The spread of networking is seeing a revolution that was beyond imagination. however researches in some of the important infrastructural aspects have not kept pace with the rapid pace of growth. To date there is no concept of cost among the majority of network users. The main reason being that the appropriate framework is non-existent. Without the global network map configuration management in the large is impossible. The management of services, an area of growing importance as the network is moving to within the reach of ordinary citizens, is impossible without a truly global distributed information infrastructure. The proposal addresses the key issues stated above.

3.6 Conclusion

The success of the scheme to use the directory for management purposes depends heavily on the deployment of the directory in general and on experimentation of the related schema that has been developed into experimental RFCs. Given wide enough experimentation the information of IP-networks in the directory could be routinely looked up for configuration management purposes. The directory service could offer a parallel DNS service. The disjoint *WHOIS* systems could be integrated in the directory. Most important, we envisage that newer , hitherto unthought of, uses of the directory will emerge. That is probably the most challenging and exciting aspect of this exercise.

第 4 章

Estimation of the Non-Optimality in FTP-traffic

4.1 Introduction

One of the most popular tools in the Internet is the quite simple file transfer protocol *ftp* [115]. This protocol enables users to send and receive files from virtually any IP-connected computer in the (network) world, provided proper access rights apply. Many organizations have set up *anonymous ftp server* which can be accessed by any user (in read-only mode in most cases). These ftp servers have been proven to hold a substantial number of useful files (software, documents, etc.), which are of interest to many people. As a result of broad usage, it has been found that **the ftp component constitutes nearly half of the network traffic along backbone and trunk routes** ([116], [117]).

Within a local or campus network, in most cases bandwidth will be sufficient to encourage file transfers via ftp. Problems arise when a user ftps (network jargon for transfering files using ftp) data from a non-local computer. Wide area connections between networks of different organizations generally have a limited bandwidth. Typically, a university campus network is connected to other networks by a link of 64 to 256 kByte/s (only in some cases 2 MByte/s and more). If a connection is established with a very remote network (e.g. from Japan to US), several transit networks have to be passed. In such a case, the end-to-end throughput might be decreased to a level below user acceptance. Thus, it makes a big difference in terms of time if a file is copied from a local or a remote server.

Naturally, there have been studies on reducing the volume of ftp-traffic. The Alex [118] system uses ftp operations in conjunction with NFS to build a local cache of retrieved files. [119] has made a case for building a hierarchically distributed cache of file objects inside Internetworks to reduce traffic. There have also been proposals to make ftp itself more efficient by incorporating ways of compression etc.

In this work we take a different line and show that the Internet already provides an environment rich with replications - the problem lies in the lack of awareness about the network configuration.

In general the same file is served by several ftp-servers in the Internet. This redundancy

if used judiciously could help in reducing network traffic. **But unfortunately without any support in the form of knowledge about the network topology this has not been the case. On the contrary, it has resulted in poor bandwidth utilization and deterioration in service.**

In this work we propose a model based on which the cost of retrieving a file is calculated. Based on this model we calculate the non-optimality involved in file retrievals by studying real life data from the logs of ftp-servers. **The results are used to show an effective way of having** *cheap* **and** *efficient* **file retrieval.**

4.2 Network traffic optimality

The global network can be considered as an undirected graph, a mesh. The edges represent lines with certain properties and nodes the gateways. Usually, the optimization problem is to find the shortest path between the nodes A and B. There are several algorithms that deal with *shortest-path routing*. These algorithms fall into two categories - *distance-vector algorithms DVA* and *link state algorithms LSA*. In the DVA a node (router) knows the length of the shortest path from each of its neighbours to every destination in the network. This information is used to compute the distance of the router itself from all destinations in the network and thereby select the shortest path. In the LSA each node (router) knows the entire network topology based on which it calculates the shortest path to the destination.

Our work is not concerned with the routing issues and algorithms. We look at the map of the network where the *shortest path* is being used by the underlying IP implementations. Thereby the reachability between the nodes is fixed - so is the communication cost for inter-node communication. Admittedly, there is an element of inaccuracy as the routing is dynamic, leading to costs that are varying. However from experience, the variation is small and the simplification is logical.

A path can be evaluated by summing the weights of all parts (edges) along this path. Routing protocols try to solve this optimization problem. In simple routing protocols, all edges have the weight 1. In that case, only the number of edges (called number of hops) counts for path evaluation and optimization. For networks, this means that the (different) bandwidth of links are not considered when estimating a path. There is, however, a significant difference between two links, when one has a capacity of say 512 kByte/s, and the other is a 9.6 kByte/s line.

For distributed network applications, path optimization is an even more difficult problem. Also with end-to-end routes optimized by lower-level protocols, a client-server-communication can be non-optimal. This can occur when the application allows a client to choose one out of several (apparently identical) servers to contact. The connection to each of these servers is in itself optimal, i.e. there is no better way to reach server X. But

the link to server X might be worse than the link to server Y. **Presently, applications like ftp do not provide a way to compare alternate client-server-connections with respect to network characteristics.**

Users are basically concerned with a "fast" connection to remote services. It does not make a difference where a server is located or which organization it belongs to. But one can foresee that in the near future users will also have to worry about the cost of sending network traffic over specific links or using certain services.

For this study, we investigated ftp traffic on the backbone of the Japanese Internet . **Connections chosen by users were compared against possible better connections.** Details of this investigation are given in the remainder.

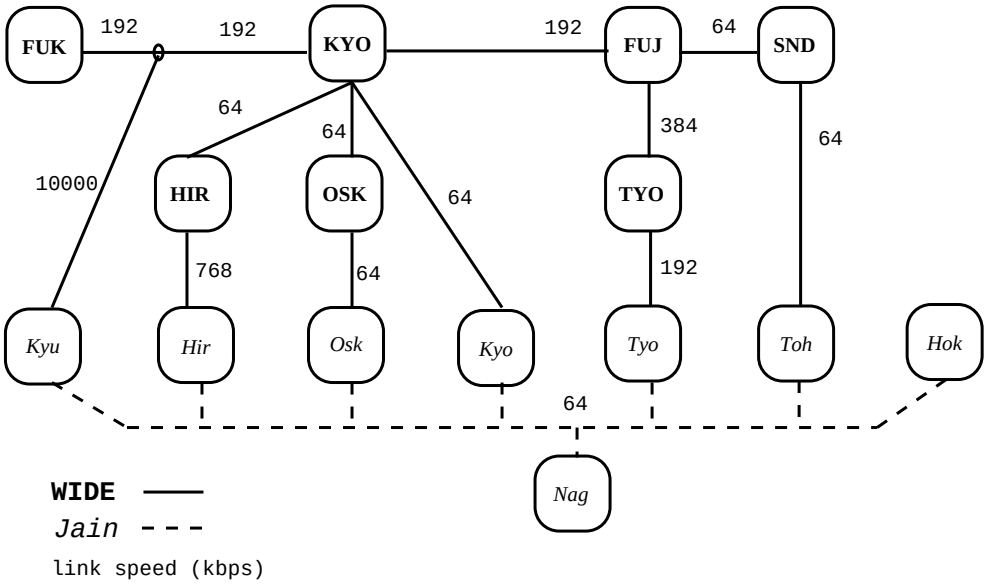
4.3 Network model for cost calculation

Strictly speaking for a mesh network consisting of n elements there will be $\frac{1}{2}n(n-1)$ *cost elements* which basically form a $[n \times n]$ symmetrical matrix. This does lead to practical computational problems in a network of reasonable size. However, fortunately enough the **configuration of networks are simpler and for all practical purposes are hierarchical with a backbone connecting regionals and the regional backbone connecting locals**. The hierarchy could be deep depending on the network.

This allows us a very useful simplification. Instead of looking at all the n elements we can concentrate on the m regionals i.e. on the inter-region costs. Generally m is much smaller and the cost computation problem is much more tractable.

In the case of our study, the medium size Japanese network, the constituent networks could be grouped into a reasonably small number of regions. For our study, we focussed on the networks belonging to the 15 WIDE or JAIN regions[1]. A region in this context is the group of regional networks which are connected to the backbone by a common gateway. Thus, they all have the same entry point for wide area traffic. Regions are interconnected by backbone links as depicted in the following figure.

[1]There are several other important constituent networks of the Japan Internet. However for the sake of simplicity we have restricted this study to the WIDE and JAIN regions only. The map itself is a simplified and is outdated now.



WIDE and JAIN regions.

As a result, the mesh graph with nodes and (weighted) edges can be used for further consideration of the network.

Traffic between regions is routed according to topology and policy rules. Unfortunately, there is no common database revealing all routes. Therefore we had to run a lot of traceroute checks to get an accurate picture of how packets are routed. Had we been interested only in the number of hops a packet takes from region A to region B, we were done with the model. The basic idea of this study was, however, to compare network connections with respect to their link properties, such as bandwidth, traffic, etc. This kind of information (attributes to existing links) is more difficult to gather. With the help of WIDE members at least the bandwidth of all backbone links was known. There was no reliable data for usage/traffic on the links. Therefore we had to estimate that all links are equal in this respect, and connection costs were calculated on the basis of link speeds only.

4.4 The Cost Model

4.4.1 Simple cost calculation

There are several possible definitions of cost. Cost could reflect tariff, speed, policy, ... with varying degree of weightage on the parameters.

For our purposes - namely estimating the non-optimality of traffic - we concentrated on the time the network packet exists in the network. In the

present situation lines are either leased or free so the tariff does not come into the picture (There were no dial-up lines in the picture that we considered).

The cost has two components:

$$cost = time_{at-nodes} + time_{in-transmission}$$

The time at nodes is of the order of micro seconds and is negligible when compared with the time in transmission along lines of max speed 1 Mbps.

Thus,

$$cost \approx time_{in-transmission}$$

As unit of cost we took the time taken for 1 Mbyte to reach from source to destination. This way, the definition of cost is reduced to

$$cost = \sum_{i=1}^{n} \frac{1000}{sp_i}$$

where sp_i is the speed of a single link i in kbps, and n the total number of links involved in the end-to-end connection.

4.4.2 Total cost calculation

The simplified communication model wherein the network comprises of 15 regions and the regions are connected by the backbone networks, greatly simplifies the total-cost calculation. It is clear that

$$\begin{aligned} total\_cost_{absolute} &= cost_{inter-node} \\ &= cost_{source-region} + cost_{inter-region} + cost_{destination-region} \end{aligned}$$

where,

- source-region [destination-region] costs are the costs from [to] the source [destination] node to [from] the backbone router and,
- inter-region cost is the cost between the source-region router and the destination-region router on the backbone.

As far as comparative costs from one fixed starting point are concerned the source-region cost will drop - so that

$$total\_cost_{compare} = cost_{inter-region} + cost_{destination-region}$$

Thus **to carry out the cost calculations we needed a [15×15] symmetric matrix of inter-region costs and an array of intra-region costs. Calculation of this matrix using network configuration information and the above cost model was straight forward.**

表 4.1: Connection costs as seen from jp-gate.wide.ad.jp (Fujisawa)

target region	cost		region	cost
WIDE Fukuoka (FUK)	156		WIDE Hiroshima (HIR)	208
WIDE Osaka (OSK)	208		WIDE Kyoto (KYO)	52
WIDE Tokyo (TYO)	26		WIDE Sendai (SND)	156
JAIN Kyushu (Kyu)	104		JAIN Hiroshima (Hir)	221
JAIN Osaka (Osk)	364		JAIN Kyoto (Kyo)	208
JAIN Nagoya (Nag)	234		JAIN Tokyo (Tyo)	78
JAIN Tohoku (Toh)	312		JAIN Hokkaido (Hok)	234

Table 4.1 gives an extract of this matrix. Here, the connection costs from one region (FUJ) to all other regions are given.

When placing client and server of an application into the according regions, we can estimate the relative cost this connection has in comparison to other connections. For anonymous ftp, a comparison of servers located in different regions is possible from the very point a user is situated.

4.5 Data Collection

For this study, we were interested in real-life ftp traffic statistics. When a user from host A received a file from host B, we want to find out if there would have been a host C which is better accessible for user at A for this specific file transfer.

For this purpose, records are necessary that contain the following information:

- ftp client host
- ftp server host
- name of file transferred
- size of file transferred
- date and time of transfer

This data can be obtained by logging all ftp client sessions . There are tools around for collecting statistical data on outgoing ftp traffic, but the resource requirements are heavy, distributed and complicated. Therefore, we chose to use the logs generated by the ftp servers. This approach is simple - processing the logs was a bit more complicated, but the information content was sufficient.

Data has been collected for the following well-known anonymous ftp servers shown in Table 4.2.

表 4.2: Summary of ftp server logs

server	operations	logtime	operations/hour
ftp.is.titech.ac.jp	4295	21 days 14:19 hours	8.3
ftp.kuis.kyoto-u.ac.jp	61607	58 days 22:32 hours	43.5
ftp.mei.co.jp	2658	27 days 22:21 hours	3.9
ftp.ricoh.co.jp	461	19 days 04:40 hours	1.0
ftp.tohoku.ac.jp	6643	12 days 23:48 hours	21.3
utsun.s.u-tokyo.ac.jp	5260	8 days 23:45 hours	24.4

Furthermore, the contents of all anonymous ftp servers within the considered area (in this case: the Japanese Internet) have to be known. For cross-checking of file retrieval possibilities, records are necessary containing:

- name of file
- size of file
- date file was written

Collection of these data records was fairly easy, because all we had to do was get ls-lR files from ftp servers. This was done for 44 anonymous ftp servers within Japan. This list of 44 servers (see table 4.3) was compiled out of several documents and the set of servers registered with the central Japanese archie server archie.wide.ad.jp.

Details of processing these records are discussed in the following section.

4.6 Data processing

4.6.1 Preprocessing of logs

Before optimality checks were carried out, the input files had to be preprocessed to generate a common input format for logs of all servers. In most cases, ftp-logs had been written by syslogd. But almost each syslogd used a different format to log date, time, name and size of file. Also, the complete information for one file retrieval is usually logged in two (not necessarily consecutive) lines: One entry containing information about the user, the other about the file retrieved. These two lines had to be merged. We defined a simple intermediate record format with the data fields:

```
ftp_server ftp_client file_name file_size retrieval_date
```

表 4.3: Major anonymous ftp servers in Japan

server	files		server	files
azabu.tkl.iis.u-tokyo.ac.jp	56		bash.cc.keio.ac.jp	3695
etlport.etl.go.jp	6907		ftp.ae.keio.ac.jp	693
ftp.ascii.co.jp	2168		ftp.c.u-tokyo.ac.jp	2148
ftp.cc.saga-u.ac.jp	5016		ftp.cs.keio.ac.jp	85939
ftp.cs.titech.ac.jp	1641		ftp.cs.uec.ac.jp	1718
ftp.csce.kyushu-u.ac.jp	3049		ftp.csrl.aoyama.ac.jp	206
ftp.dcl.co.jp	119		ftp.dit.co.jp	2678
ftp.ecei.tohoku.ac.jp	25619		ftp.elcom.nitech.ac.jp	26841
ftp.foretune.co.jp	2214		ftp.fujita3.iis.u-tokyo.ac.jp	10852
ftp.fujixerox.co.jp	152		ftp.ics.osaka-u.ac.jp	42208
ftp.is.titech.ac.jp	516		ftp.kuis.kyoto-u.ac.jp	162859
ftp.kyushu-id.ac.jp	580		ftp.mei.co.jp	9630
ftp.meiji.ac.jp	4166		ftp.nig.ac.jp	757
ftp.phys.keio.ac.jp	477		ftp.ricoh.co.jp	5676
ftp.sigmath.osaka-u.ac.jp	779		ftp.tohoku.ac.jp	4516
ftp.u-tokyo.ac.jp	51897		ftp.waseda.ac.jp	8444
ftp.wg.omron.co.jp	18583		icpcs41.icpc.fukui-u.ac.jp	1420
nuis.nuie.nagoya-u.ac.jp	3855		onlnews.kek.ac.jp	509
scslwide.sony.co.jp	2154		sh.wide.ad.jp	3948
solaris.ims.ac.jp	14		srawgw.sra.co.jp	32864
theta.iis.u-tokyo.ac.jp	19784		tutserver.tut.ac.jp	17704
utsun.s.u-tokyo.ac.jp	9120		wnoc-fuk.wide.ad.jp	3425

The hostname of the ftp_client was reduced to the domain name. This is quite simple for Japan, because there is a well structured naming of domains. Third-level names correspond to the organization, second level give the type of organization (ac=academic, co=commercial, go=government, ad=administrative), and top-level is .jp. Thus, all information below third level was supressed and user sites could easily be matched to the 15 regions shown above. All ftp accesses from sites outside Japan were mapped to the virtual domain foreign.ad.jp. They were processed as if originated at the international gateway point. The parameter retrieval_date was written in the unique way yymmddHHMMSS. Some entries in logs had to be left out because one or more parameters (e.g. originating domain) could not be identified.

Similarly the contents files of the ftp servers to check data against were preprocessed. To speed up searches, non-interesting fields were removed from ls-lR output format. As a result, a line of a contents file looked like:

```
file_name file_size write_date
```

4.6.2 The algorithm

With the network model described in subsection 4.2 and allocated cost indexes, network connections can be compared to each other. A connection with a smaller cost index is supposed to be cheaper because it uses faster links or less gateways. The basic idea of the algorithm presented here is to compare the connection a user has chosen (by retrieving a file from a certain server) to connections he could have used to retrieve the same file from another server. If there is at least one ftp server holding the same file and with a cheaper connection from the users site, we found a so called non-optimal retrieval. Correspondingly, an optimal retrieval is flagged when there was no cheaper way.

4.7 Results

The algorithm described in subsection 4.6.2 was run on the collected data (see subsection 4.5). Table 4.4 summarises the results for the studied ftp servers. Data is given in terms of number and size of files, and estimated cost.

These tables reveal that an average of 20 % of all files transferred with anonymous ftp use scarce backbone bandwidth without necessity. **For the amount of data being transferred (and this corresponds directly to packets on the network), the share of non-optimal traffic is more than 30 % of the total ftp traffic.**

It also shows that the more well known ftp-sites are subject to a larger percentage of non-optimal retrieval. This is understandable. **Without any support users tend to play it safe and attempt to obtain the file from the server that is most likely**

表 4.4: Ftp study summary

server	files		Mbytes		10^6 × cost		n-opt : total (%)		
	opt	n-opt	opt	n-opt	opt	n-opt	files	bytes	cost
ftp.is.titech.ac.jp	14	6	3	1	0	84	30.0	31.1	100.0
ftp.kuis.kyoto-u.ac.jp	14137	8341	2,154	1,647	401,027	550,794	37.1	43.3	57.8
ftp.mei.co.jp	1225	766	452	203	103,382	55,590	38.5	31.0	35.0
ftp.ricoh.co.jp	334	47	108	22	12,159	5,457	12.3	16.6	31.0
ftp.tohoku.ac.jp	4309	930	397	161	40,600	16,608	17.8	28.9	29.0
utsun.s.u-tokyo.ac.jp	2739	819	276	161	9,625	18,136	23.0	36.7	65.3

to have the file. In this study we have limited ourselves to national traffic on the national backbone only. A similar trend, and with more serious consequences could be expected to be seen if the traffic on the international backbones were examined.

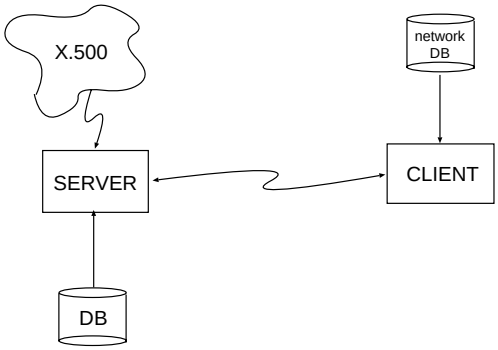
Looking at the overall cost for these file transfers, the figures reveal an even higher percentage of non-optimality traffic. Because in our cost model cost is basically equivalent to transmission time, much non-optimality means long waiting time for the user.

4.8 Summary of Results

The results presented in this paper confirm that there is indeed a lot of non-optimal ftp-traffic. Further, the non-optimality stems from the mis-use of the replication-rich file server environment in the Internet. With ever growing demand for network resources this abuse of backbone bandwidth will need to be limited. Bandwidth is a sacred resource which many users have to share. Also, with traffic accounting to come usage patterns have to change.

We have shown a way of avoiding this. **User support based on knowledge of the network topology could result in better service and utilization of network bandwidth.** We presented a model based on which the cost retrieving a file may be calculated. This shows an effective way of providing user support for *cheap* and *efficient* file retrieval. The SoftPages project [110] based on this concept is presently under experimentation.

4.9 SmartArchie



THE SMART ARCHIE

SmartArchie is a very simple improvement on the conventional Archie mechanism [120]. The conventional archie clients return a list of file-servers to the user. The list of file servers may be sorted according to some very simple ordering - for example based on a given ordering of the first-level domain components (*.jp, .us .uk, .edu .. etc.*). However, this is very inadequate - particularly when one considers the difficulty of associating any preference based on the next level (*.ac, .co .or, .ad .. etc.*) or levels below.

In its simplified form a SmartArchie client works by using the actual information on number of hops and the RTT for the various candidates. In the absence of such information - the client itself may collect the information using some simple network query tools - e.g. *ping, traceroute*. The results are cached in a local database- to improve performance. The list of servers returned by the Archie server is sorted based on this cost.

A SmartArchie server would use information from the Directory in the same manner as the SoftPages client does to sort the potential servers depending on the cost as seen by the client/user.

Conclusion

FTP traffic, one of the largest consumers of network bandwidth, is taken as a case study to investigate how effectively network configuration information can be used. A workable and scalable mechanism is shown using which a user or an application can evaluate the **cost** factor corresponding to several ftp-servers. By choosing the server with the lowest cost factor, network cost of the retreival is optimized. Cost could pertain to response time, tariff, or any other parameter. A network model based on which the comparative costs of several servers in a network can actually be calculated is shown. Two experimental implementations of the idea are discussed to demonstrate the effectiveness and practicality of the the proposal. The concept can be profitably employed in several other application which utilize ftp mechanisms e.g. World Wide File Server [121].

第 5 章

MIBs for Network Service Applications

5.1 Introduction

The rapid and widespread use of computer networking has highlighted the importance of research and development in the area of network management. In the presently accepted framework of network management, network elements [hosts, routers, bridges, ...] [107] are monitored and controlled through access to their management information, by Network management stations. The management information is viewed as a collection of managed objects [MO] and is termed as the Management Information Base.

For the SNMP suite of Protocols, MOs are defined using a subset of ASN.1, which in turn is defined in RFC 1155. The definition of **MIB**s is an ongoing process. MIBs are to be developed and defined as and when the requirement arises, while existing **MIB**s may be refined in accordance with the extensibility mechanisms defined in the SMI.

To date, network management systems have focussed on the lower three layers [Interface, IP, Transport] of the Internet suite of Protocolsa. The standard **MIB**-II covers these layers well. Further, more specific **MIB**s have been developed for particular protocols in each of these layers. However, with the maturation of network management systems , the need for managing network applications is increasing. The development in this area has become more interesting with the fact that there are a host of OSI applications [X.400, X.500, ..] running over the TCP/IP stack. As a result, the application management protocols will need to address both OSI and Internet applications.

5.2 Modelling Network Service Applications.

Unlike the lower layers where there is a very specific though complex, functionality, the application layer covers a very broad and general field of system-independent cooperation. The services required by the applications are very diverse. While some of these services are specific to the application others are used by diverse applications. It would be poor policy to approach the **MIB** development of a particular application without attempting to modularize the application into generic and specific parts. Thus, it is

intended that the application monitoring **MIB** be designed based on a model wherein the several generic submodules have been identified and corresponding **MIB**s developed. The specific submodules will be identified and developed as and when the necessity arises.

- Application process:
  Carries out the information processing for a particular application task. It is invoked either by a local user/program request, or by the initiating activity of a remote application requiring its cooperation. The Application process may be considered to be composed of two parts- the *application agent* and the *application entity*. The *Application agent* interfaces to the user and to the operating system of the end system on which it is mounted. The *application entity* performs the system independent part of the application activities.
- The Connection oriented model:
  Most network applications establish a connection with a peer application, carry out some activity , close the connection. In this simple model it is clear that the part dealing with the opening and closing of connections is common to applications while the part dealing with the activity will be application specific and will have minimal commonality with other appplications.

Ofcourse, it is clear that there will be applications which function in the connection-less fashion.

5.3 The Network Service Application MIB development

5.3.1 Why a Network Services Monitoring MIB ?

A substantial amount of work has been carried out in managing the communication infrastructure - the lines nodes and the communication protocol entities. With the basic management framework reasonably well established and with the growing spread of network applications like mail, news, DNS, Directory Services, ... , the time is ripe to take up the issue of managing network service applications[NSA]. NSAs include all applications that provide network services. The necessity of managing Network Service Applications[NSA] is particularly critical for NSAs which have (widely) distributed components such as the Mail and Directory Services.

5.3.2 The Network Management Framework

Though the target applications that we will be considering could be from the Internet suite of Protocols or from the OSI suite of protocols [or from any other suite of protocols

for that matter], we adopt the SNMP [Simple Network Management Protocol] network management framework which is perhaps the most widely and successfully deployed, to date.

Also, as will be clear in the latter part of this document we have attempted to adhere, as far as possible, to the principle of simplicity that is the underlying principle of SNMP.

The SNMP Network Management framework is laid out in the three documents-

- STD 16/ RFC 1155 [122] defines the generic structure of network management information
- STD 15/ RFC 1157 [123] defines the protocol for accessing network management information
- STD 17/ RFC 1213 [103] defines the primary set of managed objects.

The framework is adaptable/extensible by defining newer MIBs to suit the requirements of specific applications/protocols/devices/situations.

5.3.3 Scope of Management

The realm of NSA management is vast and there are several "requirements" that need to be met. However, in this document we have attempted to keep things simple and have focussed on the rudimentary aspects only. The primary goal is a simple, useful, and widely implementable scheme for NSA management which integrates nicely with the existing network management framework. The proposed NSA management centers around general monitoring

- to detect a broad range of operational problems: broken connectivity, system failure, congestion, ..
- to monitor performance and utilization

Admittedly, the capability to

- manage (monitor and control) the configuration of NSAs
- manage (monitor and control) the status of individual entries in the NSA queues.
- manage (monitor and control) the progress of a specific activity (e.g. locate and if necessary expedite a specific mail message)
- control the NSA operation - cause some action be performed (e.g. force an immediate connection and transfer of pending messages to some specific system).
- etc.

is impressive, useful and desirable from a NSA management system. But, these capabilities would require provisions for strict security checks, mandate a much more complex design, give rise to contentious alternatives between generic and implementation-specific approaches. Since such features are likely to be difficult to implement, we have opted for a simpler design that focusses on the basic monitoring aspects.

5.3.4 Relationship to Directory Services

The Directory Services [112] provide an excellent framework whence information can be retrieved for applications and management purposes. Already some NSAs [e.g. mail] are tied to the Directory Services. On some systems the Directory Services is being used to provide a dynamic mapping between application-name and the application-context. It is clear that placing static information in the directory and using it for management purposes will be advantageous as the richness of the directory information framework will not have to be duplicated in the MIB. [Static information is that which has a mean time of change of the order of days or longer.]

The emphasis in this work is on the dynamic component of the NSA related information that needs to be represented in the MIB.

To exploit the benefits of the Directory services, it is recommended that, wherever applicable, a linkage be established between the Directory component and the MIB component of management information. The following linkages may be considered:

- The MIB → Directory linkage: define a Managed Object [MO] that contains the Directory Name of the corresponding application entity. Management applications capable of performing directory queries will thus be able to supplement the dynamic information in the MIB with static information from the Directory by referring to the linkage MO. It is intended that this will be the normal case.

- Directory → MIB linkage: The Directory will contain a reference to the SNMP agent which is servicing the MIB of the application entity of interest. This will allow an SNMP capable directory query agent to probe dynamic characteristics of the application entity.

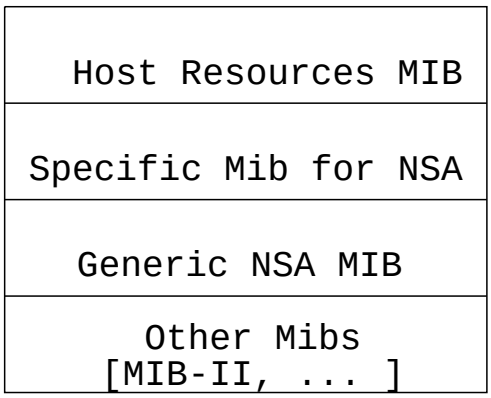
- This approach could be extended further by modelling SNMP MOs as attributes of directory entries. This would dramatically simplify the design of directory service agents.

5.3.5 The MIB Model for NSA management

A NSA-manager may wish to monitor several aspects of an operational NSA. He/she may want to know the process related aspects- the CPU, memory, .. utilization of the

operational NSA; the general network service related aspects - operational status, associations, ... and finally the NSA-specific information - its operations and performance.

The MIB proposed in this section covers only the general network service related aspects. It focusses on the NSA description, the operations and activity indications, the associations and association related details. The relationship of this MIB with the other MIBs is shown in the following figure.



Relationship of MIBs

The manager (application) of a NSA will use the Host Resources MIB [104] to obtain process related information [resource usage,..]. The proposed generic NSA MIB provides the information for the generic objects. The NSA specific objects are defined in the corresponding MIBs. For management information pertaining to the lower layer TCP/UDP/IP/... the MIB-II offers the repertoire of MOs.

There are NSAs to which some components of the the proposed generic MIB do not apply e.g. "connectionless" NSAs like DNS[95], NFS[124] do not use associations. MIBs for such NSAs for such applications may be developed independently of the generic MIB proposed in this work e.g. DNS- MIB [125][126] though the other components of the generic NSA MIB could still be used profitably for managing such NSAs.

5.3.6 NSA Managed Objects

The APPLICATION-MIB focusses on the NSA description, the operations and activity indications, the associations and association related details.

A table is defined which will have one row for each NSA running on the system. One of the columns is applName which is the Directory name of the NSA. This will be used to extract static information from the Directory. The table is indexed by an integer key applIndex which will also be used by the NSA-specific MIBs for reference purposes.

表 5.1: MOs for Network Service Application Management

MO	Parent	Type
application	mib-2.27	
applConformance	application.3	
applGroups	applConformance.1	
applCompliances	applConformance.2	
applTCPProtoID	application.4	
applUDPProtoID	application.5	
applTable	application.1	Aggregate
applEntry	applTable.1	Aggregate
applIndex	applEntry.1	INTEGER
applName	applEntry.2	DisplayString
applDirectoryName	applEntry.3	DistinguishedName
applVersion	applEntry.4	DisplayString
applUptime	applEntry.5	TimeStamp
applOperStatus	applEntry.6	INTEGER
applLastChange	applEntry.7	TimeStamp
applInboundAssociations	applEntry.8	Gauge32
applOutboundAssociations	applEntry.9	Gauge32
applAccumulatedInboundAssociations	applEntry.10	Counter32
applAccumulatedOutboundAssociations	applEntry.11	Counter32
applLastInboundActivity	applEntry.12	TimeStamp
applLastOutboundActivity	applEntry.13	TimeStamp
applRejectedInboundAssociations	applEntry.14	Counter32
applFailedOutboundAssociations	applEntry.15	Counter32
assocTable	application.2	Aggregate
assocEntry	assocTable.1	Aggregate
assocIndex	assocEntry.1	INTEGER
assocRemoteApplication	assocEntry.2	DisplayString
assocApplicationProtocol	assocEntry.3	ObjectID
assocApplicationType	assocEntry.4	INTEGER
assocDuration	assocEntry.5	TimeStamp
applGroup	applGroups.1	
assocGroup	applGroups.2	
applCompliance	applCompliances.1	
assocCompliance	applCompliances.2	

5.3.7 MIB Model for DSA Management

A DSA-manager may wish to monitor several aspects of the operational DSA. He/she may want to know the process related aspects-the resource utilization of the operational DSA; the network service related aspects e.g., inbound-associations, outbound-associations, operational status, and finally the information specific to the DSA application - its operations and performance.

The MIB defined in this part covers the portion which is specific to the DSA-application. The network service related part of the MIB, and the host-resources related part of the MIB, as well other parts of interest to a Manager monitoring the DSA-application, are covered in separate documents [105][104].

5.3.8 The DSA functions and operations

The Directory System Agent [DSA], a component of the OSI-Directory [112][127], is an application process. It provides access to the Directory Information Base [DIB] to Directory User Agents [DUA] and/or other DSAs. Functionally , a User [DUA] and the Directory are bound together for a period of time at an access point to the Directory [DSA]. A DSA may use information stored in its local database or interact with (chain the request to) other DSAs to service requirements. Alternatively, a DSA may return a reference to another DSA.

The local database of a DSA consists of the part of the DIT that is mastered by the DSA, the part of the DIT for which it keeps slave copies and cached information that is gathered during the operation of the DSA.

The specific operations carried out by the DSA are: Read, Compare, AddEntry, ModifyEntry, ModifyRDN, RemoveEntry, List, Search. There is also the special operation Abandon. In response to requests results and/or errors are returned by the DSA.

5.3.9 X.500 Directory Monitoring MIB design

The basic principle has been to keep the MIB as simple as possible. The Managed objects included in the MIB are divided into three tables - dsaOpsTable, dsaEntryTable and dsaIntTable.

- The dsaOpsTable provides summary statistics on the accesses, operations and errors.
- The dsaEntriesTable provides summary statistics on the entries held by the DSA and on cache performance.
- The dsaIntTable provides some useful information on the interaction of the monitored DSA with peer DSAs.

There are references to the Directory itself for static information pertaining to the DSA. These references are in the form of "Directory Distinguished Name" [128] of the corresponding object. It is intended that DSA management applications will use these references to obtain further related information on the objects of interest.

表 5.2: MOs for Directory Management

MO	Parent	Type
dsaMIB	mib-2.29	
dsaConformance	dsaMIB.4	
dsaGroups	dsaConformance.1	
dsaCompliances	dsaConformance.2	
dsaOpsTable	dsaMIB.1	Aggregate
dsaOpsEntry	dsaOpsTable.1	Aggregate
dsaAnonymousBinds	dsaOpsEntry.1	Counter32
dsaUnauthBinds	dsaOpsEntry.2	Counter32
dsaSimpleAuthBinds	dsaOpsEntry.3	Counter32
dsaStrongAuthBinds	dsaOpsEntry.4	Counter32
dsaBindSecurityErrors	dsaOpsEntry.5	Counter32
dsaInOps	dsaOpsEntry.6	Counter32
dsaReadOps	dsaOpsEntry.7	Counter32
dsaCompareOps	dsaOpsEntry.8	Counter32
dsaAddEntryOps	dsaOpsEntry.9	Counter32
dsaRemoveEntryOps	dsaOpsEntry.10	Counter32
dsaModifyEntryOps	dsaOpsEntry.11	Counter32
dsaModifyRDNOps	dsaOpsEntry.12	Counter32
dsaListOps	dsaOpsEntry.13	Counter32
dsaSearchOps	dsaOpsEntry.14	Counter32
dsaOneLevelSearchOps	dsaOpsEntry.15	Counter32
dsaWholeTreeSearchOps	dsaOpsEntry.16	Counter32
dsaReferrals	dsaOpsEntry.17	Counter32
dsaChainings	dsaOpsEntry.18	Counter32
dsaSecurityErrors	dsaOpsEntry.19	Counter32
dsaErrors	dsaOpsEntry.20	Counter32
dsaEntriesTable	dsaMIB.2	Aggregate
dsaEntriesEntry	dsaEntriesTable.1	Aggregate
dsaMasterEntries	dsaEntriesEntry.1	Gauge32
dsaCopyEntries	dsaEntriesEntry.2	Gauge32
dsaCacheEntries	dsaEntriesEntry.3	Gauge32
dsaCacheHits	dsaEntriesEntry.4	Counter32
dsaSlaveHits	dsaEntriesEntry.5	Counter32
dsaIntTable	dsaMIB.3	Aggregate
dsaIntEntry	dsaIntTable.1	Aggregate
dsaIntIndex	dsaIntEntry.1	INTEGER
dsaName	dsaIntEntry.2	DistinguishedName
dsaTimeOfCreation	dsaIntEntry.3	TimeStamp
dsaTimeOfLastAttempt	dsaIntEntry.4	TimeStamp
dsaTimeOfLastSuccess	dsaIntEntry.5	TimeStamp
dsaFailuresSinceLastSuccess	dsaIntEntry.6	Counter32
dsaFailures	dsaIntEntry.7	Counter32
dsaSuccesses	dsaIntEntry.8	Counter32
dsaOpsGroup	dsaGroups.1	
dsaEntryGroup	dsaGroups.2	
dsaIntGroup	dsaGroups.3	
dsaOpsCompliance	dsaCompliances.1	
dsaEntryCompliance	dsaCompliances.2	
dsaIntCompliance	dsaCompliances.3	

第 6 章

NIC 情報のディレクトリ化

6.1 はじめに

これまで、ネットワークに関する情報をディレクトリに格納し、各管理システムがディレクトリにアクセスすることにより必要な情報を得て、ネットワーク管理を実現する環境を提案してきた。

現実には、ネットワークに関する情報を保持するディレクトリは、まだ存在しないので、初めにそれを構築しなければならない。ところでネットワーク情報は、NIC(以後、国や地域のネットワークインフォメーションセンターを指す) と呼ばれる機関が管理している。

本稿では、日本の管理機関である JPNIC(Japan Network Information Center) と、アジア太平洋地域のために設立準備されている APNIC(Asia Pacific Network Information Center) の登録情報をディレクトリ化するまでの過程と問題点を述べる。

また、ディレクトリへの登録データを利用した応用システムも紹介する。

6.2 NIC 情報の現状

JPNIC の管理している情報や提供するサービスについては、例えば [129] から連載で紹介されている。一般組織向けの管理情報として、参加組織について以下の項目がある。(詳細は、[130] を参照されたい)

- ドメイン情報 (ドメイン名, 組織名, 使用 IP ネットワーク, 管理者)
- ネットワーク情報 (IP アドレス, ネットワーク省略名, 管理者)
- ホスト情報 (ネームサーバホスト名,IP アドレス)
- 個人情報 (住所, 氏名, 電話番号, メール宛先, 所属組織/部局)

このほかに、プロジェクト情報 (ネットワークプロジェクト)/NOC 情報 (ネットワーク運用センター)/コミュニティ情報 (メーリングリスト) がある。

APNIC の管理する情報は、以下の二つである。

- ネットワーク情報 (IP アドレス, ネットワーク名, 管理者)
- 個人情報 (住所, 氏名, 電話番号, メール宛先)

これらの情報は、mail や whois といったツールによって参照できる。しかし、個別管理のために以下の問題が発生する。

- 各 NIC によって、提供形式が異なっている。(項目, 言語, 表記)
- 検索時、関係しそうなホストを全て個別にアクセスする必要がある。
- 会社/部門/個人が、階層的に提供されない。

6.3　ディレクトリの利点

ディレクトリでは、実世界の対象毎に対応する情報を集め、管理の単位 (エントリ) としている。ディレクトリ全体 (Directory Information Base) は、このエントリをツリー構造 (Directory Information Tree) に沿って整理・格納したものである。

また、情報提供を受ける利用者側とディレクトリとのプロトコルは DAP(Directory Access Protocol)、ディレクトリ内部でのプロトコルは DSP(Directory System Protocol)、という形で規定されている ([112])。このディレクトリの以下の特性によって、前記の問題を解決できる。

- どの情報格納システムにアクセスした結果であっても、一定の形式で情報を取得できる。
- ツリー構造により、全体のデータ構造を知らなくても、希望するデータにアクセスできる。
- 別名や複数の属性値により、言語や表現の違いを吸収する仕組みがある。

つまり、図 6.1に示すように、現在の NIC 情報ユーザ利用者は、情報がありそうな NIC 全てにアクセスしなければならないが、ディレクトリ利用者は、ひとつのアクセス方法を知っているだけでよい。

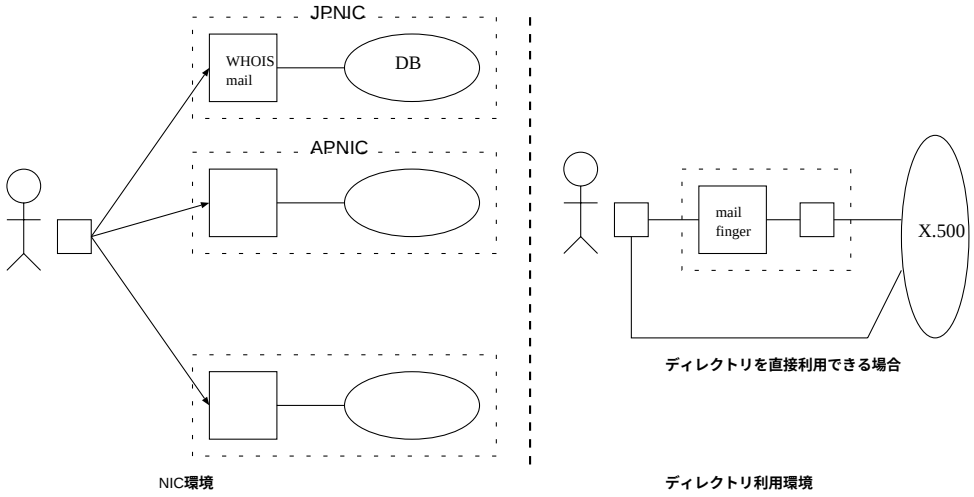


図 6.1: NIC とディレクトリ環境の比較

6.4 ディレクトリへの登録

登録のためのディレクトリ・ソフトウェアは、現在よく普及しているISODE の QUIPU[131] (DUA(Directory User Agent) は、dish) を使用した。手順は、以下の通りである。

1. ディレクトリ・エントリの構造 (オブジェクトクラス, 階層, 属性など) を決定する。
2. NIC 情報の各項目 (会社/部門/個人など) を関連付け、上記の構造に合致させる。
3. dish を用いて、エントリを add する。

ここで、今回登録したオブジェクトと NIC 情報 (登録フォームのレコード) の対応関係を図 6.2〜図 6.6に示す。(図中、[] で括られた部分が NIC 情報から得られる属性値である。)

QUIPU で標準的に用意されていないオブジェクト (ipNetworkObject など) については、[98]・[99] の定義を利用した。

```
objectclass= organization & domainRelatedObject & quipuNonLeafObject
@c=                             (エントリ位置)
o=                              [Organization]
description=                    [Description]
postalCode=                     [郵便番号]
postalAddress=                  [Address]
telephoneNumber=                (関連する個人情報の電話番号)
facsimileTelephoneNumber=       (関連する個人情報の FAX 番号)
associatedDomain=               [ドメイン名]
masterDSA=                      (登録する DUA 名)
```

図 6.2: 組織情報

```
objectclass= organizationalUnit & quipuNonLeafObject
@c=                             (エントリ位置)
ou=                             (個人情報の [Division])
masterDSA=                      (登録する DUA 名)
```

図 6.3: 組織の部局情報

```
objectClass= top & quipuObject & quipuNonLeafObject &
 CommunicationObject & ImageCommunicationObject &
 networkImage & ipNetworkImage
@c=                             (エントリ位置)
IPnw=                           [ワークアドレス省略名]
ipNwNumber=                     [IP ネットワークアドレス]
adminContact=                   [運用責任者]
techn Contact=                  [技術連絡担当者]
description=                    [Description]
IpNwMask=                       (ネットワークマスク)
associatedDomain=               (関連付けた情報)
masterDSA=
```

図 6.4: ネットワーク情報

```
objectClass= top & person & pilotObject & newPilotPerson
 & quipuObject& pilotPerson
@c=                             (エントリ位置)
cn=                             [First, Last]
sn=                             [Last]
postalCode=                     [郵便番号]
postalAddress=                  [Address]
telephoneNumber=                [電話番号]
facsimileTelephoneNumber=       [FAX 番号]
mail=                           [電子メイル]
whoisIdent=                     [NIC ハンドル]
```

図 6.5: 個人情報

```
objectClass= top & domain & quipuObject & quipuNonLeafObject
@c=                             (エントリ位置)
dc=                             (JP の直前のドメイン)
o=                              (ドメイン名の xx.JP を除いた部分)
postalAddress=                  [Address]
postalCode=                     [郵便番号]
description=                    [Description]
associatedName=                 [Organization]
masterDSA=
```

図 6.6: ドメイン情報

6.5　登録における問題

上記の手順において以下の問題点があり、データ形式変換などの処理を必要とした。

1. NIC 情報の形式と dish のスキーマとの不一致の修正
   - 住所を郵便 O/R アドレス (30 文字 6 行) の形式に変換する。
   - '#' や'&' など、属性値スキーマに一致しない文字を変換する。
2. 所望するモデルのための名前付けの工夫
   - ipNetworkImage。JPNIC 情報では、ひとつのネットワーク省略名に複数の使用 IP ネットワークが対応しているので、ネットワーク省略名だけでは同一エントリ位置になる場合がある。
     今回は、'IP ネットワークアドレス%ネットワーク省略名' とした。
   - 同姓同名の区別。特に、APNIC 情報からは組織名を機械的に取得できないので、姓名だけでは同一エントリ位置になる場合がある。
     今回は、同姓同名の場合、' 姓名%NIC ハンドル名' とした。

6.6　データの実際

以上のように登録したエントリの dish による表示画面を図 6.7〜図 6.10に示す。([113])
JPNIC 情報は、図 6.7の organizationName=Japan Network Information Center
APNIC 情報は、organizationName=Asia Pacific Network Information Center の位置にそれぞれ格納されている。特に JPNIC 情報では、組織名の下に部局名が、その下に個人名が登録され、階層構造が正しく構築されているのがわかる。

```
Dish -> squid
Connected to rhea at '0101'H/Internet=150.80.254.2+17003
Current position: @c=JP
User name: @
Current sequence: default
Dish -> list
1   pss.4000000.1.2="A"

6   organizationName=AIC Systems Laboratories

8   organizationName=Asia Pacific Network Information Center

16  organizationName=Japan Network Information Center

28  organizationName=WIDE
```

図 6.7: エントリ位置:@c=JP

```
Connected to rhea at '0101'H/Internet=150.80.254.2+17003
Current position: @c=JP@o=Japan Network Information Center
@l=Registered Organizations
User name: @
Current sequence: default
Dish -> list
57  organizationName=3Com K.K.

77  organizationName=Advanced Integration Technology, Inc., Japan
79  organizationName=Advanced Intelligent Communication Systems
 Laboratory, LTD.
80  organizationName=Advanced Materials Lab., Inc
81  organizationName=Advanced Mobile Systems, Inc.
82  organizationName=Advanced Software Institute
```

図 6.8: エントリ位置:@c=JP@o=Japan Network Information Center @l=Registered Organizations

```
Dish -> squid
Connected to rhea at '0101'H/Internet=150.80.254.2+17003
Current position: @c=JP@o=Japan Network Information
Center@l=Registered Organizations@o=Advanced Intelligent Communication
 Systems Laboratory, LTD.
User name: @
Current sequence: default
Dish -> list
111 ipNetworkImageName=AIC-NET1
112 organizationalUnitName=Intelligent Network Management Division
```

図 6.9: エントリ位置:@c=JP@o=Japan Network Information Center@l=Registered Organizations@o=Advanced Intelligent Communication Systems Laboratory, LTD.

```
Dish -> squid
Connected to rhea at '0101'H/Internet=150.80.254.2+17003
Current position: @c=JP@o=Asia Pacific Network Information Center
@l=Registered Persons
User name: @
Current sequence: default
Dish -> list
120 commonName=Adam Kaye-Smith
121 commonName=Adam Olmstead
122 commonName=Adi Indrayanto
123 commonName=Adrian Corston
124 commonName=Adrian Ward
125 whoisIdent=(undefined)%surname=Harianto%commonName=Agus Harianto
126 whoisIdent=AH107%surname=Harianto%commonName=Agus Harianto
127 whoisIdent=AH11%surname=Harianto%commonName=Agus Harianto
128 commonName=Akane Shindo
129 commonName=Akinori Saitoh
130 commonName=Akio Ichikawa
131 commonName=Akira Fujii
132 commonName=Akira Kato
133 commonName=Akira Nagamine
134 commonName=Akira Okihana
135 commonName=Akira Tomoyoshi
136 commonName=Akiro Kosakada
137 commonName=Alan Brooks
138 commonName=Alan Brown
139 commonName=Alan Hogan
140 commonName=Alan Ramadan
141 commonName=Alan Roberts
```

図 6.10: エントリ位置:@c=JP@o=Asia Pacific Network Information Center@l=Registered Persons

6.7　まとめと今後の課題

NIC 情報を機械的にディレクトリに登録する方法を述べた。このディレクトリを利用した現在稼働中の情報検索システムを示す。このシステムでは、mail や finger のデーモンを修正し、

x500-query@wnoc-snd-ss2.wide.ad.jp に mail を送信するか (本文を find key とする)、または、

finger key@wnoc-snd-ss2.wide.ad.jp を実行することによって、従来の whois では行なえなかった電話/ファクス番号による検索も可能にしている。(図 6.11・図 6.12参照)

今後の課題を以下に示す。

- 情報源 (NIC) の更新をディレクトリにも反映するツールの開発。
- マルチ言語 (今回は、JPNIC 情報の日本語部分は欠落している。) やディレクトリの国際化への対応。
- NIC にない情報のディレクトリへの登録とその応用システムの開発。

```
Received: by wnoc-snd-ss2.wide.ad.jp (5.65+1.5W/2.7W)
id AA00697; Tue, 10 May 94 19:43:27 JST
Date: Tue, 10 May 94 19:43:27 JST
Return-Path: <daemon@wnoc-snd-ss2.wide.ad.jp>
Message-Id: <9405101043.AA00697@wnoc-snd-ss2.wide.ad.jp>
From: "X.500 Query Program" <X500-Query@wnoc-snd-ss2.wide.ad.jp>
Subject: Re:
In-Reply-To: Your message of "Tue, 10 May 1994 19:44:22 +0900"
             <9405101044.AA01028@XXXXXX.aic.co.jp>
To: Ouchi Masatomo <oouchi@XXXXXX.aic.co.jp>

2 phone number matches were found for '279-3310':
  "Advanced Intelligent Communication Systems Laboratory
  Glenn Mansfield         Dr.
```

図 6.11: x500-query@wnoc-snd-ss2.wide.ad.jp からのリプライ

```
XXXXXX.oouchi 31:  finger 279-3310@wnoc-snd-ss2.wide.ad.jp
[wnoc-snd-ss2.wide.ad.jp]
X.500 Finger Service...
2 phone number matches for "279-3310":
    Name                 Whois ID  e-mail
    ====                 ========  ======
  "Advanced Intelligent Communication Systems Laboratory
  Glenn Mansfield                  glenn@aic.co.jp
```

図 6.12: finger 279-3310@wnoc-snd-ss2.wide.ad.jp の実行画面