MASTER'S THESIS

Network Monitoring by Observing Message Codes

by F.A. De Almeida
Advisor: J.S. Baras

CSHCN M.S. 96-2
(ISR M.S. 96-11)

The Center for Satellite and Hybrid Communication Networks is a NASA-sponsored Commercial Space Center also supported by the Department of Defense (DOD), industry, the State of Maryland, the University of Maryland and the Institute for Systems Research. This document is a technical report in the CSHCN series originating at the University of Maryland.

Web site http://www.isr.umd.edu/CSHCN/

Network Monitoring by Observing Message Codes

by

Fernando Alexandre De Almeida

Thesis submitted to the faculty of the Graduate School of the University of Maryland in partial fulfillment of the requirements for the degree of Master of Science 1995

Advisory Committee:

Dr. J. S. Baras (Advisor / Chairman)

Dr. M.O. Ball

Dr. S. Marcus

Dedication

To my parents Rudesinda and Jose Joao, to my wife Nancy and to my daughter Marian, who came in the middle of this work.

Acknowledgments

I would like to express my sincere thanks to my wife and my daughter. They have been patient and very supportive during the challenging times at graduate school. With their continuous positive attitude I have found inspiration to set and continue pursuing higher goals in my career and in our lives.

I would also like to express my thanks to my advisor Dr. John Baras, for his continuous support, encouragement and patience throughout this work. His contagious enthusiasm has been a great source of motivation in my work. His experience pointing out research directions have helped me to widen my views on applied research.

Thanks to Dr. Baras, the Institute for Systems Research and the CSHCNS for the financial support they provided to me in the form of research assistantships, without which it would be virtually impossible to pursue and achieve these goals.

I would like to thank Dr. Ball and Dr. Baras for their efforts to get me an internship with a company in the telecommunications area; this internship helped me link the research we do at ISR/CSHCNS with the real needs faced by industry. The internship was a major breakthrough in my comprehensive understanding of the problems addressed here.

I would also like to express my appreciation to my thesis committee for their time, effort, and service. In alphabetical order, these three faculty members are: Dr. John Baras, Dr. Mike Ball, and Dr. Steve Marcus.

Section Page

Table of Contents

List of Figures

List of Tables

List of Abbreviations

The following acronyms are based on definitions presented by the major data and voice network carriers as well organizations that develop communications standards:

Binary Code: An electrical representation of quantities expressed in the base 2 number system.

Bit: Contraction of "Binary DigIT", the smallest unit of information in a binary system.

Bit Rate: The speed at which bits are transmitted, usually expressed in bits per second.

CRC: Cyclic Redundancy Checksum.

CMIP: Common Management Information Protocol

CMIS: Common Management Information Services

DL: Data Link

IP: Internet Protocol

ISO: International Standardization Organization

LLC: Logical Link Control.

MIB: Management Information Base

Network: A series of points connected by communications links

SAP: Service Access Point

SNMP: Simple Network Management Protocol

Chapter 1. Introduction and Problem Definition

1.1. Introduction

Today's communication networks are large, utilize several media types and are becoming increasingly intelligent. A typical network is characterized by its heterogeneity, its layered structure, and the distribution of its resources. Users are all painfully aware of their dependency on their networks. Networks linking critical business elements become vital. A multimillion dollar data center that becomes isolated through communications failures is worthless to its users. Likewise, a network link that supports a crucial application becomes extremely important because loss of the link can have a dramatic negative impact on an organization's bottom line.

The increase in complexity brings along with it several issues related to network management. Two of these issues are network configuration and network health status. Network management systems (NMSs) are the technological solutions to users' demand for network availability control. As NMSs monitor networks for unusual conditions, test components and communication lines, reconfigure devices to accommodate problems or changes, analyze and plan capacity, account for usage, and control network access, network administrators must increase their knowledge of the relationships between systems and the environments they manage.

A good example is related to network configuration and health status. The network configuration and health status have to be updated in real time to make sure the network operation is making the best use of network resources. This need for information has a price, which manifests itself as overhead of control information crossing the network, which may affect the overall Quality of Service (QOS). In addition there is need for fast processing this control information.

The network management process, despite the level of sophistication that it may currently have, still relies strongly on the human factor represented by the network operators. The human interpretation of a network event varies with the operator's experience. It means that for the same network performance pattern, different operators at the Network Control Center may react differently. Furthermore, the successive arrival at NCC of all sort of event messages, from a variety of network elements, may impair the operators' ability to make the best judgement on the severity of the network state. This may become very dangerous in a scenario where the network is migrating into a critical state that may disrupt the entire service. The avalanche of management data may as well impair the capability of fault management systems to perform their function in a timely basis.

The requirements over the Network Control Centers are changing at a very fast pace to keep up with the new technology being deployed in the field, as well with new applications made available to network users. This dynamic scenario reinforces the idea that the effectiveness of network surveillance relies on how fast Network Control Centers can interact with the network elements and also on how much they can anticipate and prevent critical trends. By understanding quickly the impact of network events, operators and fault management tools can warn network users and on-line generate new traffic routes in order to avoid potential problems.

The scope of this work is to add performance to currently available Network Management Systems. The technique described mostly in chapters 4 to 7 offers an additional dimension when updating views of the network. The interpretation of its results can be wisely used to modulate the amount of overhead traffic, provide a crude prediction of the trend of the network and create uniform action-reaction guideline when network operators face different network situations. Chapters 2 introduces basic

definitions related to networking and network management. Chapters 3 and 4 introduce the proposed technique by means of formulating the Multiple Attribute Decision Making problem as a fuzzy MADM problem. Chapter 5 explains how to apply the results of the fuzzy MADM. Chapters 6 and 7 describe experimental results, conclusions and suggest additional research. A series of appendixes have also been provided. They cover basic issues related to Quality of Service and networking.

1.2. Problem Definition

Through Element Management Systems, network devices send messages relative to network happenings to event windows on operator workstations, thereby keeping the Network Control Center (NCC) operators informed. Each event is parsed or translated into user-defined messages, which can be augmented to include information such as element affected, date and time of occurrence, condition description, and specific parameter values exceeded.

In this scenario, one of the critical network surveillance problems is related to the ever increasing size of the networks and the large amount of event messages that arrive at each operator's screen. Part of this problem is because the surveillance strategy relies on polling-based protocols. Information is gathered by polling; consequently, managers generally receive information only when they request it and then receive information whether or not it has changed. The polling process wastes network resources, especially in WAN environments. The large amount of information raises yet another problem, the ability of operators to perform preliminary assessment and respond quickly to the more critical situations.

Most of the Network Control Centers still adopt reactive instead of proactive approach to fault management. Detecting a problem is always desirable, however, preventing it

from happening by observing early indicators is extremely attractive.

The evaluation of hazards presented by a new situation depends at large on the operator's expertise in dealing with similar problems. This creates a lack of consistency on the management of the network. Implicitly several correlations happen during the operator's decision making process. This lack of consistency can be described as the missing capability of managers to create and maintain a uniform action-reaction mechanism when facing different network situations.

The network monitoring problems here addressed can be summarized as:

I. Establishment of a common measure that uniformly indicates the degree of severity of the situation faced by the network.

II. Ability to quickly identify potential network performance variations regardless of network complexity.

III. Ability to estimate network performance trends in a complex network environment.

IV. Ability to reduce overhead traffic based on preliminary network health monitoring.

Chapter 2. Fault Management, Manager and Agents

2.1. Introduction

Communication networks include three distinct network domains: the customer's premises, the local exchange network, and the interexchange. Customers receive a variety of services including voice, data and video. To ensure the highest level of quality of service, network operators, customer support personnel and field engineers operate and interact through the Network Control Center.

In today's business, equipments, systems, and applications are tightly integrated. A typical corporate data network often contains several separate networks to meet a variety of data communications requirements. A transaction network may be required to handle such data as credit card verification; a remote job entry network, for the input from batch computers; a time sharing network, for word processing and time sharing computers. Each network may operate its own communication facility using separate hardware and protocols. This approach may lead to a large, fragmented system often requiring separate communication links between identical points in the network to handle different protocols.

Business applications ultimately dictate the requirements for quality of service and network availability. Executive decisions at application level have to be supported by the fault management system. A good fault management system has to provide in a timely basis, all levels of correlation between a network problem and its effects on the overall business enterprise. In this context, one may define network management data as anything that carries important information about the state of equipment, systems, or applications. A network consists of one or more manager systems, or network management stations, and a collection of agent systems, or network elements.

2.2. Event Definition

Are discrete occurrences that happen at a particular point in time in the system. Any intelligent component may generate a message when it observes some event that might be of interest to operators. These events carry information on statistics and changes in the state of nodes. Changes in the state may be originated by reloading a new configuration table, by sudden alteration in traffic patterns, by software malfunctions or by hardware malfunctions.

2.3. Management Information Base

Conceptually, the information on the agent is known as the Management Information Base (MIB). The MIB is not a physically distinct database, but rather logically encompasses configuration and status values normally available on the agent system. A specific type or class of management information is called a MIB object (for example, a system description or an interface status). The existence of a particular value for a MIB object in the agent database is called an instance. Some MIB objects have only a single instance for a given agent system (for example, system description). Other MIB objects have multiple instances for a given agent system (for example, interface status for each interface on the system).

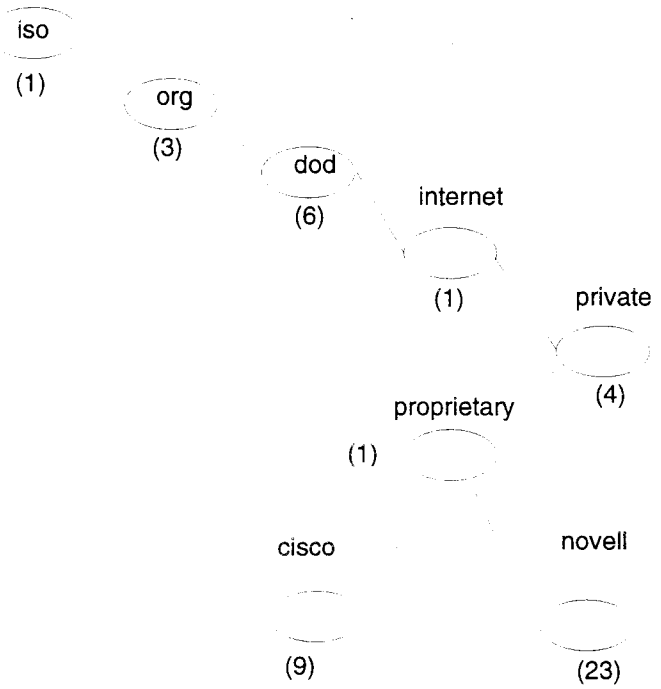
Internet MIB objects are defined using the Internet-standard structure of management information (SMI) and compose a virtual data store on the agent system. This structure is defined by RFC 1155: Structure and Identification of Management Information for TCP/IP-based Internets and amended by RFC 1212: Concise MIB Definitions. RFCs 1155 and 1212 define the structure of management information for SNMP-based management. SNMP agents contain the necessary mechanisms to access the MIB values.

Internet MIBs are organized into MIB modules. A MIB module is a file defining all the MIB objects under a subtree. The foundation module is the standards-based MIB-II module defined by RFC 1213: Management Information Base of Network Management of TCP/IP internets: MIB-II. Besides the standard MIB objects, many hardware manufactures have defined their own extensions to MIB-II. Some examples are HP, IBM, Novell, Cisco Systems. These MIBs are usually defined as proprietary.

As one may expect, the miriade of MIB objects are organized in a hierarchical tree structure. Each branch in the tree has a unique name and numeric identifier. The "leaves" of the tree represent the actual MIB objects. A full object identifier consists of the identifier of each branch along the path through the tree hierarchy, from the top of the tree down to the leaf.

For instance, Novell registers its proprietary MIBs under the numeric identifier 1.3.6.1.4.1.23, having authority over the Novell subtree; and, Cisco registers its enterprise MIBs under 1.3.6.1.4.1.9, having authority over the cisco subtree, figure 1. The MIB variable names are based on Abstract Syntax Notation One (ASN.1) standard.

Figure 1 Example of MIB Organization



2.4. Fault Management Tools and Strategies

The new dimension in network fault management, enterprise level, requires tools and architectures that allow for integration, correlation, and flexibility. In the network world it is very common to find clusters of network elements that support the Simple Network Management Protocol (SNMP) as well as legacy systems that are either of proprietary nature or do not support open network management protocols.

These clusters usually constitute subnetworks. They are managed by specific network management tools, such as HP OpenView, IBM NetView/6000, NetLabs Manager/AT&T StarSENTRY or network management tools that support legacy systems such as COMMAND/Post.

Network managers organize their networks according to a three-tiered management system. The entire network enterprise can be divided in views, domains or layers. Each

domain has its own management system. Network nodes and devices constitute the first tier (lowest layer in the hierarchy). Individual stand-alone management systems usually monitor regions and constitute the middle layer of the hierarchy. The top layer is controlled by an enterprise management system, also called "manager of managers" which has the responsibility to correlate information from all segments of a large enterprise. An example of this type of solution already available in the market is the AT&T's Accumaster Integrator. It follows the three-tiered approach.

Since a single, central management station might be swamped in a large network and would itself represent a single failure point, modern network design practices include provisions for both peer management relationships between control centers and hierarchical relationships between devices, domains or network elements and their management centers. In general network management systems interoperate in one of the three following relationships:

1. Higher-level management systems — in conformance with "manager of managers" or network management platform relationships.
2. Peer level — sharing control, providing control point redundancy, exchanging critical notifications, and performing other functions.
3. Lower-level management systems — interoperating with managed devices or agents residing within network products and actually affect the management changes or request status.

From the types of interaction above, the trend is to have distributed management and at the same time, hierarchical network surveillance or views. All major vendors have products supporting distributed management. The Open Software Foundation is a vendor consortium defining a vendor-neutral Distributed Management Environment, specifying

an entire management system framework for distributed, multi vendor networks. DME is a set of specifications intended to guide developers in creating interoperable network management systems. HP OpenView is an example of lower layer management tool oriented toward networks supporting SNMP. Several vendors have responded with artificial intelligence tools. G2 real time expert system and NetExpert are examples of tools that offer some degree of flexibility to perform functions required by a manager of managers. They provide the capability to create mechanisms for alarm correlation. Other less main stream approaches consider the use of neural network-based pattern matching technology to the same end. CORRELATOR from Applied Computing Devices' uses such strategy.

2.5. Managers and Agents

1. A manager system executes network management operations which monitor and control agent systems. The implementation of these network management operations is called the manager.
2. An agent is the interface to a managed object. An agent system is a device, such as a host, gateway, terminal server, hub, or bridge, that has an agent responsible for performing network management operations requested by the manager.

The Simple Network Management Protocol (SNMP) communicates management information between a manager and an agent. Managers invoke an SNMP client on their local computer, and use the client to contact one or more SNMP servers that execute on remote machines. SNMP uses a fetch-store paradigm in which each server maintains a set of conceptual variables that include simple statistics, such as count of packets received, as well as complex variables. SNMP messages either specify that the server should fetch values from variables or store values in variables. SNMP permits the following activities:

1. A manager can retrieve (get) management information from an agent. The manager sends requests for information to the agent, and the agent sends back replies containing the information requested.
2. A manager can retrieve the name and value of the next instance in the managed object using the get-next operation.
3. A manager can alter (set) management information on an agent.
4. An agent can send information to the manager without an explicit request from the manager. Such an operation in SNMP is called a trap.

Traps proactively alert the manager of changes that occur on the agent system, such as a reboot. The agent knows which manager system to send traps to through a configurable trap destination. Traps however have some limitations. SNMP neither defines the mechanism for where a Trap should be sent, nor explains what the agent should provide as part of a Trap. Thus, Trap is implementation specific. Even more importantly, Traps can only monitor foreseeable events. In other words, Traps can only report on preprogrammed events; if a different failure occurs, the Trap will report it incorrectly or not at all.

One can use a proxy system to allow SNMP access to nodes which do not support SNMP. A vendor wishing to migrate its network management scheme to SNMP, for example, but managing devices with a proprietary protocol can implement an SNMP proxy to manage those devices in their native mode. The SNMP proxy acts as a protocol converter, translating the SNMP manager's commands into the proprietary scheme. When one configure a proxy, the proxy agent receives SNMP request and forward it to the requested node using a non-SNMP protocol. How the proxy gets information from the target node depends on the target, figures 2 and 3.

Figure 2 Data Collection

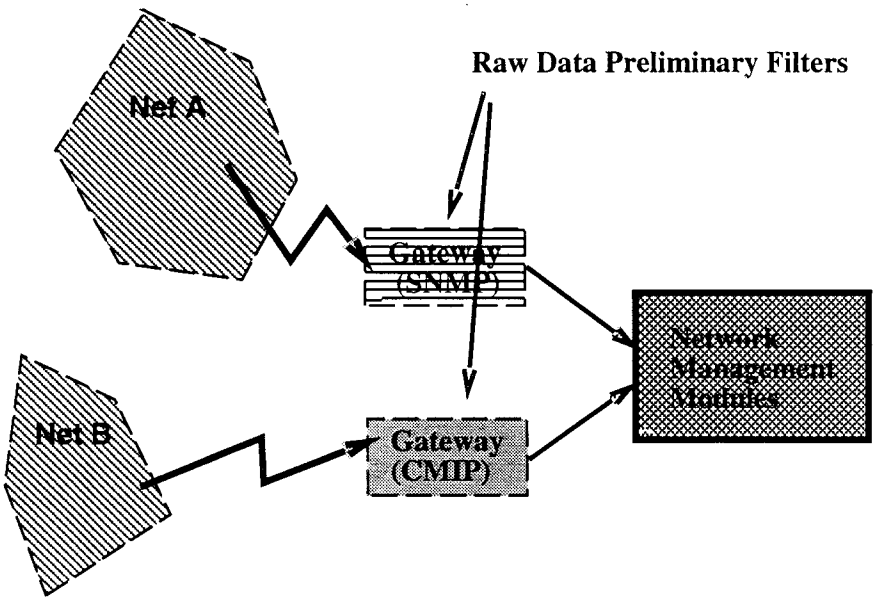
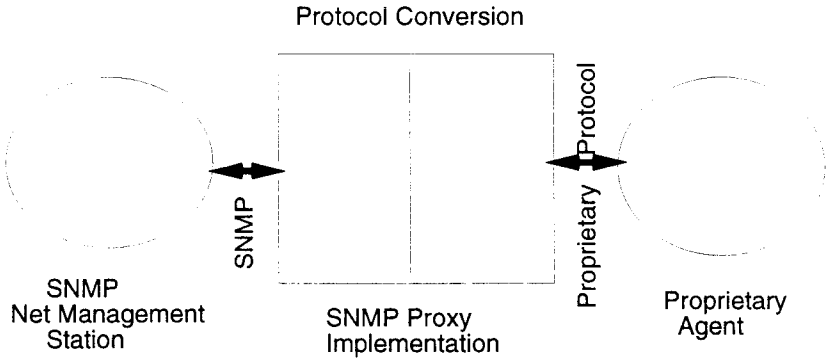


Figure 3 Protocol Conversion



In order to fix some weaknesses of SNMP, the SNMP Version 2 is being created. SNMPv2 brings some enhancements to SNMP mostly on issues related to a reliable security mechanism. In addition to security, SNMPv2 also features two new PDUs and manager-to-manager capability.

The Common Management Information Protocol: CMIP is the basic protocol for message exchange between management modules in the OSI proposal for network management.

Chapter 3. A Fuzzy Logic Approach to Network Management

3.1. The Approach

3.1.1 Introduction

Fuzzy logic is being used in this research to provide a means to correlate common knowledge of network operation. From the moment an event message is displayed at the network monitor's screen, until an action is taken, the process of human thinking is working. The message codes at the screen carry qualitative information and so does the way of human thinking, which most of the time is expressed by carrying a certain degree of relative information (individual operators have different levels of expertise). The informal protocol (defined by the interaction between operators and alarms on the workstation's screen) consists of a set of conditional "if-then" statements where the first part of each contains a so-called condition (antecedent) while the second (consequent) part deals with an action (control) that has to be taken. Therefore, it conveys the human strategy, expressing which reaction to pursue when a certain state of the network is observed. The rules indicate how the operator perceives the composition of alarms (incoming information) that ultimately define the state of a switch/network by variations on quality of service measures.

We use the popular Multiple Attribute Decision Making (MADM) mechanism to manipulate network information. The innovation of handling network information in this research is the formulation of the MADM problem as a fuzzy MADM problem.

As opposite to crisp MADM problems, in our research we do not obtain final ratings as real numbers. Instead, the scores of network elements as function of reported events are expressed by fuzzy sets. As a result, the final ratings are expressed by fuzzy sets. We have also introduced the concept of reference fuzzy set. A ranking method between

fuzzy sets compares the score of the fuzzy MADM with the reference fuzzy set. The result is an indication of the variation of QOS and is used by operators in their decision making process. This number is also used by network management systems to adapt their pooling cycle based on this preliminary QOS information.

3.1.2 Multiple Attribute Decision Making

Multiple Attribute Decision Making (MADM) refers to making selections among several courses of action in the presence of multiple attributes. The decision to be made is to close observe one or another network element depending on the QOS ranking that they have as a function of network management information that has been reported to the NCC. Multiparametric decisions cannot be made straight forward. The imprecision or difficulty level come from different sources such as:

1. Unquantifiable information.
2. Incomplete information.
3. Nonobtainable information.
4. Partial ignorance.

Here the key concept is to take into account the difference between crisp and fuzzy MADM problems. According to [8], for a crisp MADM problem the final ratings are expressed as real numbers. The ranking order can be easily obtained by comparing these real numbers. In a fuzzy MADM problem the score with respect to the network information available is expressed by a fuzzy set. As result the final rating is expressed by fuzzy set. Obtaining the ranking order of these fuzzy sets is not a trivial task. Reference 8 presents several ways to approach fuzzy MADM problems and obtain the final decision by ranking its results.

3.1.3 Proposed Ranking Method

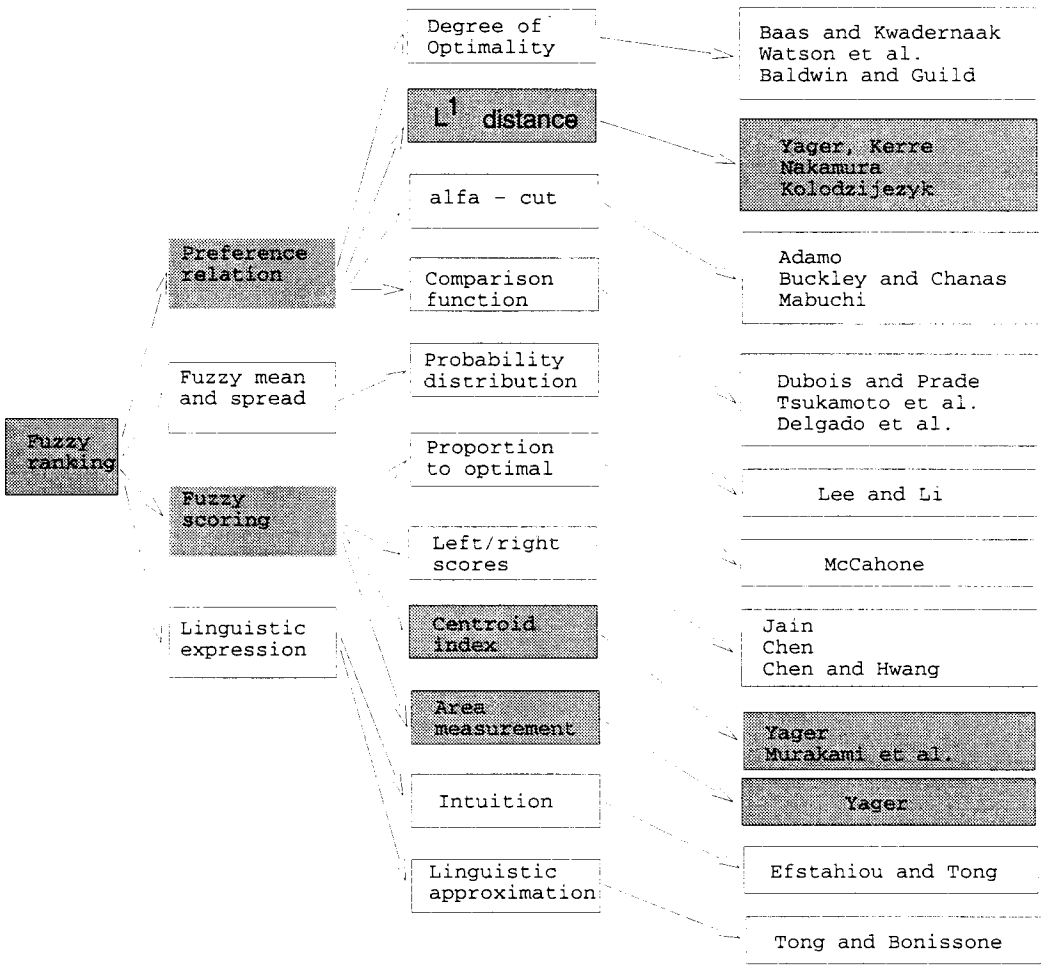
When fuzzy data are incorporated into the MADM problem, the final ratings of the alternatives are no longer crisp numbers; they are fuzzy numbers [8]. The ranking between the fuzzy sets can be viewed as variations on quality of service (please refer to fuzzy set definitions). Since a fuzzy number (identifying a QOS measure) represents many possible numerical values, each having different membership, it is not easy to compare the final ratings. The final rating (relative to a reference set) ultimately determines in which direction is the particular measure of QOS moving, either improving or worsening QOS. To resolve this problem many researchers have proposed fuzzy ranking methods which can be used to compare fuzzy numbers.

Reference [8] analyzes some 20 ranking methods and classify them into three major classes according to the means employed by each method. There are preference relation methods, fuzzy mean and spread methods, fuzzy scoring (or direct comparison) methods, and linguistic methods (see figure 4). According to reference [8], each main class is further subdivided based on the technique used. For instance, methods using degree of optimality are subclass of the preference relation class, methods using centroid indexing are a subclass of the fuzzy scoring class, and methods using linguistic approximation are a subclass of the linguistic methods class.

Each method has pros and cons. Due to the simplicity of the method, our model is implemented using $\mathbf{L}^1$ distance as a preference relation.

We have defined a fuzzy set to represent network element's initial condition. Instead of using Yager's or Kerre's reference we use the defined initial condition to compare fuzzy sets.

Figure 4 Taxonomy Of Fuzzy Ranking Methods from [8].



3.1.3.1 Ranking using L^1 distance The L^1(l^1 for discrete variables) distance between two fuzzy sets M and N is defined as:

$$d(M,N) = \int_{-\infty}^{+\infty} \left| \mu_M(x) - \mu_N(x) \right| dx$$

for continuous variable ranges (i.e. continuous membership function), and

$$d(M,N) = \sum_{i=0}^{k} \left| \mu_M\left(x_i\right) - \mu_N\left(x_i\right) \right|$$

for discrete variable ranges (i.e. discrete membership function).

3.2. Definition of Fuzzy Sets

The decision making process can be handled efficiently by using fuzzy sets and operations on fuzzy sets and matrices. In order to do so we have to:

1. Define a library of fuzzy sets maintaining one to one relationship between fuzzy sets and message codes. This process of assigning fuzzy sets to message codes is called quantification. There are many ways to perform quantification of values of variables into fuzzy sets. Some of them are: use of membership function, experts direct assessment and use of linguistic variables.
2. Define a library of fuzzy sets for load distribution. It can be done by segmentation of the range of node utilization. Each level of node utilization is further quantified into a fuzzy set, maintaining the one to one relationship.
3. Define a library of fuzzy sets for classes of elements of the network. Each class is quantified into a fuzzy set.
4. Define a library of fuzzy sets for time of day. It is done by segmentation of days into discrete amount of hours. Each windows defined this way is quantified into a fuzzy set.
5. Define weights according to the relative contribution of each element above. A message code indicating serious problems receives higher weight as opposite to message codes indicating minor problems in the network.
6. Define a ranking method. In order to generate the final measure the output of the multiple attribute decision matrix has to be compared to a pattern that is assumed to be the one planned by the network engineers. Examples will be provided in this thesis to clarify the notion of the ranking method.

3.2.1 How Fuzzy Sets are Created

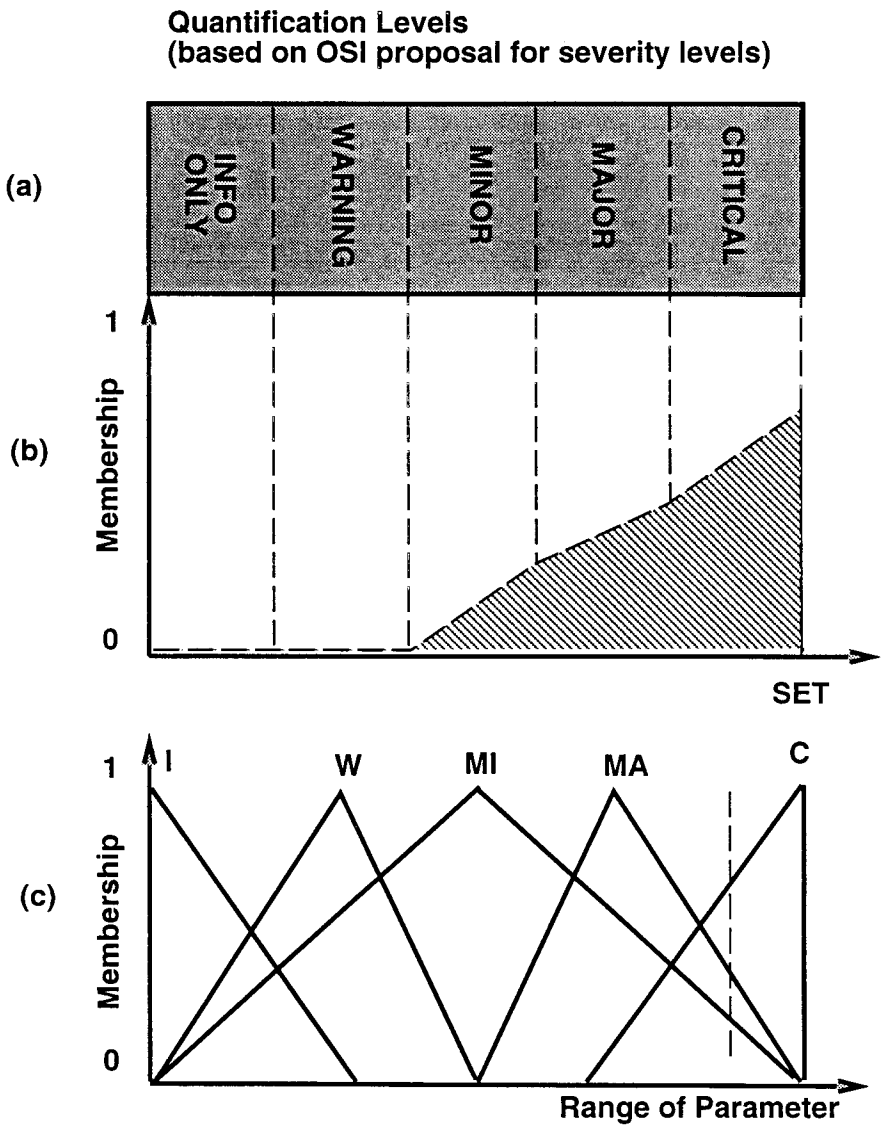
The qualitative information brought by message codes is mapped into a fuzzy set. This mapping is previously done by network operators and designers, which have the knowledge and working experience to correlate the measured information with its effects on network's performance. This mapping is a declarative knowledge, in other words, the operator's experience. The procedural knowledge is obtained by manipulating the fuzzy sets. In doing so, we are, in fact, emulating operator decisions when facing network problems.

The network operation should not be dependent either on the specific shift or operator, and therefore the correlation process through fuzzy sets should not be affected by the individual operator's ability to identify trends and or priorities in dealing with network events. In this research every fuzzy set is created by understanding the impact that an event originator of a message has in the quality of service. Example, in the case of the network event entering buffer congestion, the corresponding fuzzy set must reflect the relation between buffer congestion and QOS.

All pre-defined messages must be mapped into fuzzy sets. Because topology and time of day are very important in the network's health assessment, they are also converted into fuzzy sets. The elements of information above are the most common ones used by operators in their decision making process.

As required by most of MADM problems, weights and a ranking method are established. For instance, a problem reported by a node in the backbone may have greater impact in QOS than if similar problem happened in a node that belongs to a Local Area Network (LAN).

Figure 5 Model Of Fuzzy Set Applied To Network Environment



The mapping of variables (symbolic, numeric, etc) and fuzzy sets is done by quantification. Figure 5–a shows an example of quantification levels. These levels are based on OSI proposal for degrees of severity associated to network events. Figure 5–c shows membership functions to quantify variables into elements of a fuzzy set. The doted line gives an example of a measurement and its mapping into a fuzzy set. The resulting mapping is shown in figure 5–b.

Instead of using membership functions, one can use linguistic variables as shown below. Table 1 defines a possible mapping function for the linguistic representation of certain network events [24]).

Table 1 Linguistic Functions.

Fuzzy ling. values	Fuzzy Set						
high	0	0	0.1	0.3	0.7	0.9	1
medium	0	0.2	0.7	1.0	0.7	0.2	0
low	1	0.9	0.7	0.3	0.1	0	0
unknown	1	1	1	1	1	1	1
undefined	0	0	0	0	0	0	0
aveg high	0	0	0.3	0.5	0.85	0.95	1
very high	0	0	0	0.1	0.5	0.8	1
h-chance	0	0.1	0.5	0.7	0.9	1	1
s-chance	1	1	0.9	0.8	0.5	0	0
vs-chance	1	1	0.5	0.3	0.1	0.1	0

We can think of this mapping between network events and corresponding impact over the QOS as a very primitive level of correlation. The correlation here is between network event and variation in enterprise's QOS.

3.2.2 The Moving Window and Pool of Fuzzy Sets

Every time that an event happens, it's corresponding fuzzy set is sent to a pool of fuzzy sets associated to its network element. Along with the event fuzzy set are also sent the time of day and topology sets. The pool of events changes in size according to the amount of events associated to a certain network address. The QOS variation related to the network address depends on the quality of events being reported.

If a node has high activity, in other words, is related to several events, it generates many message codes. Take for instance the following fuzzy sets:

$$F_1 = \left(a_1, b_1, c_1, d_1, e_1, f_1\right)$$

$$F_2 = \left(a_2, b_2, c_2, d_2, e_2, f_2\right)$$
$$F_3 = \left(a_3, b_3, c_3, d_3, e_3, f_3\right)$$

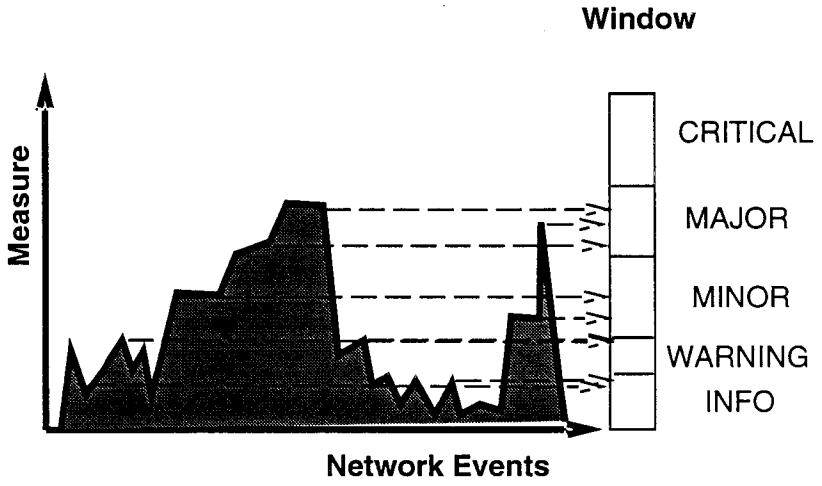
Fuzzy sets F_1, F_2 and F_3 together define a matrix or pool of events. The impact of each event in the course of changing the QOS is not equally important, hence they have different "weights".

Network events are removed from the pool either by opening a trouble ticket and fixing the problem, or by defining a time based moving window.

In this work we define the following states: critical, major, minor, warning and information only. The scale used to measure variation in QOS is divided into five sectors and each sector range is assigned to one state. For instance, in a QOS scale ranging from 0 to 7, one can assign any QOS variation between 0 and 1 as "information only".

The model adjusts itself to different requirements originated by different networks. Take for instance a network management system monitoring networks carrying applications under different QOS. To make sure that different requirements would be taken into account we should assign QOS variation ranges according to individual application requirements. For instance, a network carrying critical business applications may not tolerate QOS variations between 0 and 1 as "information only". They should instead be reported as "Warning" (figure 6).

Figure 6 Mapping Between Measures And States



3.2.3 An Example Of Fuzzy Logic Application

Lets assume that the QOS measure under surveillance is delay. To be more specific, lets focus on transaction response delay as the QOS measure.

Vector s is representing individual fuzzy sets = $\{s_i\}$ and fzP_D = fuzzy pattern due to effects on QOS.

By looking at delay as the typical quality of service measure, the fuzzy patterns could be described as bellow:

$$fzP_D = \left\{ \mu \begin{pmatrix} d \\ s_i \end{pmatrix} \right\}$$

μ is the membership function / fuzzy set, i is the set indicator, then the variation in QOS can be obtained based on:

- $fzQOS_D$ = fuzzy set indicating effect on application delay.
- w_D = weight vector to act on fuzzy pattern assembled after events that may affect delay.

$$fzQOS_D = w_D \text{ o } fzP_D$$

$$fzQOS = \left(fzQOS_D\right)$$

Every time a new message code arrives, a new fuzzy pattern is generated by adding or deleting lines in the matrix. The continuous variation of this pool is used to inform the network operator of fluctuations or trends in network operation (figure 17).

Next we give a numerical example to illustrate how to approach quality of service variations. In this example we assume network operation during peak-hour. The steps taken are as follows:

1. Node belonging to the backbone reports that it is slowing down due to congestion.
2. At arrival at NCC, the message is mapped into the corresponding fuzzy set and added to the pool of fuzzy sets associated to this particular node. Node topology and time of day are also mapped into fuzzy sets and added to the pool. The corresponding weight vector is w_D = [0.5 0.3 0.2 0.1 0.1 0.1 0.1].
3. The operation between weights and fuzzy sets is: $fzQOS_D = w_D \text{ o } fzP_D$

$$= [.5 \quad .3 \quad .2 \quad 0.1 \quad 0.1 \quad 0.1 \quad 0.1]o \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 \\ 0 & 0 & 0.1 & 0.3 & 0.5 & 0.5 & 0.5 \\ 0 & 0 & 0.1 & 0.3 & 0.7 & 0.7 & 0.7 \\ 0 & 0 & 0.3 & 0.3 & 0.7 & 0.9 & 0.9 \\ 0 & 0 & 0.1 & 0.3 & 0.8 & 0.9 & 1 \\ 0 & 0 & 0.2 & 0.3 & 0.7 & 0.9 & 1 \end{bmatrix}$$

4. The result is = {0,0,0.1,0.2,0.2,0.2,0.2}. This is a fuzzy MADM and represents the variation in QOS. At this point we know the fuzzy MADM output. We do not know the trend followed by its corresponding QOS. The trend in QOS is achieved by ranking this vector against the pre-defined reference vector. The result, a measure of variation, is further mapped into windows of severity.

5. The process is repeated for every new arrival. A moving time window is established in order to flush old alarms and their fuzzy sets.

Table 2 illustrates the behavior of QOS measures.

Table 2 An Example of QOS Measures and Their Possible General Behavior

QOS Measure	General Behavior
Throughput	Not changed
	Increasing
	Decreasing
	Decreasing fast
Delay	Not changed
	Decreasing
	Increasing
	Increasing fast
BER	Not changed
	Decreasing
	Increasing
	Increasing fast

The underlying correlation when handling network measures using fuzzy sets can be better explained by the following example: The declaration "high CRC" depends on the CRC count threshold that has been set in the configuration table. There is no formal way to exactly quantify how much a network/subnetwork is affected by a noise line, port or controller card malfunction. Rather we rely on the experience of the operators to assess the impact of the measures and their relationship to the network element's health. This procedure represents a classical situation of multiple attribute decision making.

Another example: It is not easy to quantify how much a high error rate on one or more links degrade the QOS of applications running through these links. By the same token, the relation between message code reporting high buffer occupancy and the slow down suffered by the corresponding network element is not a deterministic process. Data

uncertainty and system uncertainty are often handled in a similar manner by probability theory and statistics. A precise evaluation of the real time network health would demand a tremendous computing effort.

3.3. Manipulating Fuzzy Sets

3.3.1 Review of Operations on Fuzzy Sets

This section gives a short overview on operations involving fuzzy sets. The relationships of fuzzy subsets A and B of X having membership values $\mu_A(x)$ and $\mu_B(x)$ for $x \in X$, respectively, are listed as follows:

1. A is equal to B, A=B

$$\mu_A(x) = \mu_B(x)$$

for all $x \in X$

2. A is a complement of B, A = $\bar{B}$

$$\mu_A(x) = \mu_{\bar{B}}(x) = 1 - \mu_B(x)$$

for all $x \in X$

3. Empty fuzzy set $\emptyset$ is defined by

$$\mu_A(x) = 0$$

for all $x \in X$

4. A is contained in B, $A \subset B$

$$\mu_A(x) \leq \mu_B(x)$$

for all $x \in X$

5. The union of A and B, $A \cup B$

$$\mu_{A \cup B}(x) = max\left[\mu_A(x), \mu_B(x)\right]$$

6. The intersection of A and B, $A \cap B$

$$\mu_{A \cap B}(x) = min\left[\mu_A(x), \mu_B(x)\right]$$

3.3.2 Review of Operations on Fuzzy Matrices

Let A = $[a_{ij}]$ and B = $[b_{ij}]$ be two fuzzy matrices, where $0 \leq a_{ij} \leq 1$ and $0 \leq b_{ij} \leq 1$. Then

$$A \cup B = max\left[a_{ij}, b_{ij}\right] = a_{ij} \vee b_{ij}$$
$$A \cap B = min\left[a_{ij}, b_{ij}\right] = a_{ij} \wedge b_{ij}$$
$$AoB = max_k\left[min\left(a_{ik}, b_{kj}\right)\right]$$

$A \leq B$ exists if $a_{ij} \leq b_{ij}, \forall i, j$

$$\bar{A} = \left[1 - a_{ij}\right] or \bar{B} = \left[1 - b_{ij}\right]$$

where 'o' is the sign for matrix composition. Proof can be found in Kandel (1986), [25].

Chapter 4. The Simple Network Monitor

4.1. Introduction

As stated in the problem definition, the purpose of the Simple Network Monitor is to function as a element that adds performance to a comprehensive fault management system. It establishes a measure of QOS, estimates trends in this measure, provides a reference for scheduling of tasks inside a NCC, and alert operators of any critical trend. Another important added benefit is to serve as a modulating mechanism to manage (control) the overhead traffic saving network resources.

The model is strongly based on principles of Multiple Attribute Decision Making and uses fuzzy sets, matrices and ranking methods to generate the QOS measure.

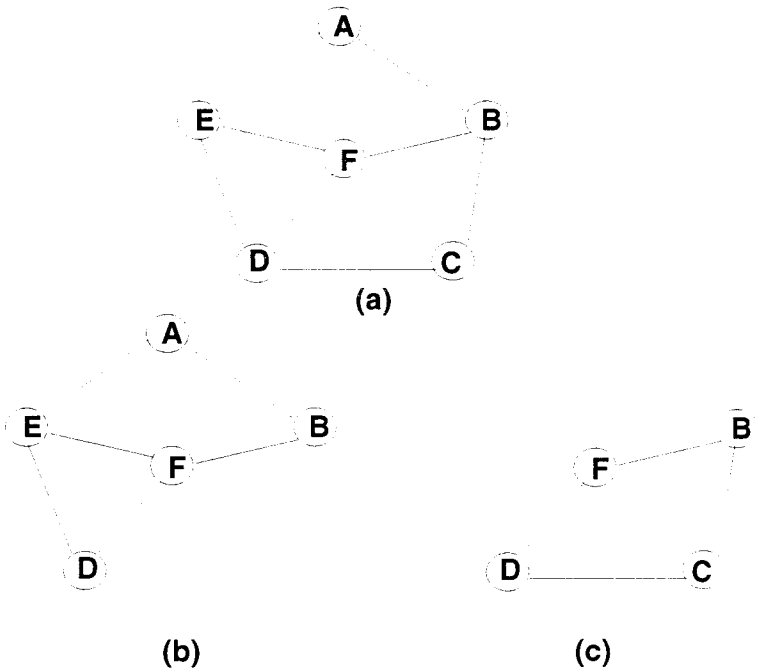
Message codes are the main indicators of events over the network. These events vary from performance related to hardware malfunction. Performance related events are the most difficult to manage due to its possible multiple origins. Hardware malfunctions are in general well defined and the message codes associated to them make possible to easily identify the source of the problem.

Traffic over the network varies from peak to off-peak hours. If a malfunction occurs during peak hours its effects on QOS may certainly be stronger than if the same malfunction occurred at off-peak hours.

Tolerance to network perturbations (variations from the reference state) changes from network to network. The reason is because different networks may be running under different set of requirements. In order to account for such distinct requirements, this model permits the creation of a set of severity windows, one for each network. It means that the same measure of perturbation can be mapped into different states depending on the network being monitored. This is perfect for integrated network management.

As an example refer to the network in figure 7–a. We notice that the same physical network has paths shared by subnets (b) and (c). These subnets may be carrying applications subject to different quality of service (QOS) requirements:

Figure 7 Interleaved Private Networks.



1. SEARS Network: source (B) —> target (D) under constraints (f_2, t_2)
2. FAA Network: source (B) —> target (D) under constraints (f_1, t_1)

Every action taken either by an operator or by an automated fault management system has to take into account the criticality of the operation in both networks. The impact may be quantified as, for instance, the amount of existing applications that would have to be disconnected, or the amount of applications that would be blocked if one tried to establish a connection through the troubled subnet.

The following sections cover the basics for implementation of the model. They describe the quantization process, vector of weights, reference set, extraction of a measure

and mapping it into specific windows (states). Figure 8 provides an overview of the Simple Network Monitor.

4.2. Experimental Work

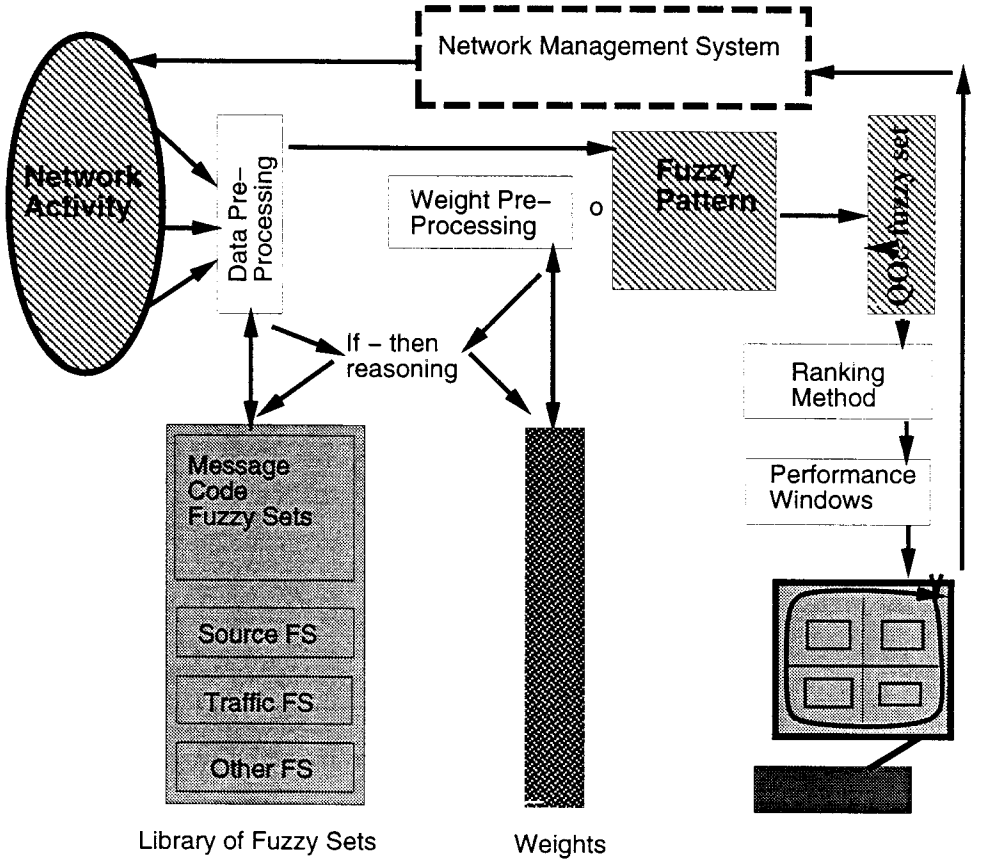
4.2.1 Introduction

A conservative way to establish the minimum QOS at specific node is to use the upper bound requirement among all applications as the minimum acceptable QOS at specific node. In doing so we guarantee that no problem is likely to happen due to lack of QOS that may be requested by certain application. It may happen that applications requiring different levels of QOS are carried over the same physical path.

Variations in QOS have to be correlated to the degree of severity of events. Marshal Rose has proposed a list of severities based on the OSI model. His list is as follows:

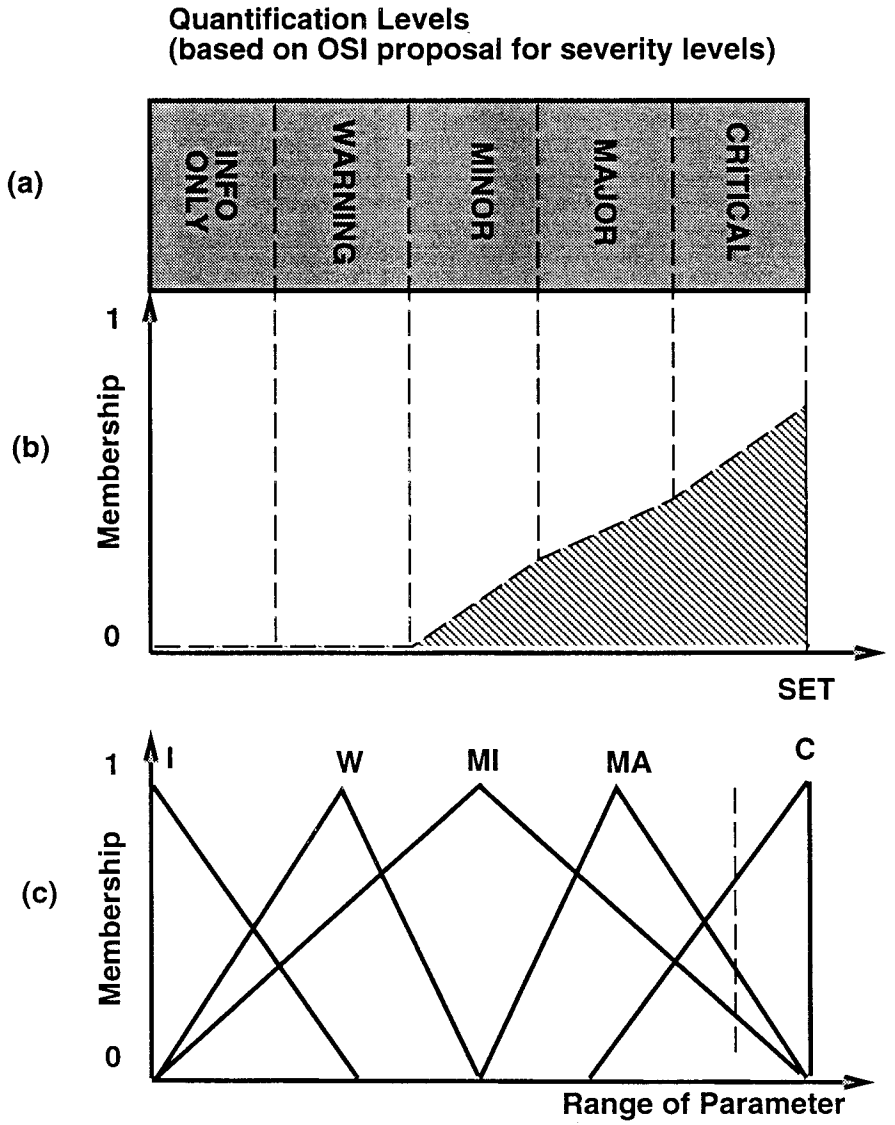
1. **Indeterminate:** Severity level cannot be determined.
2. **Critical:** An error has occurred resulting in the managed object being completely out of service. Therefore, immediate corrective action is required to restore its capability.
3. **Major:** An error condition has occurred resulting in severe degradation of the capacity of the managed object. Urgent action is required to restore its full capability.
4. **Minor:** An error condition has occurred resulting in a reduced capability of the managed object. Corrective action should be taken as soon as possible.
5. **Warning:** Indicates that a potential or impending error condition exists. Action should be taken as soon as possible to avoid an increase in the severity level.
6. **Cleared:** Equipment clears itself; any condition is now gone.

Figure 8 Simple Network Monitor.



The severities used by the SNM are to alert operators to critical trends. They were inspired on the list proposed by Marshal Rose and are as follows: CRITICAL, MAJOR, MINOR, WARNING and INFORMATION ONLY (figure 9).

Figure 9 Severity Levels And Fuzzy Set.



Network parameters carry the following uncertainties and assumptions:

1. Network complexity is in general ill defined.
2. Distinct applications may require different QOS.
3. Each layer has its own QOS.
4. Routing tables establish paths capable of providing the same QOS.
5. Congestion control dictates the side-effects on nodes in the network.

6. The main drive to take preventive actions to maintain the network's health is based on operators experience.

4.2.2 Message Code Quantization

Message codes are mapped into fuzzy sets based on operators' experience. In this implementation we grade each element of the fuzzy set based on the message code and its effects on QOS (in this report we use delay as the QOS measure).

In this experience operators take into account the impact that specific event may have on applications running through certain network elements. For instance, the message code "entering buffer congestion", operators are asked to give grades (0 to 10) to each element inside the corresponding fuzzy set. They are asked how they perceive buffer congestion affecting application delay. Example: From 0 to 10, how much do you perceive "buffer congestion" as critical event affecting response time of applications running through this node? How much do you estimate "buffer congestion" as major event affecting response time of applications running through this node? How much do you estimate "buffer congestion" as minor event affecting response time of applications running through this node? Answers are based on operator's experience leading with such event, network and application in the past. This question is asked to a group of operators in order to obtain an unbiased opinion. The resulting sets are averaged and further normalized. Event weights are defined in similar way.

One could use membership functions or linguistic variables, among other methods, to achieve equivalent results. The table below (table 3) lists a sample of events, their weights, and associated fuzzy sets:

Table 3 Sample of Network Malfunction Description

MC number	Message Description	Weight	Fuzzy Set

Table 3 (Continued) Sample of Network Malfunction Description

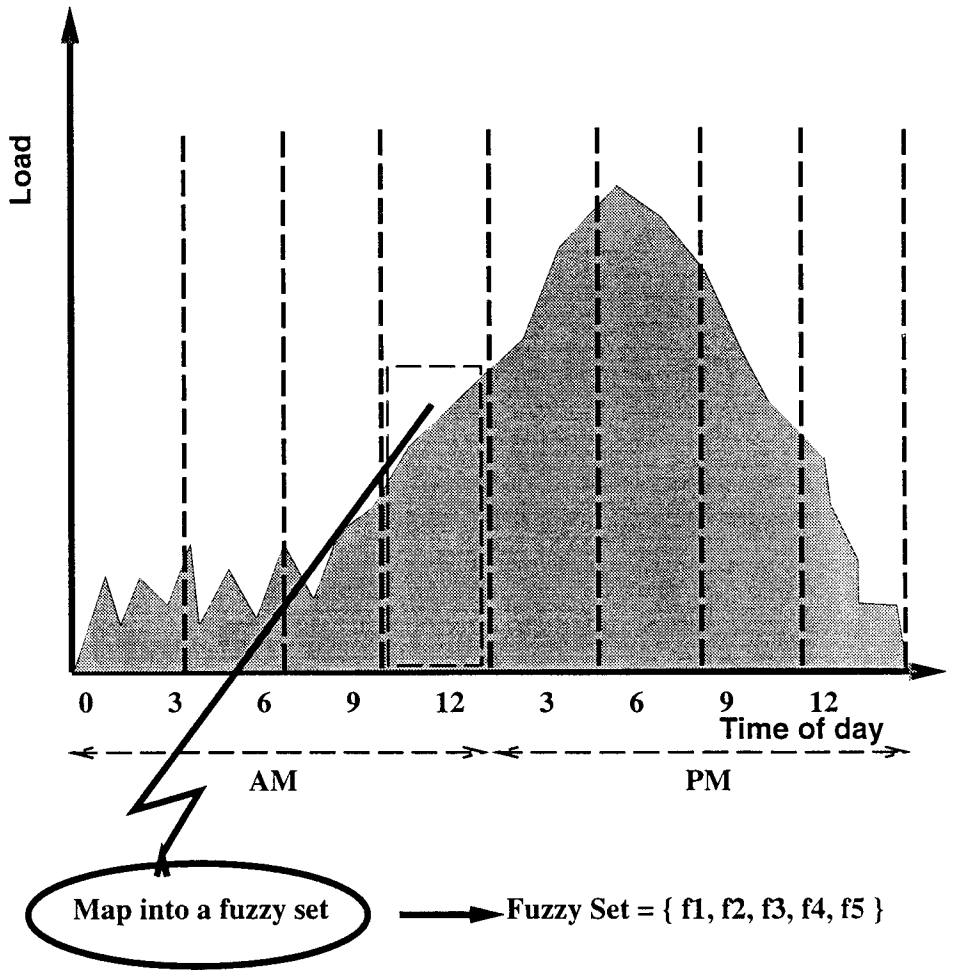
1	Initial Condition	0	[1,.5,.2,0,0,0]
2	Entering buffer congestion	1	[0,0,0.2,0.3,0.7,1]
3	Leaving buffer congestion	0	[1,1,0,0,0,0]
4	Line status DOWN	.85	[0,0,0.3,0.5,0.7,0.8]
5	Line status UP	0	[1,1,0,0,0,0]
7	High CRC count	.77	[0,0,0.1,0.4,0.7,0.6]
6	Retransmissions	.71	[0,0.1,0.3,0.4,0.6,0.4]
8	BER Very High	.70	[0,0,0,0.2,0.7,0.9]
9	BER High	.64	[0,0,0.2,0.3,0.6,0.3]
10	BER Medium	.59	[0,0,.4,.6,.3,0]
11	Buffer error	.50	[.8,1,.3,0,0,0]
12	Time: 0->3	.20	[0.7,1,.5,.1,0,0,0]
13	Time: 3->6	.25	[.7,1,.5,.25,0,0,0]
14	Time: 6->9	.33	[.65,.8,.7,.5,0.3,0.1]
15	Time: 9->12	.41	[.1,.2,.3,.6,.9,0.5]
16	Time: 12->15	.48	[.0,.0,.0,.3,.7,.9]
17	Time: 15->18	.41	[.2,.2,.3,.6,.6,0.6]
18	Time: 18->21	.34	[.5,.7,.4,.2,0.1,0]
19	Time: 21->24	.22	[.7,.75,.6,.1,0,0]
20	Connections I	.23	[.8,.8,.3,.1,0,0]
21	Connections II	.34	[.7,.8,.4,.4,0,0]
22	Connections III	.43	[.6,.7,.5,.3,.3,0.1]
23	Connections IV	.55	[.4,.5,.6,.6,.3,0.2]
24	Protocol error	.19	[.7,.8,.2,0,0,0]
25	Software error	.15	[.9,.9,.2,0.0,0]
26	Line timeout	.12	[.9,.9,.2,.1,0,0]
27	Maintenance alarm	.09	[.95,.9,.3,.1,0,0]
x	Memory buffer error		[0,0.8,0.9,0.2,0,0.7]
x	Environmental alarm		[0,0.5,0.1,0.5,0.5,0.5]

4.2.3 Practical Load Distribution Quantization

The 24 hours period is divided into smaller time windows (figure 10). Peak and off-peak hours make the network more or less sensitive to the occurrence of certain events. For instance, some events that happen during the peak hours receive higher degree of severity than if they happened during off-peak period. During each time

windows, the network element is exposed to certain load distribution. Following the example of message code mapping, traffic intensity windows are mapped into fuzzy sets.

Figure 10 Load Distribution And Fuzzy Set.



In a X.25 packet switched data network, a crude approximation on traffic density per switch can be obtained by using Jackson's Theorem for open network of queues. Assuming randomization at each node and Kleinrock's approximation law, the effective traffic at each node is aproximately

$$\lambda_j = r_j + \sum_{i=1}^{k} \lambda_i \times P_{ij}$$

and the utilization is

$$\rho_j = \frac{\lambda_j}{\mu_j} \leq 1$$

, where P_{ij} is the fraction of traffic that arrives at node j coming from node i, r_j is the portion of traffic generated at node j, λ_j is the effective traffic at node j, and μ_j is the switching capacity at node j. For this implementation, table 4 presents the load distribution over the twenty four hours period:

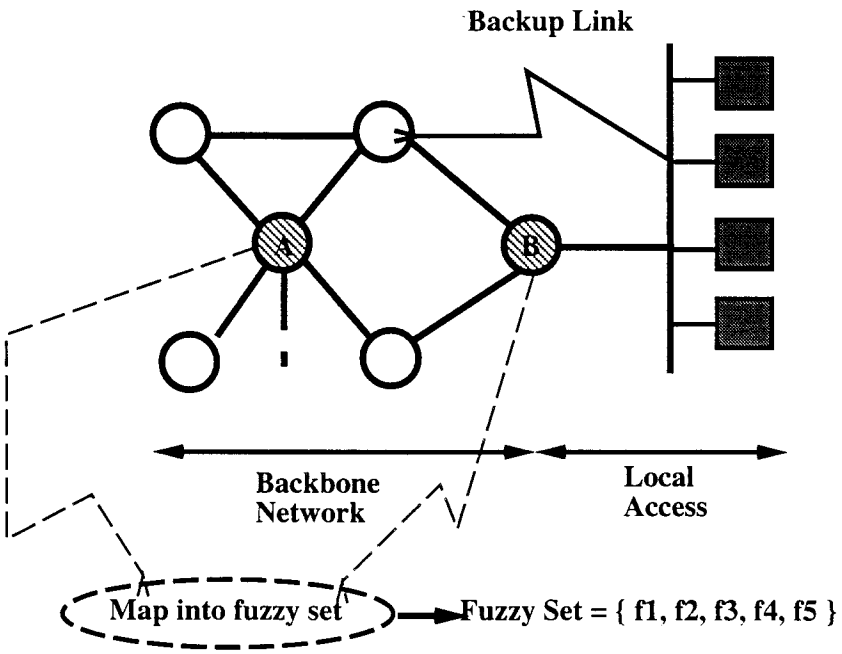
Table 4 Load Distribution For 8 Windows (3 hours each)

Load Distribution (i , i+3)	Traffic
0 -> 3	Very Low
3 -> 6	Low
6 -> 9	Medium
9 -> 12	High
12 -> 15	Very High
15 -> 18	Very High
18 -> 21	Medium
21 -> 24	Low

4.2.4 Information on Topology

The relative importance of a network element also contributes to the network's sensitivity to malfunction affecting such element. If a major switch shuts down, all of its immediate neighbors suffer due to redistribution of traffic. In the figure bellow switch A functions as major node in the backbone network. Switch B also belongs to the backbone, but its main function is to collect local traffic and transfer it to the backbone network. Intuitively a failure in switch B would cause less distress to the network than if switch A had failed (figure 11).

Figure 11 Failure Magnitude



The network element's importance as related to network topology is defined in this application by the number of connections that it has to its neighboring network elements, table 5. Other criteria could as well have been established.

Table 5 A Suggestion To Grade The Importance Of Network Elements.

Connections	Importance of a Switch
6 and up	Very High
3	Low
4	Medium
5	High

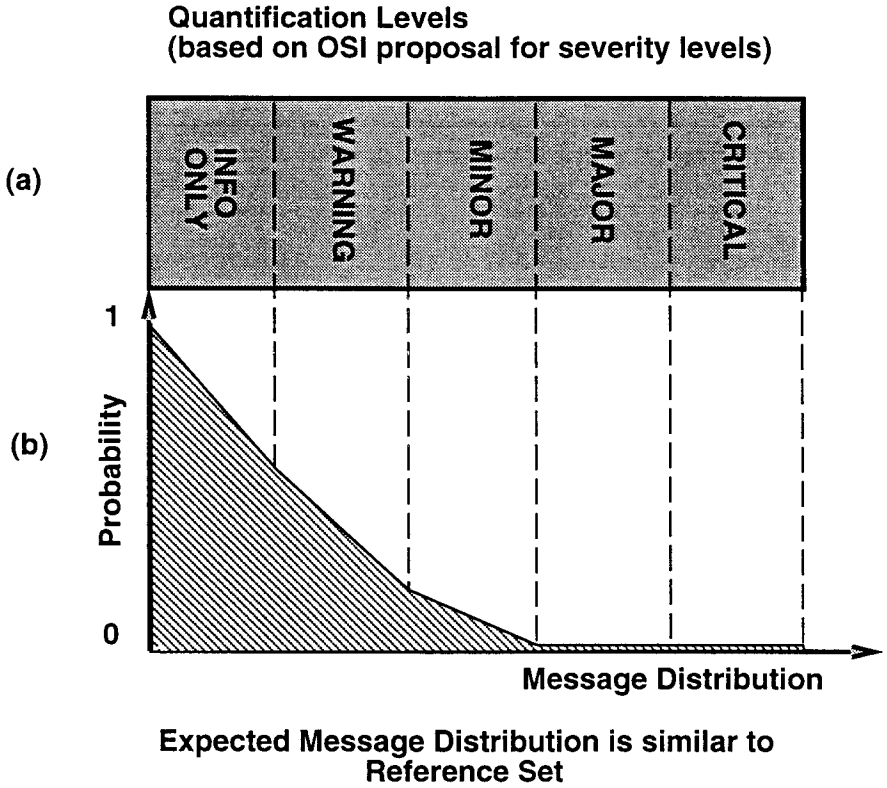
4.2.5 Defining The Reference Set

Reference set is the fuzzy set associated to the node's defined normal QOS. The normal operation is defined by network engineers and operators. During normal operation certain amount of messages is supposed to arrive at Network Control Center. The

distribution of these messages is expected to be skewed toward INFORMATION ONLY, followed by WARNING and eventually MINOR. Messages indicating MAJOR or CRITICAL events are expected to have their distribution near zero or going asymptotically to zero (these two types of events are supposed not to happen if the network is operating according to normal conditions). Figure 12 shows expected message distribution and the reference set.

The reference set can be abstracted as the ideal network state. By comparing other fuzzy sets with the ideal set one is able to identify network trends.

Figure 12 Network Message Distribution And Reference Set.



4.3. Establishment of a Measure

Fluctuations in QOS are quantified by a measure using events reported back to the Network Control Center. This measure of deviations from the ideal state is a random

variable. By naming this measure $X_n, n = 1, 2, ..., k$ one can study its distribution and based on its behavior predict the network element's QOS. By observing the pmf of X one can correlate network instability to the quality of service that the network may experience under certain conditions.

4.4. One Step Prediction of QOS

A very simple way to predict the next QOS based on the current QOS measure and past measures is by using the aging algorithm. The measure of perturbation due to the next event can be predicted as an exponential average of the measured perturbations of previous events. Let x_n be the measure of the n^{th} event, and let τ_{n+1} be our predicted measure for the next event. Then, for α, $0 \leq \alpha \leq 1$, define:

$$\tau_{n+1} = \alpha \times x_n + (1 - \alpha) \times \tau_n$$

, which is an exponential average.

The value of x_n contains our most recent measure; τ_n stores the past history. The parameter α controls the relative weight of recent and past history in our prediction. If $\alpha = 0$, then $\tau_{n+1} = \tau_n$, and recent history has no effect (current conditions are assumed to be transient); if $\alpha = 1$, then $\tau_{n+1} = x_n$ and only the most recent event matters. The initial estimate, τ_0 = average based on past measurements (history). By expanding the formula to $n + 1$ events, we get:

$$\tau_{n+1} = \alpha \times t_n + (1 - \alpha) \times \alpha \times t_{n-1} + ... + (1 - \alpha)^j \times \alpha \times t_{n-j} + ... + (1 - \alpha)^n \times \tau_0$$

4.5. Bayesian Estimation

Lets take the following underlying model in the context of the measurement problem:

1. Density $f_\theta(\theta)$ is called prior (prior to the measurements), and models the traffic of all nodes.

2. Density $f_{\theta}(\theta \mid X)$, called posterior (after the measurements), and models the traffic of all nodes of measured event x.
3. Density $f_{x}(X \mid \theta)$, models all measurements of a particular node of true traffic θ. This density, considered as a function of θ, is called the likelihood function.
4. Density $f_{x}(x)$ models all measurements of all nodes.

The Bayesian model for this problem is a product space $\zeta = \zeta_{\theta} \times \zeta_{x}$ where ζ_{θ} is the space of the random variable $\boldsymbol{\theta}$ and ζ_{x} is the space of the random variable $\mathbf{X}$. The space ζ_{θ} is the space of all nodes and ζ_{x} is the space of all measurements of all nodes.

The problem here is to estimate θ in terms of the n samples x_{i} of $\mathbf{X}$. By using the MS criterion, we obtain

$$\widehat{\theta} = \mathbf{E}\{\boldsymbol{\theta} \mid \mathbf{X}\} = \sum_{-\infty}^{+\infty} \theta \times \mathbf{f}_{\theta}(\theta \mid \mathbf{X}) \times \mathbf{d}\theta$$

and

$$f_{\theta}(\theta \mid \mathbf{X}) = \frac{f(X \mid \theta)}{f(X)} \times f_{\theta}(\theta)$$

In the above, $f(X \mid \theta)$ is the conditional density of the n random variables x_{i} assuming $\boldsymbol{\theta} = \theta$. These random variables are conditionally independent, then $f(\mathbf{X} \mid \theta) = f(x_{1} \mid \theta) \ldots f(x_{n} \mid \theta)$

The estimated value $\widehat{\theta}$ can be an indicator of potential variations of the selected QOS measure.

4.6. SNM Basic Algorithm

It can be divided into 6 major steps or phases according to the following break down:

Phase A: Initialization of variables

Initialize:

Connectivity Matrix

Quality of Service Thresholds (necessary for each sub-network)

Initial node condition (pre existing conditions if necessary)

Weight vector

Message code timers

Phase B: Process Message Code

Receive Message Code

Extract code from Message Code

Determine the corresponding fuzzy set

Extract Information on topology (switch originator)

Determine corresponding fuzzy set

Extract Information on load distribution

Determine the corresponding fuzzy set

Locate switch's fuzzy pattern (previous condition)

Update fuzzy pattern by adding the new fuzzy set

Phase C: Update weights

Locate switch's vector of weights

Locate switch's fuzzy pattern

Identify the importance of the new fuzzy set (switch rows inside matrix)

Update weight vector

Phase D: Quality of service fuzzy set generation

Combine weight and fuzzy pattern to obtain new QOS (new fuzzy set)

Plot current QOS fuzzy set (interface to operator)

Locate fuzzy set reference (planned fuzzy set)

Using ranking method, compare current QOS fuzzy set with the reference

Phase E: Display results

Locate information on switch's last quality of service ranking

Identify thresholds (each subnet has its own windows of tolerance)

Plot network's current topology and display faulty switch with corresponding color according to tolerance windows

Plot all quality of service ranking reported at switch level, including the most recent one (interface to operator)

Perform one step QOS prediction (*) (interface to operator)

Plot predicted QOS and comment on trend (*) (interface to operator)

Plot all quality of service at global level

Phase F: Update side effects (*)

Find all neighbors to switch reporting problem

Identify severity level of quality of service variation for node reporting problem

Determine the corresponding fuzzy set for node's neighbors at level 1

Locate nodes's fuzzy pattern

Update node's fuzzy pattern by adding this new fuzzy set

—> While there is a switch to be updated (chain effect)

Go Back to Phase C

—> end

Look for expired timers

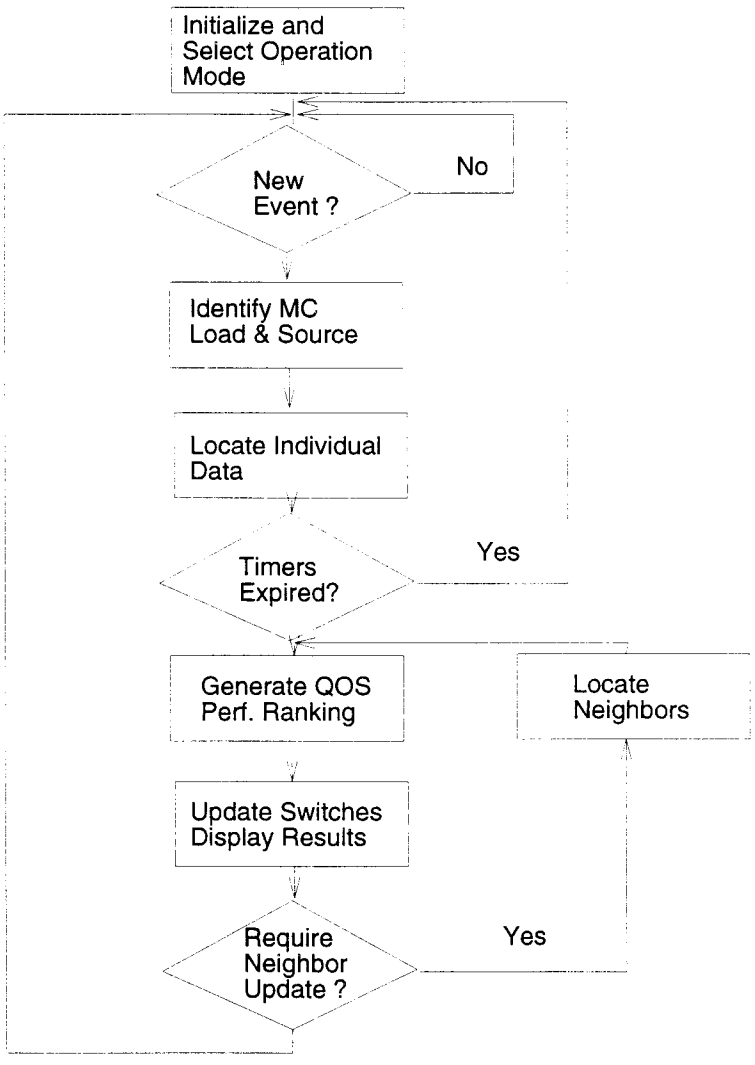
If there is any expired timer fix fault condition associated to such timer

Look for new message codes — back to phase B

If new arrival (Message Code) is from a neighbor of last reported MC preempt the calculation of neighbor effect and proceed from phase B

4.7. SNM Blocks Diagram

Figure 13 Simple Network Monitor Blocks Diagram



The Simple Network Monitor was implemented in Matlab 4.1 (figure 13) and was designed for two modes of operation: interactive or automatic. The interactive mode

is included to help identify patterns and window thresholds before setting it up for automatic operation.

4.8. Special Network Scenario

If an alarm arrives during off-peak hours, most likely, the event being reported is related to hardware failure or environmental changes near to the network element. In this case the load distribution would have low importance in the QOS variation. However, the message code and the source of such message code (network topology) would be considered and operators would still receive the proper warning. The following example illustrates this fact: Suppose that the nominal transaction delay in a path through node X is 10 seconds. Suppose yet that this node operates with double processor, each processor having nominal capacity of 100Mbytes/sec. Assume that the event being reported is the failure of one processor and that the effective traffic at that node is 40Mbytes/sec. If one processor fails, a message indicating processor failure is issued. The remaining processor will handle the traffic (40Mbytes/sec) and no hazards in terms of buffer overflow and delay are going to be reported. Would this be a situation where the SNM wouldn't properly report the potential hazard to network operators? Answer: No.

The SNM always reports potential perturbations over the network. The fact that at the moment of event generation the traffic was low will only reduce a little bit the measure of QOS. Operators will still receive a warning in their screen indicating the potential for damage due to the event "processor failure".

4.9. Summary

Multiple Attribute Decision Making techniques explore the network events and their relative importance toward a decision to be made. Our model uses this idea and handles the combination between a vector of weights and corresponding fuzzy pattern to extract

the quality of service level at specific network element. Each quality of service level obtained is represented by a fuzzy set. To correlate quality of service fuzzy sets with a reference, one has to use a ranking method. A ranking method offers a relative measure, and in this example is used to compare two fuzzy sets. The result, a numeral, represents the relative deviation between the two fuzzy sets.

Chapter 5. Network Troubleshooting

5.1. Introduction

Troubleshooting is not the goal of this research. This chapter was introduced here to indicate ways to approach troubleshooting based on network-related experience. In many practical problem-solving situations, the available knowledge is incomplete or inexact. In cases like these, the knowledge is inadequate to support the desired sorts of logical inferences. However, humans have ways of drawing inferences from incomplete, inexact, or uncertain knowledge and information. Although our knowledge is not complete, we can and do make and use generalizations and approximations that help us summarize our experience and predict aspects of things we don't yet know. Generalizations are often subject to error, and yet we use them anyway.

Probability reasoning methods allow AI systems to use uncertain or probabilistic knowledge in ways that take the uncertainty into account. In addition, probabilistic methods can help us accumulate evidence for hypotheses in a fair way; they are appropriate tools in making "just" decisions. Decision theory, related to theory of probability, provides additional techniques that help to minimize risk in making decisions. One of these techniques to help reduce the risk of bad decisions could be the QOS measure as indicated in this research.

5.2. Correlation

The principle of correlation is to establish common and related causes when network messages arrive at NCC. There are several tools of the shelf that offers capability for event correlation. The full knowledge to achieve correlation always resides with users and network-specific experience.

Take for instance NetExpert from Objective Systems Integrators. NetExpert has a module called Intelligent Dynamic Event Analysis Sybsystem (IDEAS). It contains an expert engine and knowledge base used to analyze events. It correlates events across multiple-managed objects, test thresholds, adds descriptive data, and determines priorities based on user-configurable severity levels. IDEAS surfaces the primary events as alerts to an operator's workstation and suppresses the sympathetic events by writing them into a log file. Another example is correlation using COMMAND/Post. COMMAND/Post provides a very primitive correlation method by using filters. These filters use rules to identify fields in the incoming network messages. For instance, one may select the fields containing the code of the incoming message and the IP address of the network element. A very basic correlation could be to only issue alerts for every 10 messages of specific type originated at the pre-defined IP address.

Knowledge of networking techniques, protocols, and devices is fundamental to implement a reasonably intelligent correlation scheme. The bad side is that fancy correlation over large networks require longer processing time.

The most common and simple types of correlation are based on time and network topology. Correlation based on time is very straight forward and works by focusing on incoming messages during certain time frame. Correlation based on network topology uses the parent — child relationship. This relationship relies on direct connection between objects. For instance, if a line processor card fails, most likely all ports controlled by such card report individual failure messages. This is easily detected at NCC by establishing a filter correlating the line processor card failure with all messages related to lines connected to this card.

Our proposed correlation of events is based on the two simple methods listed above: time and network topology. It can however be used with any type of correlation. In

our scheme we take advantage of the correlation functionality provided by software tools such as NetExpert and COMMAND/Post. Our method is based on quality of service variations across objects. If a wide area service — T1 and T3, for example, fails it will trigger a cascade of message codes from several network elements and applications. The first network element to sense the failure will send a message code to the NCC reporting loss of connection across specific link. The first correlation is by setting a time window based on the arrival of the first message. This time window will be used to identify redundant messages and control the QOS update process. The size of the window is based on network-experience when dealing with similar situation. Basically all messages arriving during this time frame will have their, say, IP addresses parsed. The second type of correlation, network topology, will help to establish the neighbors of the network elements reporting malfunctions. All parsed IP addresses will then be checked against the network topology to identify network elements reporting messages due to the same network event. The quality of service variations are then calculated and attached to the network element event originator. Redundant messages are sent to specific log and not used to update QOS of neighboring network elements. Depending on the situation, QOS may also be update in network elements directly connected to the malfunctioning element.

A tool such as ACCUGRAPH will then retrieve data from the database, including the new update in QOS, and display to users the network topology and QOS (by color coding). Operators will be able to see where the network problems are and QOS (time stamped color coding) degradation trend. Operators will also be able to identify the QOS over specific mission-critical enterprise applications and take preventive actions. Trouble tickets are then populated using a tool such as REMEDY Action Request System.

5.3. Preparing for Diagnosis

Making a decision means choosing among alternative courses of action with or without all the relevant information and often with uncertain information as well. Because of the lack of knowledge of the exact conditional probability distribution for the various possible states of evidence (symptoms) given the various possible states of nature, successful inference networks cannot usually be developed directly from Baye's rule. A reasonable alternative is to develop a hierarchy of "fuzzy" assertions or hypotheses and use substantiated hypotheses at level k to substantiate hypotheses at level $k+1$ Baye's rule can be used directly to substantiate (establish probability values for) level-1 hypotheses from the evidence if the evidence may be regarded as certain. Then "fuzzy inference rules" are used to obtain probabilities for other hypotheses, given the evidence. If there is uncertainty associated with the evidence, then fuzzy inference may be used at the first level as well.

The difficult problem of building an inference network appropriate to a given problem domain can be broken down into simpler steps. The basic steps are the following:

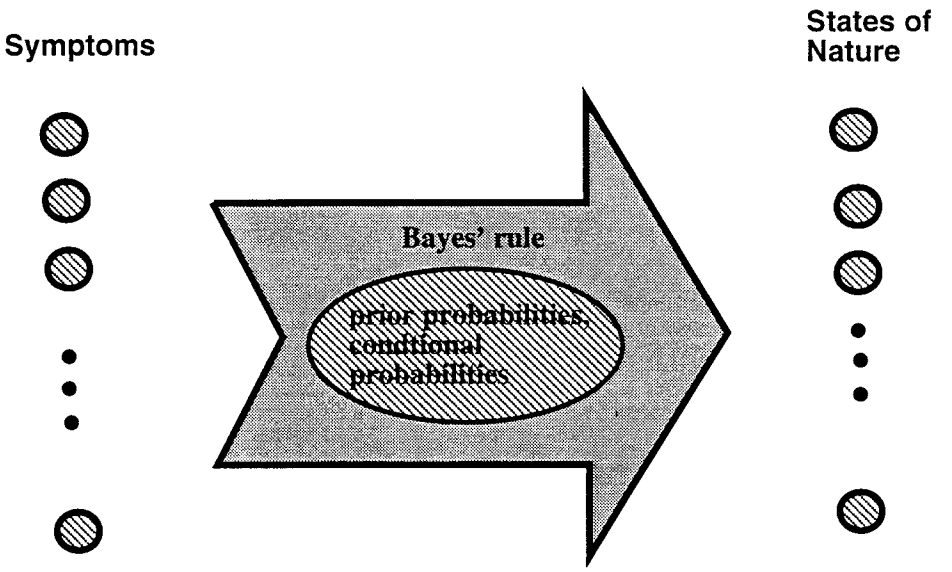
- ☐ determination of the relevant inputs (set of possible evidence or symptoms that can be easily determined — properties of object under study or of its environment),
- ☐ determination of states of nature or decision alternatives (conditions that support certain evidence),
- ☐ determination of intermediate assertions that may be useful in the inference network (several levels of intermediate assertions may be necessary to link symptoms to conditions that support these intermediate assertions). In general these are attributes which are not directly observable, but probabilistically related to the inputs and states of nature in some reasonably understood way. They could also be defined as partial

characterization of the states of nature.

- ☐ formulation of inference links (establishment of logical relationships among inputs, assertions or states)
- ☐ tuning the probabilities and/or the fuzzy inference functions (updating functions used to propagate information through the inference network). In general the relationship and probabilities needed to construct an inference network are provided by experts.

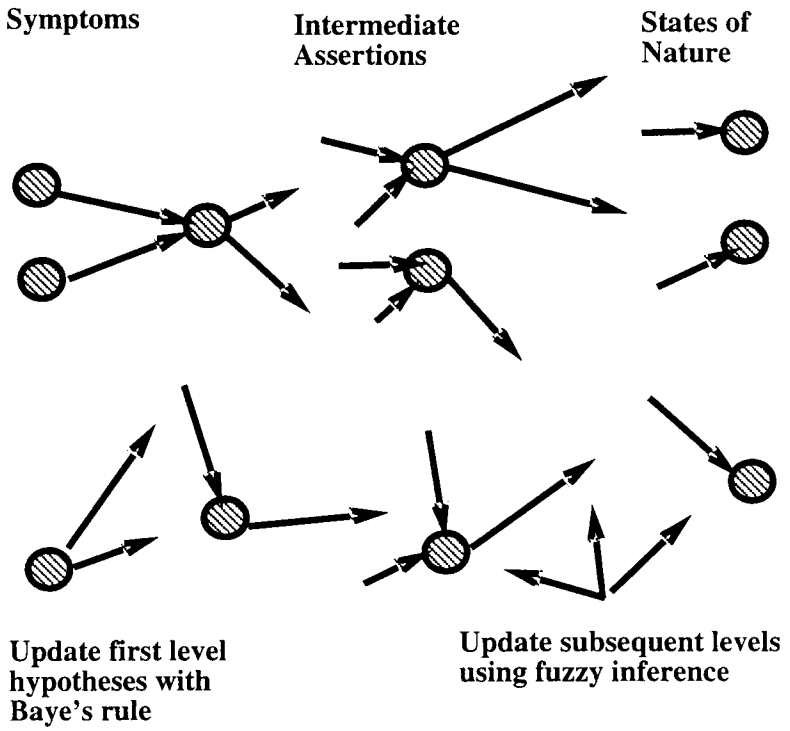
If a network can be developed directly from Bayes' rules, it would follow a scheme like shown in figure 14.

Figure 14 Bayes' Network



If the inference system followed combined Bayes' — heuristic logic, it would have the shape presented in figure 15.

Figure 15 Inference System Based on Combination Bayes' and Heuristic



5.4. Example of Inference Rules

Fuzzy inference rules are functions for propagating probability values. The general form of such a function is:

$$f : [0,1]^n \rightarrow [0,1]$$

Thus a fuzzy inference rule takes some number of n probabilities as arguments and returns a single probability. The choice of f for a particular situation is a modelling decision that requires some understanding of the relationships among the phenomena described by the hypotheses.

Table 6 presents sample of inference rules for propositional calculus and two sets of fuzzy inference rules. The first of these two fuzzy inference rules employes min and max and is called "possibilistic logic". The second assumes a probabilistic approach to the

relationship between A and B. It is so called probabilistic logic rule. The possibilistic logic rule for $A \oplus B$ is xor (a,b) = max(min(a, 1–b), min (1–a, b)). The probabilistic logic rule for $A \oplus B$ is Xor (a,b) = $a + b - 2ab + a^2b + ab^2 - a^2b^2$

Table 6 Inference Rules For Propositional Calculus And Two Fuzzy Logics

A	B	¬A	A&B	A or B	A→B	$A \oplus B$
F	F	T	F	F	T	F
F	T	T	F	T	T	T
T	F	F	F	T	F	T
T	T	F	T	T	T	F
a	b	1-a	min(a,b)	max(a,b)	max(1-a,b)	xor(a,b)
a	b	1-a	ab	a+b-ab	1-a+ab	Xor(a,b)

5.5. Example of Application Using Possibilistic Logic Rules

The application domain is a network with a layered node architecture and functional dependencies as described in appendixes A and B. Data pertained to all nodes is collected at Network Control Center. The problem here is to identify potential causes of poor performance.

In this example (figure 16) we dedicate special attention to the lowest four layers: transport, network, data link and physical. In general a crude classification of problems by layers could be done as follows:

I. Problems at the physical layer tend to manifest themselves as line outages or errors.

II. Problems at the data link, network, transport, and higher layers tend to cause degraded performance but not actual outages and errors.

The symptoms are described by the following statements:

S1: System slow noticed at node B.

S2: Slow at B noticed for all sessions.

S3: Node A hasn't reported any network perturbation.

S4: Node E hasn't reported any network perturbation.

S5: Node B has not reported buffer overflow.

S6: No problems reported with the maximum window for VR that begin in node B.

S7: Threshold for entering slowdown at note B was set above 15% (percentage of free buffers below which the switch will enter slowdown).

The final state of nature whose probability we wish to infer is the possibility of SLOWDOWN threshold being too conservative —> SN.

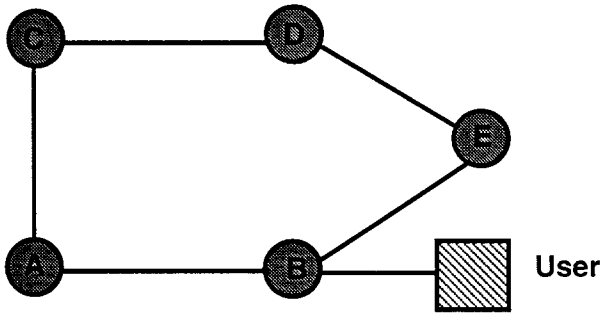
Some intermediate hypothesis are now included. These are first level hypothesis:

H1: Too much traffic.

H2: Outbound links flow constricted.

H3: Node B is in slowdown state.

Figure 16 Sample Network



Strategy: The first step is to eliminate the possibility of physical problems by establishing a bottom-up troubleshooting. The idea is to discard the dependency between service user and service provider. The physical layer is the first service provider and so, by eliminating problems at physical level we reduce troubleshooting efforts by detecting any cascading effect toward upper layers.

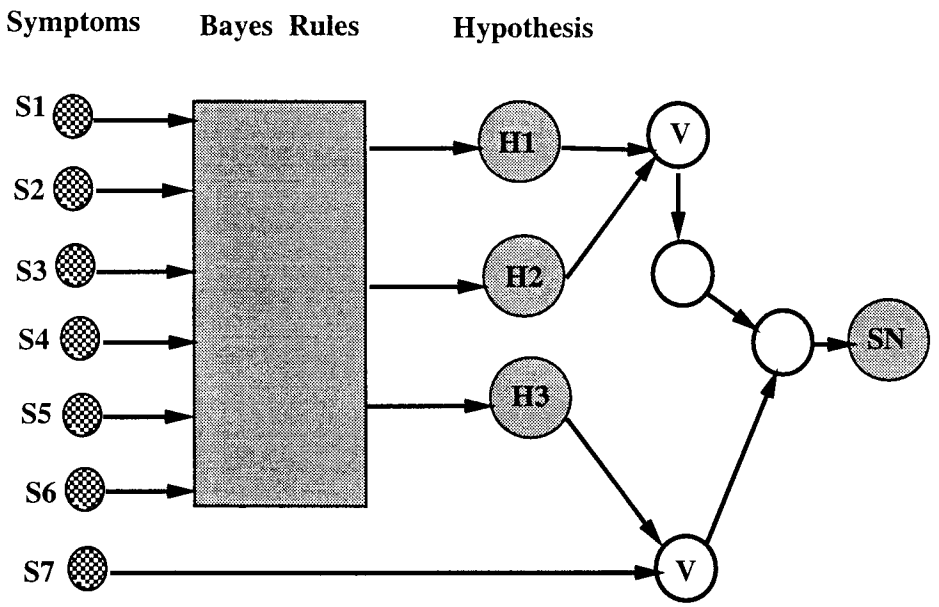
The best sensor of functional problems at physical layer is the amount of network perturbations due to problems at link level. For instance, high CRC check is a good

indicator of problems with link (physical layer). Another good indicator of problems with links is the average queue at each node.

Data is not always available to be used in troubleshooting. Most of the information related to the dynamics of the network is collected based on data logged for accounting and other type of data that is not reported back to NCC in a regular basis. It makes the problem solving a more complicated task and heuristics play a major role. In general any troubleshooting system combines heuristic techniques, analytic models, and fuzzy logic to diagnose problems.

By using Bayes' relations and fuzzy logic rules we may model the dependence between symptoms and assertions (hypothesis): $SN = \neg(H1 \vee H2) \wedge (H3 \vee S7)$

Figure 17 A Probabilistic Inference Network



In the network above (figure 17) each hypothesis is related to one or more symptoms. We choose to express such relationship so that Bayes' rule may be used to establish probabilities for the hypotheses that reflect a particular set of symptoms. This example

presents one option to deal with network symptoms and hypothesis toward diagnose. The trade-off between cost benefit helps to select the most convenient approach.

Chapter 6. Experimental Results

The algorithm to measure QOS was implemented using Matlab 4.1. The advantage of this model relies on the fact that detection of critical trends are easier to realize. The model converts message codes into measures that are further incorporated into a database as part of the set of attributes describing a network element. By introducing a common baseline for measure of perturbations over the network, the decision to establish priorities of one problem over another becomes a partnership between operators and the Network Management System. This feature makes it possible to achieve quicker responses to network malfunctions.

The quality of service fluctuation is indicated by a gradient of colors. The algorithm also allows surveillance at specific node while the SNM goes through its cycle. It permits nodes that require particular attention to be regularly monitored.

Network pooling by the underlying network management tool (HP OpenView) is controlled by the one step prediction QOS. The major benefit is to reduce overhead traffic saving network resources. It freed network management system processing power to focus on network elements that have or are about to have problems.

The data used is based on interviews with network operators. About 200 interactions (message codes) where analyzed by the Simple Network Monitor and the results showed consistency with the expected values. Figure 18 shows a typical SNM screen.

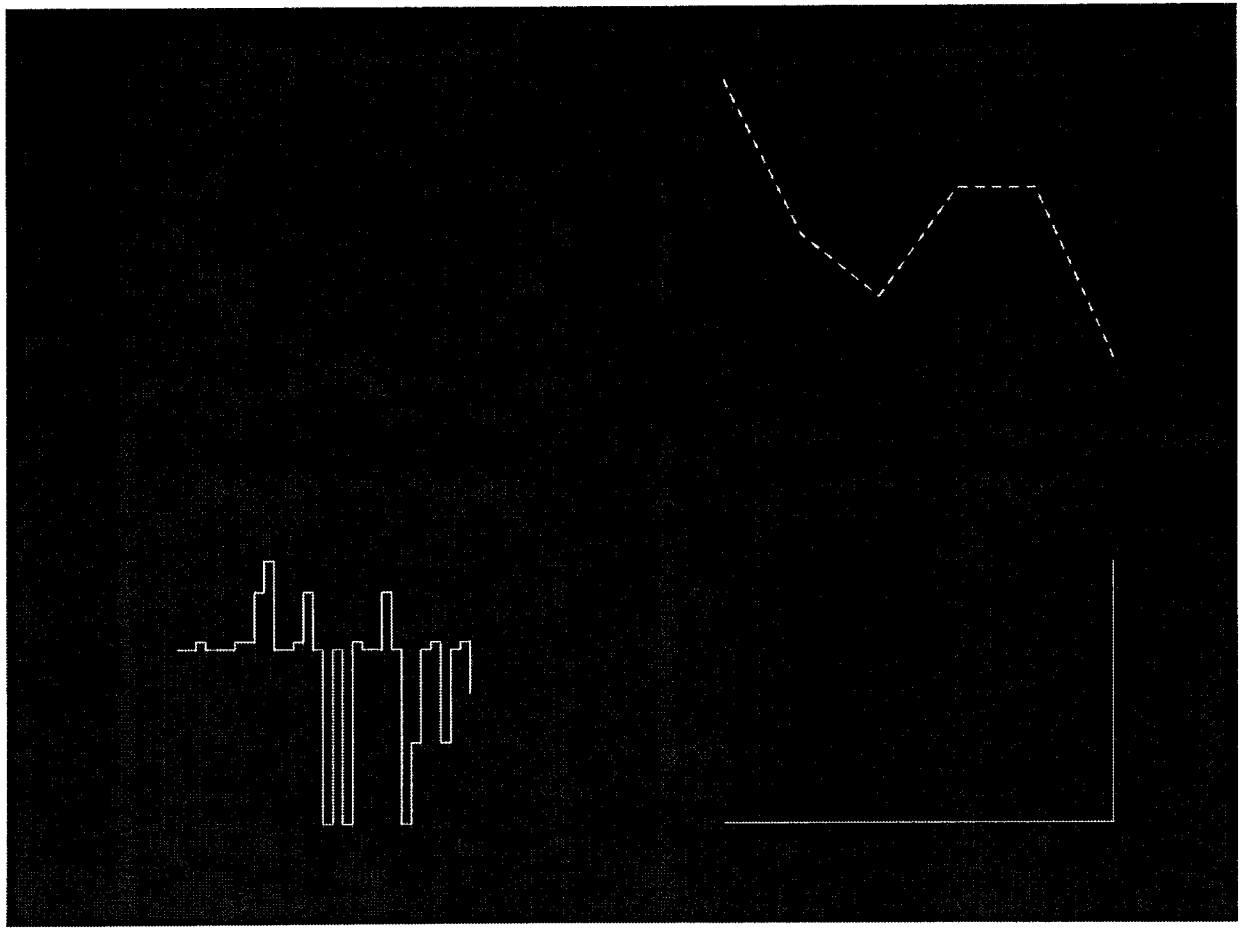


Figure 18 Simple Network Monitor Typical Screen

The next set of tables correlate event reporting and the ranking process performed by the Simple Network Monitor. For the column QOS, the following convention is used: —: small fluctuation, D: Degrades, and I: Improves. Set-1 of message codes indicates the SNM' sensitivity to incoming message codes. The message codes are sent by the same node during fixed time window.

Table 7 Set—1 of message codes and expected QOS variation

Message Code	Load Window	Switch	QOS
Initial Condition	9->12	E	-
Maintenance alarm	9->12	E	D
Software error	9->12	E	D
Protocol error	9->12	E	D
Buffer error	9->12	E	D
BER Medium	9->12	E	D
BER High	9->12	E	D
BER Very High	9->12	E	D
High CRC count	9->12	E	D
Retransmissions	9->12	E	D
Line status DOWN	9->12	E	D
Line status UP	9->12	E	I
Entering buffer congestion	9->12	E	D
Leaving buffer congestion	9->12	E	I

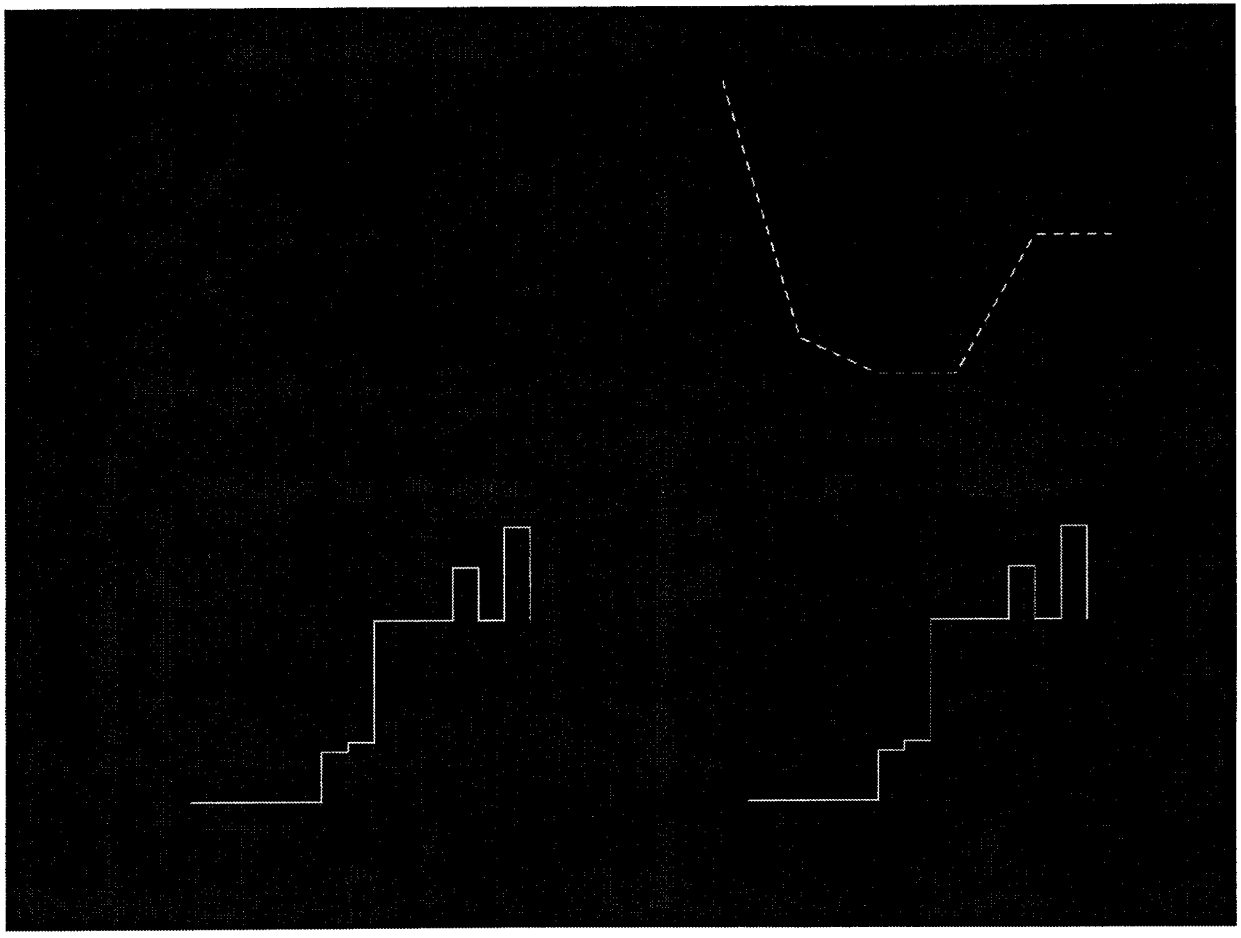


Figure 19 Output from SNM referent to set—1 of message codes

Set-2 of message codes correlates load distribution and its impact over node G. The impact on QOS is partially dependent on the traffic load present at the moment of the message arrival. In this experiment the message code and switch are kept constant for a variable traffic load.

Table 8 Set—2 of message codes and expected QOS variation.

Message Code	**Load Window**	**Switch**	QOS
Initial Condition	0->3	G	-
Protocol Error	0->3	G	-
Protocol Error	3->6	G	-
Protocol Error	6->9	G	D
Protocol Error	9->12	G	D
Protocol Error	12->15	G	D
Protocol Error	15>18	G	I
Protocol Error	18->21	G	I
Protocol Error	21->24	G	I
Protocol Error	0->3	G	-
Protocol Error	3->6	G	-
Protocol Error	6->9	G	D
Protocol Error	9->12	G	D
Protocol Error	12->15	G	D

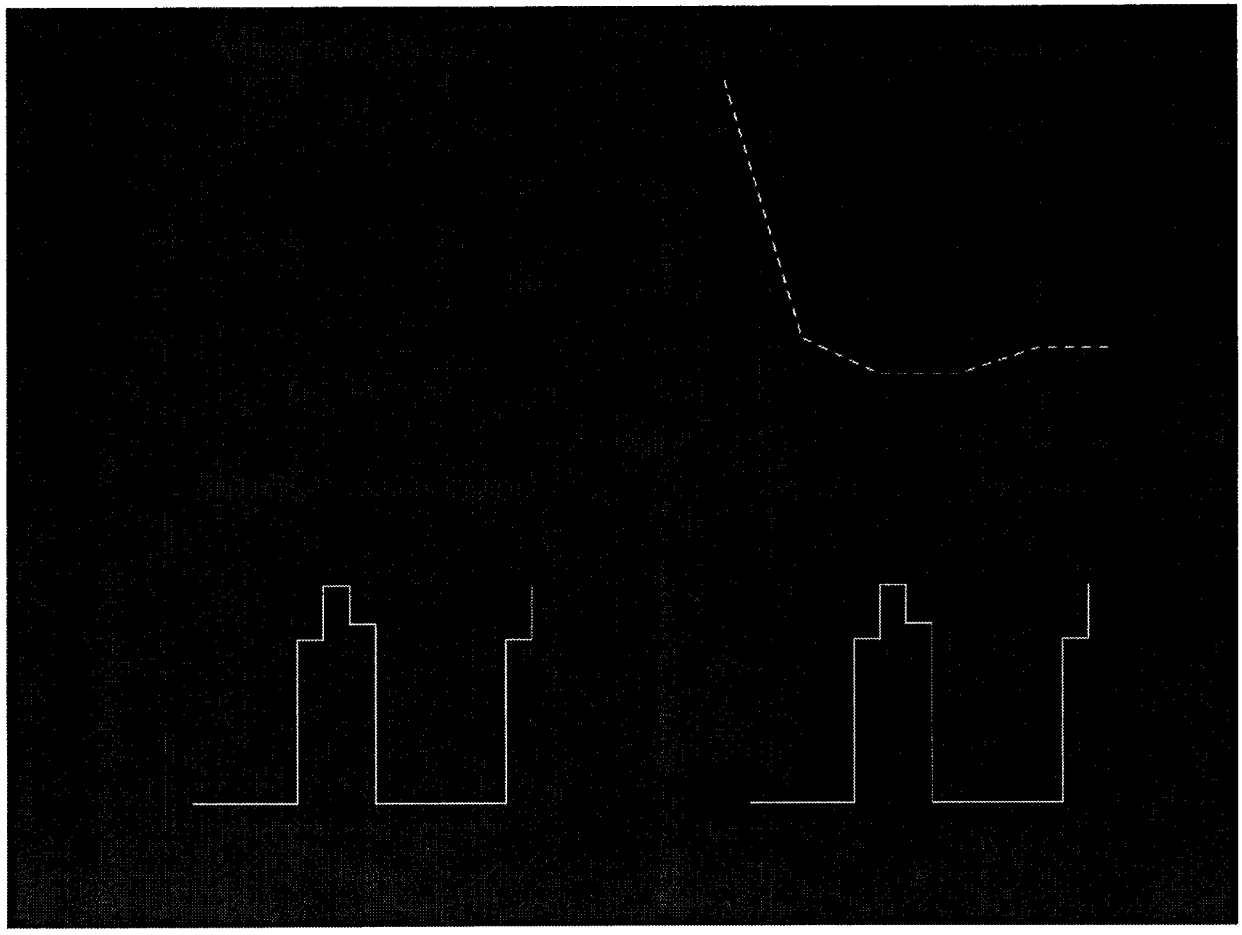


Figure 20 SNM's output referent to set—2 of message codes

Set-3 of message codes shows the SNM's sensitivity to the topology of node reporting malfunction. Here the QOS perturbation depends on the importance of the node reporting malfunction in the context of network topology. The criteria adopted was the amount of links in and out of the network element. Other criteria could as well be defined. In this table the convention for QOS variation is: S: Small variation, M: Medium, and H: High.

Figure 21 Set—3 of message codes

Message Code	Load Window	Switch	QOS
Initial Condition	3->6	A	S
Maintenance alarm	3->6	A	S
Maintenance alarm	3->6	B	M
Maintenance alarm	3->6	C	M
Maintenance alarm	3->6	D	M
Maintenance alarm	3->6	E	M
Maintenance alarm	3->6	F	M
Maintenance alarm	3->6	G	M
Maintenance alarm	3->6	H	M
Maintenance alarm	3->6	I	M
Maintenance alarm	3->6	J	M
Maintenance alarm	3->6	K	H
Maintenance alarm	3->6	L	M
Maintenance alarm	3->6	M	M

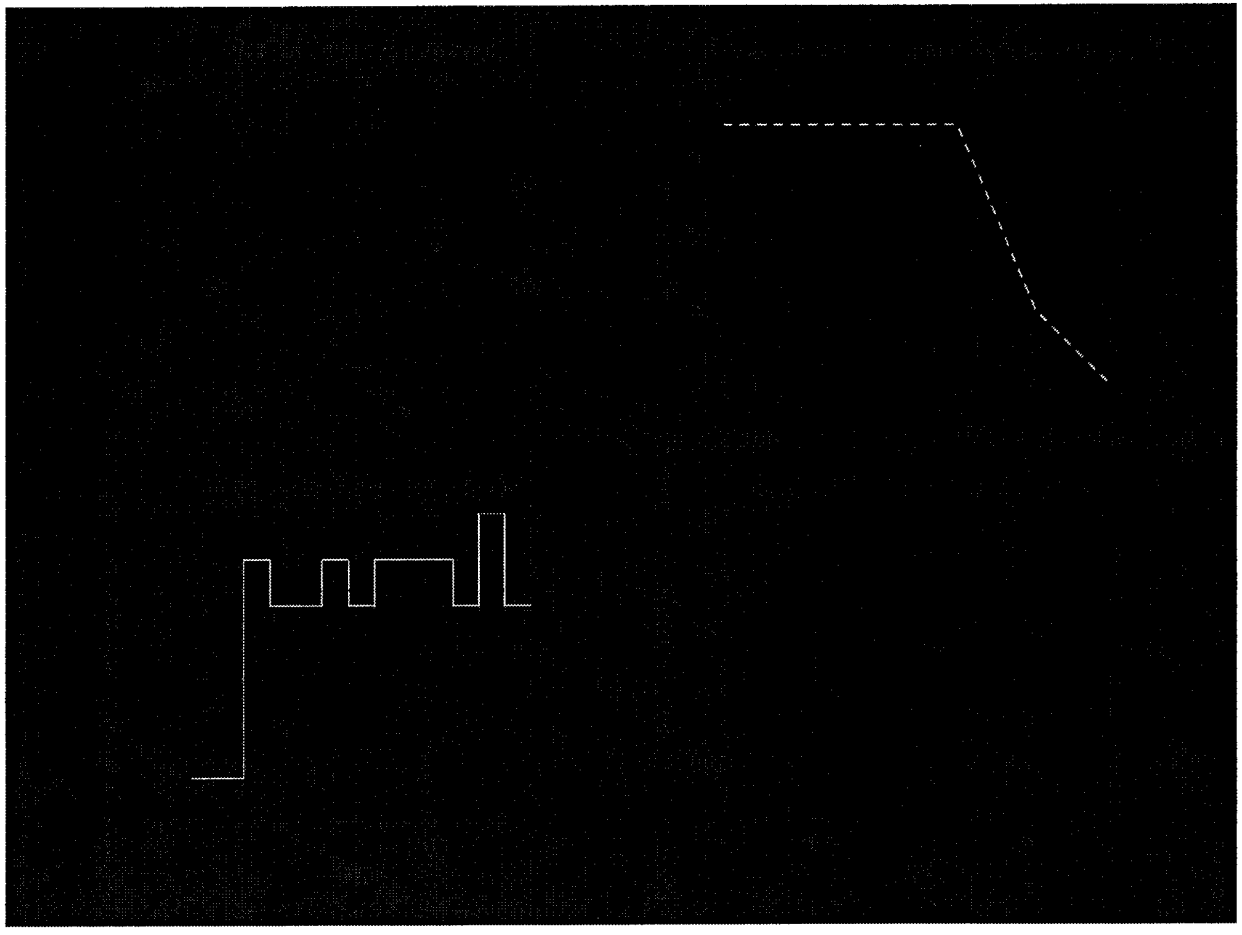


Figure 22 SNM's output for set-3 of message codes

Set-4 of message codes shows the SNM' sensitivity to incoming message codes. Traffic load and network element reporting malfunction are kept constant.

Table 9 Set—4 of message codes

Message Code	**Load Window**	**Switch**	Variation QOS
Initial Condition	6->9	A	No
Maintenance alarm	6->9	A	Down
Software error	6->9	A	Down
Protocol error	6->9	A	Down
Buffer error	6->9	A	Down
BER Medium	6->9	A	Down
BER High	6->9	A	Down
BER Very High	6->9	A	Down
High CRC count	6->9	A	Down
Retransmissions	6->9	A	Down
Line status DOWN	6->9	A	Down
Line status UP	6->9	A	Up
Entering buffer congestion	6->9	A	Down
Leaving buffer congestion	6->9	A	Up

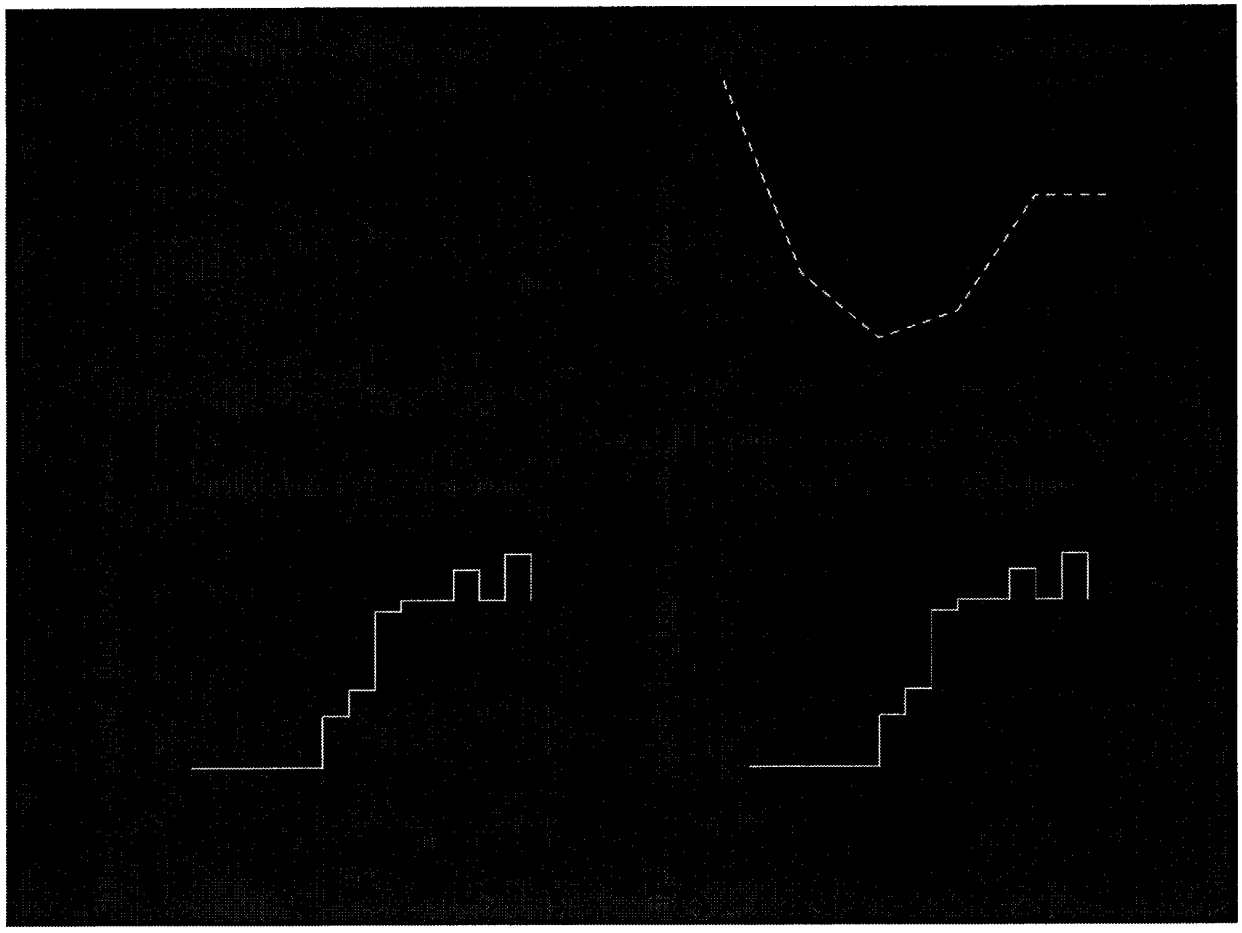


Figure 23 SNM's output corresponding to set—4 of message codes

Set—5 of message codes shows network element "A" reporting malfunction. Traffic load is kept constant.

Table 10 Set—5 of message codes

Message Code	Time	Switch	QOS
Initial Condition	0->3	A	-
Maintenance alarm	0->3	A	D
Software error	0->3	A	D
Protocol error	0->3	A	D
Buffer error	0->3	A	D
BER Medium	0->3	A	D
BER High	0->3	A	D
BER Very High	0->3	A	D
High CRC count	0->3	A	D
Retransmissions	0->3	A	D
Line status DOWN	0->3	A	D
Line status UP	0->3	A	I
Entering buffer congestion	0->3	A	D
Leaving buffer congestion	0->3	A	I

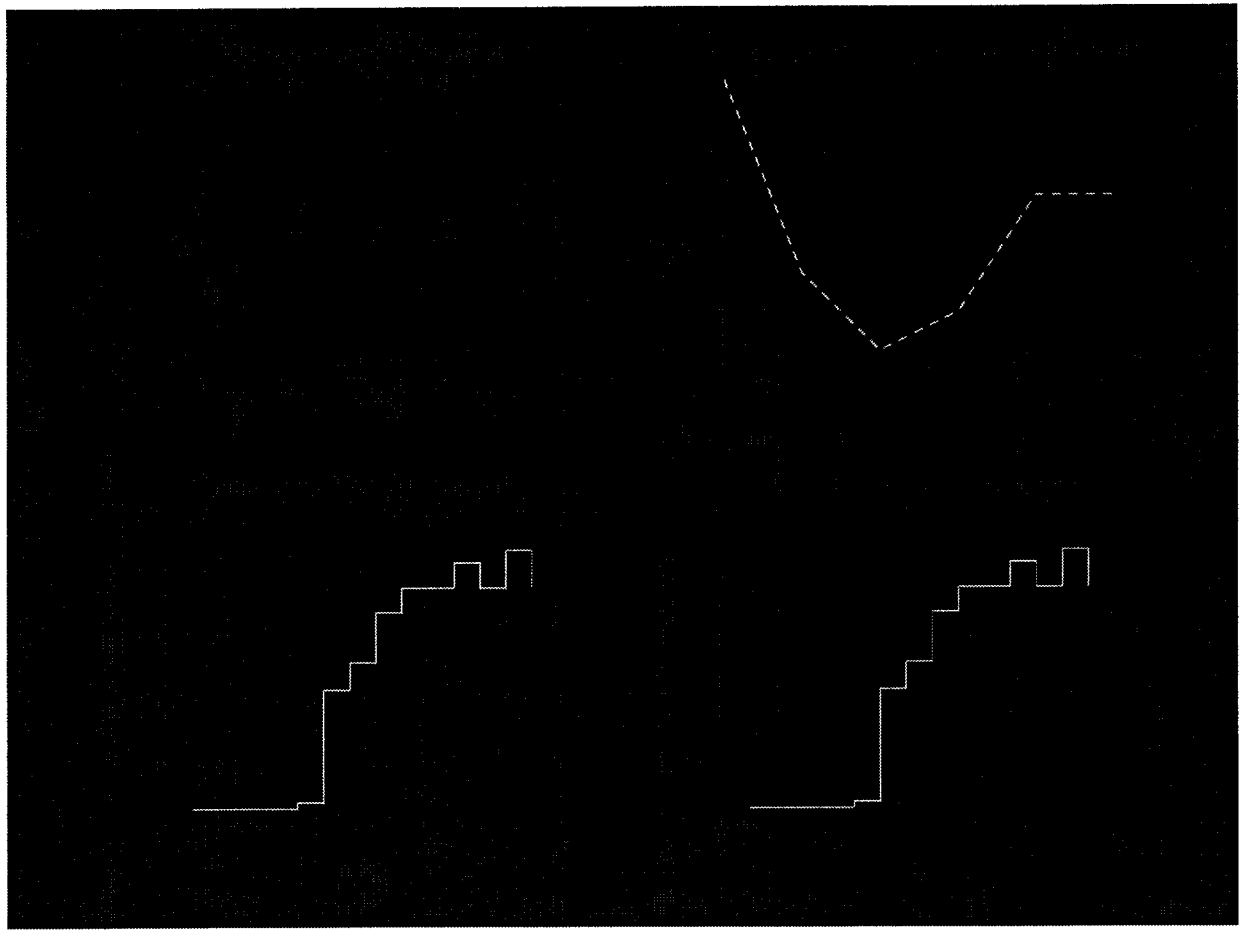


Figure 24 SNM's output for set—5 of message codes

Set-6 of message codes shows SNM' sensitivity to time of day.

Table 11 Set—6 of message code

Message Code	Load Window	Switch	QOS
Initial Condition	0->3	A	-
Software Error	0->3	A	D
Software Error	3->6	A	D
Software Error	6->9	A	D
Software Error	9->12	A	D
Software Error	12->15	A	D
Software Error	15>18	A	I
Software Error	18->21	A	I
Software Error	21->24	A	I
Software Error	0->3	A	D
Software Error	3->6	A	D
Software Error	6->9	A	D
Software Error	9->12	A	D
Software Error	12->15	A	D

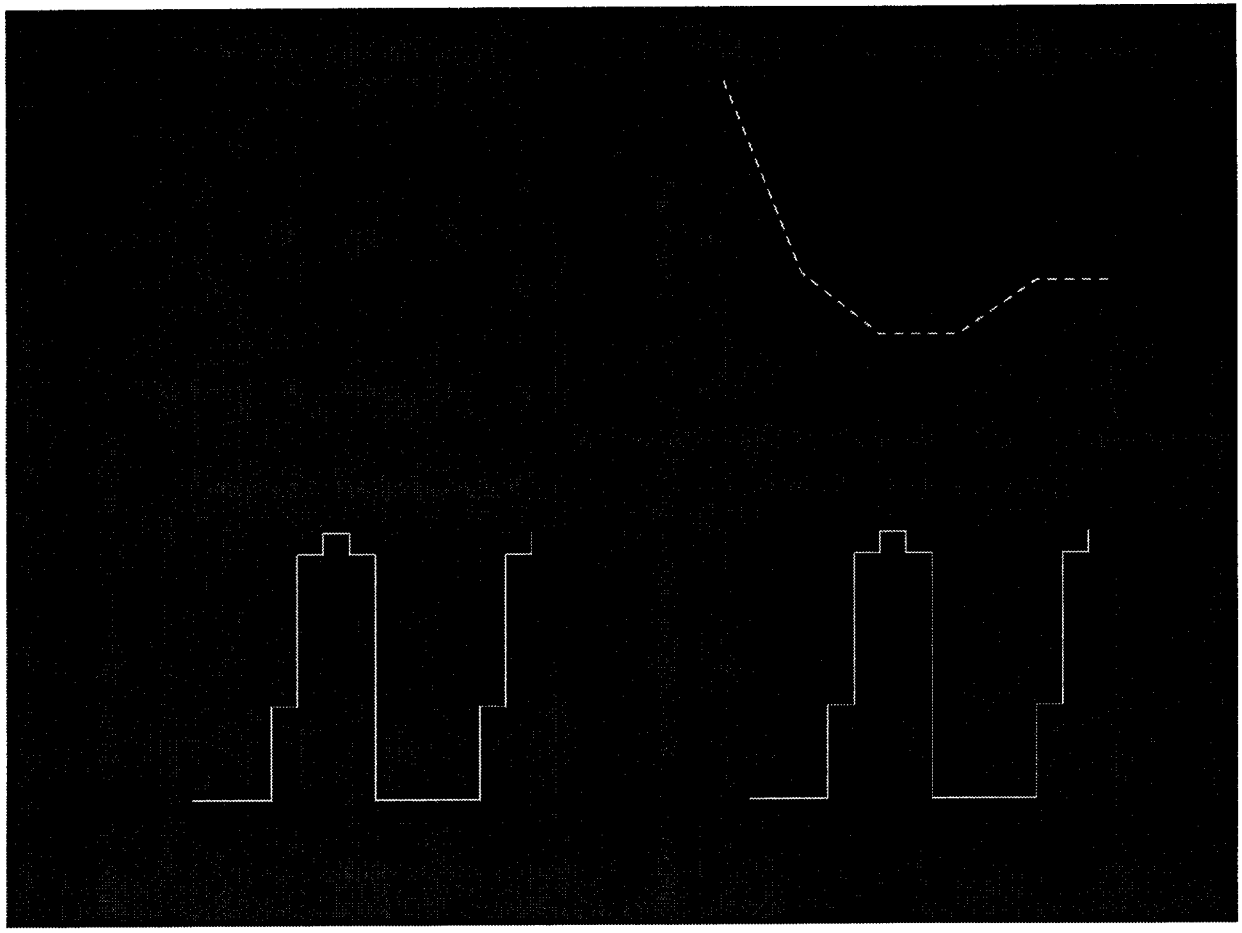


Figure 25 SNM's output for set—6 of message codes

Set—7 is the result of near 200 message codes generated randomly. Traffic load distribution during an entire day indicates peak and off-peak hours. The sensitivity of the network to malfunctions is higher during peak hours.

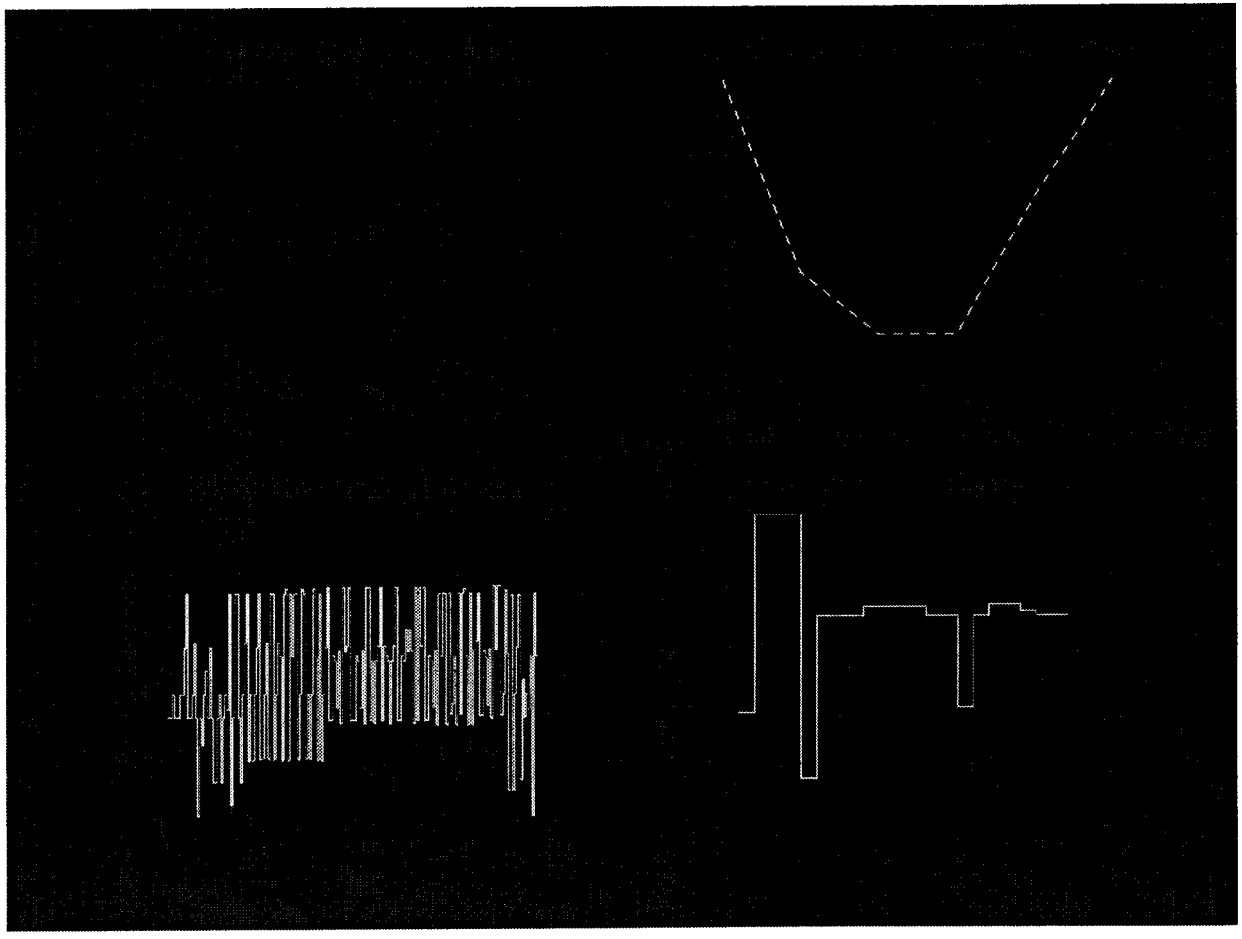


Figure 26 NHM's output for set—7 of message codes

Set—8 compares two switches reporting the same message code during the same time window. The switches reporting malfunction are switch A and switch H. According to our experimental topology, network element A has less importance than network element H. SNM takes this information in consideration when warning operators and scheduling network management pools.

Table 12 Set-8 of message codes

Message Code	Load Window	Switch	QOS
Initial Condition	0->3	A	-
Software Error	0->3	A	D
Software Error	3->6	A	D
Software Error	6->9	A	D
Software Error	9->12	A	D
Software Error	12->15	A	D
Software Error	15->18	A	I
Initial Condition	0->3	H	-
Software Error	0->3	H	D
Software Error	3->6	H	D
Software Error	6->9	H	D
Software Error	9->12	H	D
Software Error	12->15	H	D
Software Error	15->18	H	I

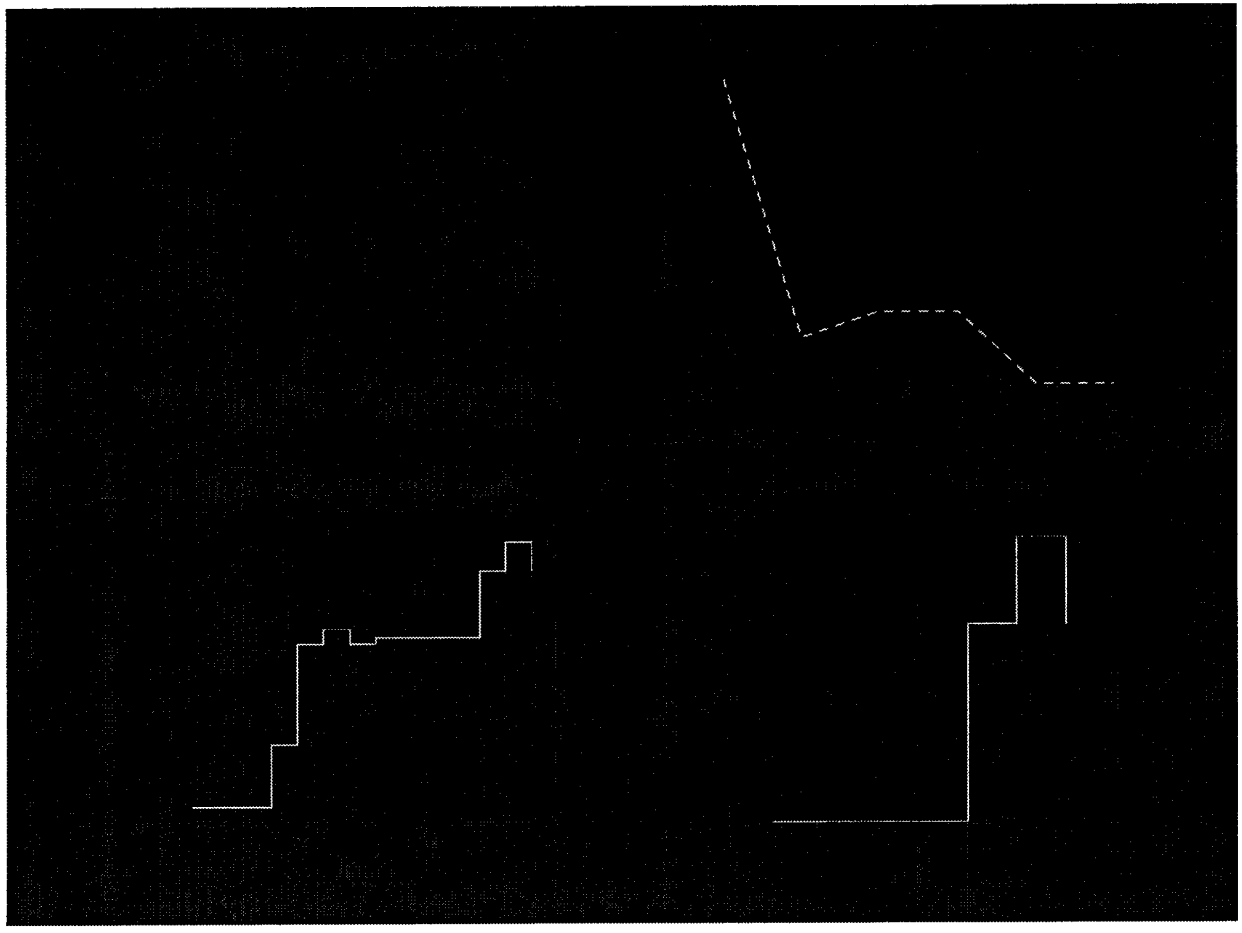


Figure 27 NHM's output referent to set-8 of message codes

Chapter 7. Conclusion

Precise definition of network events, states and transitions between states is very difficult. It becomes even more troublesome at Network Control Centers that manage several custom networks. The difficulty exists due to the fact that network states, events and transitions are directly linked to the QOS expected from such networks.

Considering all these potential uncertainties, the Simple Network Monitor as described in this report finds good application at custom or private networks. The Simple Network Monitor can be customized to serve a broad range of networks. It works independently of the network management protocol and makes the surveillance of enterprise networks an easier and more accurate task.

SNM also allows for effective control of overhead management traffic by emphasizing pools on trouble devices and deenphasizing from devices under normal operation. Network resources are saved and performance of the supporting Network Management System increases.

The SNM employs a combination of straight forwarded methods of multiple attribute decision making. It becomes a flexible and attractive tool to help operators to prioritize and take uniform actions when performing network surveillance. Operators will not have to think about what is and what is not to be considered critical on different networks. The Simple Network Monitor will help them in this task.

The SNM attenuates the following problems:

1. Management of integrated networks (multiple technologies under different set of constraints).

2. Management of hybrid networks (applications running over different medias may be better monitored if adequate parameters are established for each section of the hybrid network).
3. Management of custom networks (there will be no problem in prioritizing responses based on customer requirements).
4. Reduces overhead traffic by controlling pooling intervals over the network elements.
5. Prediction of unwanted trends in network performance.

A final and very important application to the QOS measure is in broadband networks such as ATM. The simplicity of the technique permits small modifications to support high speed networks. The algorithm can easily be put in firmware to comply with speed requirements when managing broadband networks. Speed in processing QOS info is a valuable performance asset that could be incorporated into any supporting Network Management System.

7.1. Further Research

1. Identify network malfunctions and critical trends, learn such patterns and procedures to bring the network back to normal operation (operation as planned). Interface to a configuration management module to bring the network to the operational state independent of operator's assistance.
2. Integrate the fuzzy approach of this model into a more formal statistical treatment to make it possible to apply other analytical measures.
3. Integrated into a comprehensive Network Management System. Results found here can be attached to inference networks to help speed up the troubleshooting process.

8. References

[1] Kirschen D. and Wollenberg B., Intelligent Alarm Processing in Power Systems, IEEE Transactions Power Systems, Vol. 80. N 5, May 1992.

[2] Bennett L. and Chou W., An Expert System for Diagnosing Performance Problems in SNA Networks, Network Management, Plenum Press, New York, 1990.

[3] Shier Douglas R. , Network Reliability and Algebraic Structures, Oxford Science Publications, Clarendon Press, Oxford, 1991.

[4] Tanenbaum Andrew S. , Computer Networks, Second Edition, Prentice Hall, Englewood Cliffs, New Jersey, 1989.

[5] Jain B. N. and Agrawala A. K., Open Systems Interconnection: Its Architecture and Protocols, Elsevier, New York, 1990.

[6] Dai S. and Wang M., Reliability Analysis in Engineering Applications, Van Nortrand Reinhold, New York, 1992.

[7] James C. Bezdek, Sankar K. Pal, Fuzzy Models For Pattern Recognition — Methods That Search for Structures in Data, IEEE Press, 1992.

[8] Chen S. J. and Hwang C. L., Fuzzy Multiple Attribute Decision Making — Methods and Applications, Springer-Verlag, 1992.

[9] Gupta M.M. and Yamakawa T., Fuzzy Computing — Theory, Hardware, and Applications, North-Holland, 1991.

[10] Terplan K., Communication Networks Management, Second Edition, Prentice-Hall, Inc., 1992.

[11] Rubino G., Sensitivity Computation in Network Reliability Analysis, Institut National de Recherche en Informatique et en Automatique — France, November 1991.

[12] Maxion R.A., Toward Diagnosis as an Emergent Behavior in a Network Ecosystem, Physica D 42, 1990.

[13] R.S. Cohen, H.K. Kan, R.J. Pennotti. Unified Network Management from AT&T, AT&T Technical Journal, November/December, 1988.

[14] Sprint's class notes on Network Planning, 1990.

[15] Sprint's Message Code Dictionary, 1992.

[16] Hughes's Personal Earth Station Network, Event Messages Users Guide, Release 7.6A, August, 1992.

[17] Zadeh L. A., Knowledge Representation in Fuzzy Logic, An Introduction to Fuzzy Logic Applications in Intelligent Systems, Kluwer Academic Publishers, 1992.

[18] Yager R. R., Expert Systems Using fuzzy Logic, An Introduction to Fuzzy Logic Applications in Intelligent Systems, Kluwer Academic Publishers, 1992.

[19] Germond A. J. and Niebur D., Power System Security Assessment Using The Kohonen Neural Network Classifier, Department of Electrical Engineering, Swiss Federal Institute of Technology, Lausanne, Switzerland, 1991.

[20] Kauffels F.J., Network Management — Problems, Standards and Strategies, Addison-Wesley, 1992.

[21] Nojo S. and Watanabe H., Incorporating Reliability Specifications in the Design of Telecommunication Networks, IEEE Communications Magazine, June 1993.

[22] D. Bertsekas and R. Gallager, Data Networks, Prentice Hall, Englewood Cliffs, New Jersey, Second Edition, 1992.

[23] S. L. Tanimoto, The Elements of Artificial Intelligence, Computer Science Press, New York, 1990.

[24] W. Karwowski and A. Mital, Applications of Fuzzy Set Theory In Human Factors, Elsevier, Oxford, 1986.

[25] A. Kandel, Fuzzy Mathematical Techniques With Applications, Addison-Wesley, 1986.

[26] M. Daneshmand and C. Savolaine, Measuring Outages in Telecommunications Switched Networks, IEEE Communications Maganize, June, 1993.

[27] HP OpenView Reference Manual.

[28] COMMAND/Post Reference Manual.

[29] NetExpert Reference Manual.

Appendix A: Quality of Service in the OSI Model

A.1. Introduction

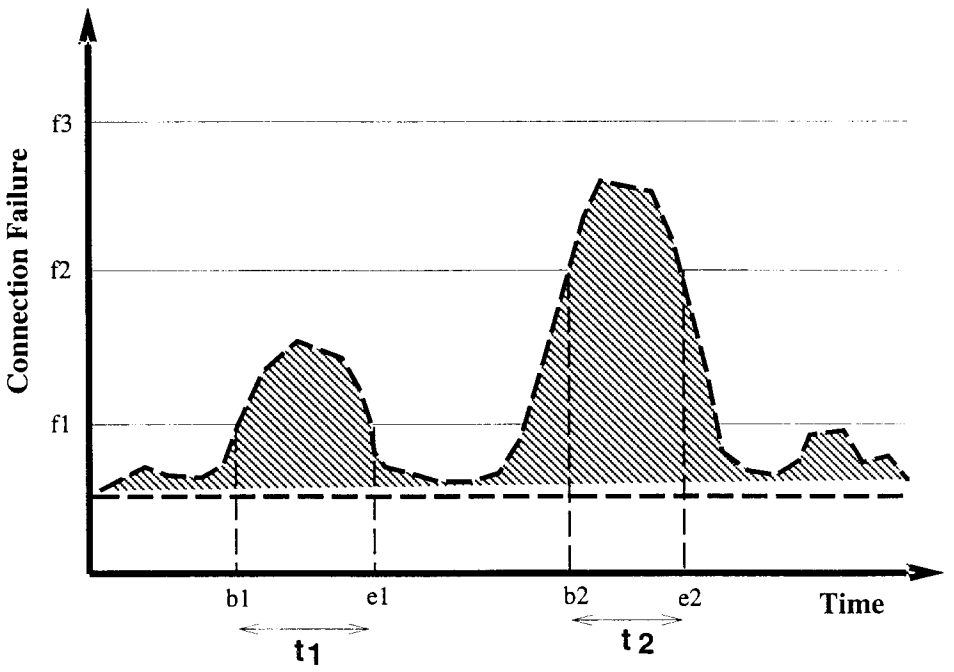
Quality of service (QOS) can be defined in terms of customer perception of the health of the enterprise. Definitions of network enterprise degradation follow at least two major approaches: those based on origin — destination pairs and those treating the enterprise as whole. The definition presented here is based on outage definition as found in [26]:

1. Origin-Destination Instability: A network instability, from an origin i to a destination j, has occurred when f, the probability of communication failure from the source i to the target j, exceeds the prespecified failure threshold valuef_1 for a period of time at least t_1. Let $O_{ij}(f_1, t_1)$ denote such a malfunction, where ij means from origin i to destination j. The malfunction definition is illustrated in figure 28. Here we show two arbitrary perturbations for an origin — destination pair ij. The two perturbations are characterized by two arbitrary sets of threshold values (f_1, t_1), and (f_2, t_2). The values of (f, t) need to be specified. They depend on many factors, including the type of service, location, and application. For instance, for the perturbation described by $(f1, t1)$, figure 28 shows that the network has been performing its required function up to the time b_1, it failed to continue performing the required function since time b_1, and has been in failure mode until time e_1. During the period $D = e_1 - b_1$, the network has been restored and started performing its required function again at time e_1. A similar description can be given for the perturbation described by (f_2, t_2). At this point, it is important to note that during the (f_1, t_1) perturbation, shown in figure 5, the subnetwork connection i to j, ij is still partially immune to such fluctuations because the performance fluctuations are bellow the allowed maximum threshold. However, it will be classified as "failed" with respect to the pre-specified failure

threshold indicators of f_1 and t_1. The term origin—destination subnetwork can, for instance, describe the situation where two offices are establishing a communication transaction. We will say that the origin-destination pair ij is in outage state (f_1, t_1)

2. Network Outage: A network is in outage state characterized by (f_1, t_1) if, and only if, at least one origin—destination subnetwork of the network is in outage state$(f1, t1)$, for some f_1, t_1

Figure 28 Two Examples Of Perturbations From Origin i To Destination j.



A.2. OSI Structure of Nodes

In this section we discuss the basic structure of the OSI architecture. Networks are highly complex entities and are organized in a structured way in order to reduce their design and operational complexity. In this work we assume a network organization based on a proposal developed by the International Standards Organization (ISO). The proposed model is called the ISO/OSI Open Systems Interconnection Reference Model because it deals with connecting open systems — that is, systems that are open for

communication with other systems [4]. In this OSI model a network node is organized as a series of layers, each one built upon its predecessor. Faults happening at lower layers propagate upstream and can eventually present distinct effects on the service being provided, according to the layer's performance parameters.

The OSI model has seven layers. The general principles that were applied to arrive at the seven layers are as follows [4]:

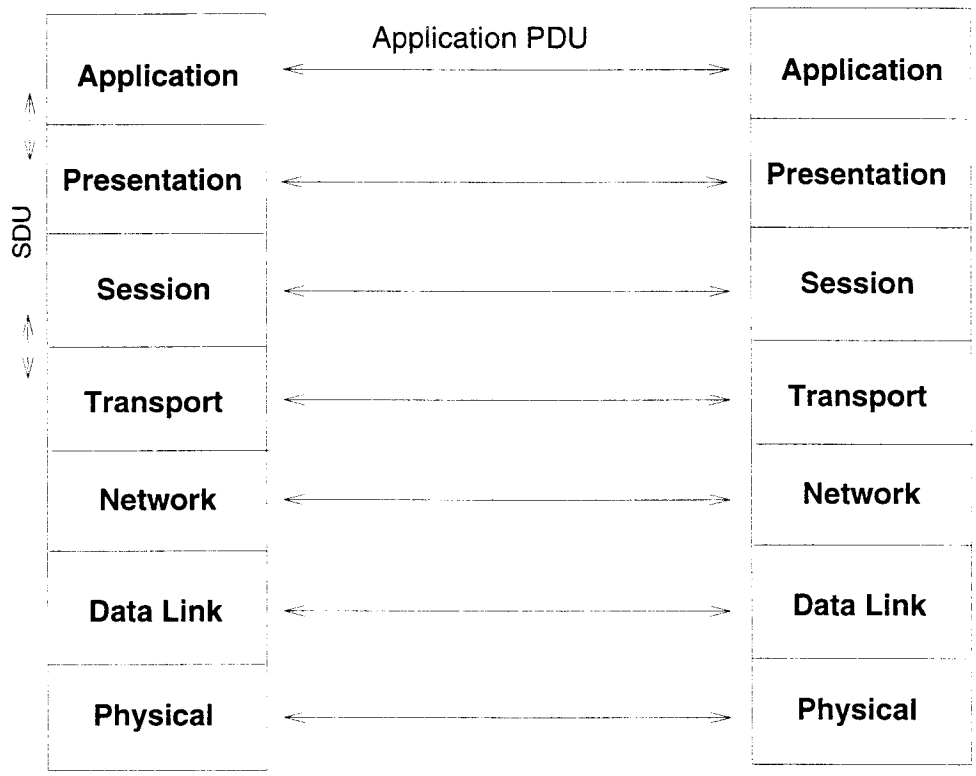
1. A layer should be created where a different level of abstraction is needed.
2. Each layer should perform a well defined function.
3. The function of each layer should be chosen with an eye toward defining internationally standardized protocols.
4. The layer boundaries should be chosen to minimize the information flow across the interfaces.
5. The number of layers should be large enough that distinct functions need not be thrown together in the same layer out of necessity, and small enough that the architecture does not become unwieldy.

The OSI model consists of the following layers, from bottom-up: physical, data link, network, transport, session, presentation and application. Tables describing each layer's function are further described in summary form in this thesis. The application layer is the highest and the closest to the user's perspective. Figure 29 shows the seven layers and their logical communications over the network. We note the logical communication through PDUs (Protocol Data Units) and inter-layer message interchange by SDUs (Service Data Units). By looking at these messages one could correlate basic cause-effect relationship when troubleshooting a network problem.

A.2.1 OSI Layers And Functions

The protocol that defines inter and intra-layer interaction also defines the functions to be performed by each layer. These functions can be classified as: expedited data-transfer, sequencing, acknowledgment, error detection, reporting and recovery, message multiplexing and de-multiplexing, splitting and recombining, segmentation and reassembly, blocking and de-blocking, concatenation and separation.

Figure 29 OSI Layers



Protocols are classified under specific classes where each class defines a subset of functions contained in the protocols of the specific class layers. Table 13 describes the purpose and functions of the OSI Application and Presentation layers. Tables 14 and 15 describe the purpose and functions of the OSI Session and Transport layers, respectively.

Table 13 Function of Application and Presentation Layers

Application
Purpose: Serve as a window between correspondent application processes so that they may exchange information in the open environment. Functions: . determination of resource adequacy to provide an acceptable quality of service . identification of communicating Application entities . determination of their access rights and user authentication . synchronization of cooperating applications . negotiation of the "abstract syntax" of Application protocol and user data . the user of lower layer services . error detection and notification
Presentation
Purpose: Provide appropriate representation of all information communicated between Application entities. It is primarily concerned with data syntax and its logical structure. Functions: . connection establishment and termination . negotiation and possibly re-negotiation of the abstract syntax of Application protocol-data-units . syntax transformation including data compression, if required . encryption, . data transfer

A.3. Quality of Service

Quality of service (QOS) is application sensitive: different applications require different levels of quality of service. For example, two service users that need to communicate using the services provided by a third entity will require the quality of service that is needed to maintain the connection. On the other hand, the service provider will indicate the QOS that it can support over that required connection. The connection is established if both parts agree on values assigned to parameters that define the connection. The connection may be broken if, at some point, the quality of service deteriorates to levels below what is acceptable by the service users. The initiative to discontinue a connection belongs to the users. The service provider, in general, does not interrupt a connection due to fluctuations or degradation in the ongoing quality of service.

Table 14 Function of Session Layer

Session
Purpose: Provide Presentation layer entities with the means to organize exchange of data over a connection either in the full-duplex or half-duplex mode of communications. Functions: . connection establishment and its maintenance . orderly connection release, which may optionally be negotiated . normal data transfer, which may be half-duplex or full-duplex . typed data transfer, which is not subject to restrictions imposed by the half-duplex mode of communication . expedited data transfer, which is not subject to flow control restrictions . establishment of synchronization points and resynchronization . activity management . address translation . reporting of exceptional conditions

Table 15 Function of Transport Layer

Transport
Purpose: Optimize the use of Network services and ensure that the quality of Transport services is at least as good as that requested by the Session entities. Due to the fact that the characteristics and performance of the Network service may vary substantially, a variety of Transport protocols are available to ensure that the service that it provides is largely independent of the underlying communication network. Functions: . selection of network service as a function of parameters such as throughput, transit delay, set-up delay, and error characteristics . connection establishment and its maintenance . establishment of appropriate data unit size . normal and expedited data transfer . error detection and reporting for lost, damaged, duplicated, misordered, or misdelivered data units . error recovery . end-to-end sequence control of protocol-data-units . multiplexing or splitting of transport connections onto Network connections . end-to-end flow control in a Session oriented data transfer

The following description illustrates the dynamics of protocol selection to perform layer functions and also provides an overview of inter-layer negotiation; in a connection oriented data transfer, during the connection establishment the required functions are negotiated and the protocol class to support such connection defined. The negotiation process happens between the service users and their corresponding service providers. This cross-layer negotiation takes place on connection oriented calls by defining the quality of service required versus the quality of service provided. For instance, depending on the reliability of the network layer, the transport layer may choose between four levels of

protocols. If the network layer is reliable enough, the protocol selected by the transport layer performs only trivial data verification. On the other hand, if the network layer does not offer assured reliability, the network layer will have to employ its highest level of protocol in order to guarantee full data integrity. The additional overhead required by more sophisticated protocols tend to increase the transmission delay and diminish data throughput. This is a continuously evolving trade-off that ultimately defines the entire network performance. Tables 16 and 17 describe the purpose and functions of the OSI network, data link and physical layers.

Table 16 Function of Network Layer

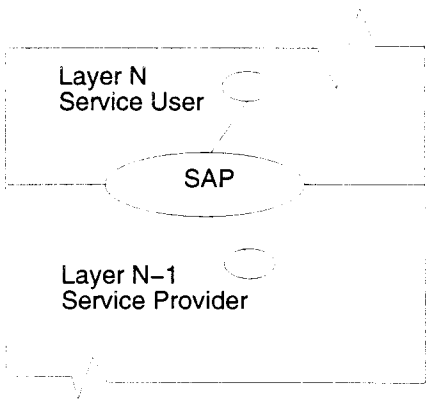
Network
Purpose: Provide the means to establish, maintain and terminate Network connections between open systems. It specifies the functional and procedural means to transfer user data between Transport entities over a Network connection. Functions: The required functions are specific to the communication subnetwork and must be implemented by each open system in the subnetwork, including intermediate systems. Intermediate systems are capable of routing and relaying information between hybrid communication networks. . connection establishment and its maintenance . multiplexing and possibly splitting network connections onto data-link connections provided by the next lower level . re-initialization, or reset, of connection . addressing, routing and relaying . normal and expedited data transfer . sequencing . error detection, notification and possibly recovery to support desired quality of service . flow control . termination services when resquested by a using party . congestion control . billing information

Table 17 Functions of Data Link and Physical Layers

Data Link
Purpose: Provide functional and procedural means to establish, maintain and release connections between Network entities and to transfer user data. It is also responsible for detection and possible correction of errors occurring over the Physical connection. Functions: . connection establishment and release . delimiting, mapping and synchronization of protocol-data-units . error detection and recovery (CRC) . multiplexing of one data-link connection onto several physical connections . flow control and sequenced delivery
Physical
Purpose: Provide mechanical, electrical, functional and procedural means to establish, maintain and release physical connections and for bit transmission over a physical medium. Functions: . connections establishment and in-sequence transmission of bits over a data circuit . bit error control.

According to Bijendra N. Jain and Ashok K. Agrawala [5] (Open Systems Interconnection), the QOS is described as the relationship between two layers in which one is the service provider and the other is the service user. This inter-layer relationship is established whenever the user starts a new application. The success or not of establishing a new application over the network is the result of the QOS negotiation across-layers. Figure 30 illustrates the inter-layer relationship. The service access point, SAP provides the means for transferring SDUs across layers.

Figure 30 Inter-layer Relationship.



The overall classification of QOS parameters can be further divided as parameters for connection-oriented and connection-less oriented type of data exchange. In the context of connection-oriented services [5], the QOS of two connections established between two different pairs of (N)-SAP (Service Access Point for layer N) may be different. Furthermore, the QOS of two connections over the same (N)-SAP may also be different. The quality of service as defined by OSI standards is applied to each of the seven layers at a node and each layer carries its own parameters. A general classification of QOS can be given in two major groups: performance related and additional features offered [5], as shown in figure 8.

Figure 31 Overall QOS Parameters.

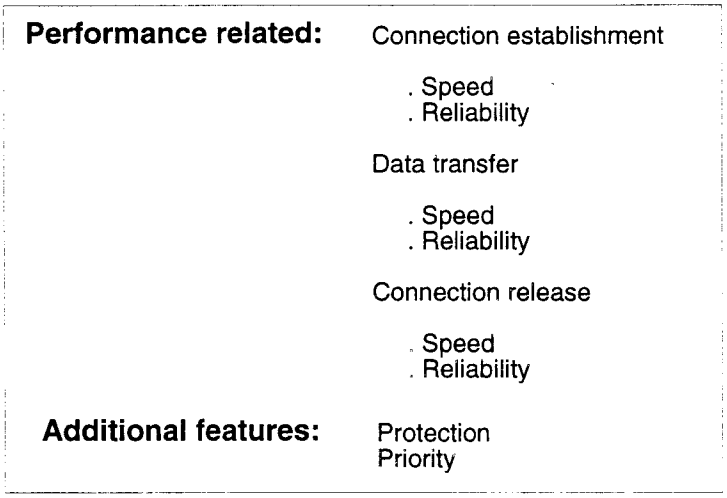


Table 18 shows a break down of the performance related QOS parameters by phase to establish a session.

Table 18 Performance-related QOS Parameters.

Performance-related QOS parameters		
Phase	Performance Criterion	
	Speed	Accuracy/Reliability
Connection Establishment	Establishment Delay	Establishment Failure Probability
Data Transfer	Transfer Delay Throughput	Residual Error Rate Transfer Failure Probability Resilience
Connection Release	Release Delay	Release Failure Probability

The definition of each term is given in table 19::

Table 19 QOS Parameters Definition.

QOS Parameters Definition
Connection Establishment: call setup.
Data Transfer: effective data transfer.
Connection Release: call disconnect.
Establishment Delay: It is the time spent from the moment a connection request was issued until a connect confirmation was received back.

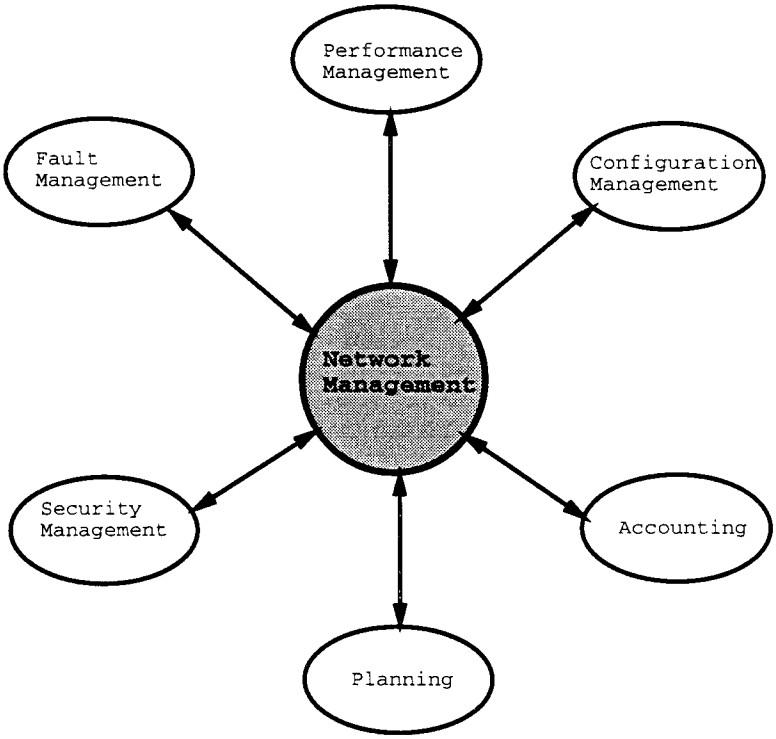
Table 19 (Continued) QOS Parameters Definition.

Establishment Failure Probability: For each connection request there is a maximum acceptable delay. Assuming that the service-user does not incorporate any chances of failure in the process of connection establishment, the establishment failure probability measures the probability that a connection will not be established in a time smaller than the maximum acceptable delay. It is a measure of the capacity of the service-provider to establish the links as requested.
Transfer Delay: Measure the time spent between a data request and the corresponding data indication.
Throughput: Is the data rate that may be successfully transferred over a connection on a sustained basis.
Residual Error Rate: Is the estimated probability that information is lost, transferred with error or a duplicate copy is transferred.
Transfer failure probability: Is the estimated probability that the observed performance will be worse than the specified level.
Resilience: Is the estimated probability that a service-provider will, on its own, release the connection, or reset it, within a specified interval of time [5].
Release Delay: It is the time spent between the issuing of a request for disconnection and receiving the corresponding disconnection indication.
Release Failure Probability: As during connection establishment, during the release there exists a maximum acceptable time to release certain connection. The release failure probability is the estimated probability that the service-provider is not capable of releasing a connection inside that maximum acceptable time.

Appendix B: Network Management Functions

A Network Management System is a collection of software modules and hardware aimed to implement command and control operations to prevent traffic congestion, maintain the expected quality of service and maximize network resources availability. A short description of each of the Network Management functions is given below, figure 32.

Figure 32 Main Network Management Functions



Configuration management: Responsible for keeping track of all changes that happen in network topology and devices. These changes include managing active network configuration from the logical and physical stand points.

Fault management: Includes network status supervision, end-to-end and segment testing, diagnosis of problems and repair, backup and reconfiguration, and trouble

tracking. These functions allow a network manager to identify a condition affecting service, isolate the problem through tests and other diagnostics, reconfigure around the problem while it is fixed, and track the trouble, on-line, with a historical status recording capability [13].

Performance management: Involves performance definition and monitoring as well as identifying trends and thresholds [13].

Accounting management: Supports budgeting, bill-back, and bill verification activities for network managers [13]. The data from accounting management also helps network engineers to reevaluate and improve network design. This data assessment is usually processed overnight or on weekly basis due to the enormous amount of data generated by network activity.

Security management: Includes the activities needed to establish and maintain network access security and partition access by authorization level, function, geography, time of day, or other criteria [13]. It plays a very important role when two or more customers lease their private networks over common hardware resources such as links, switches, etc.

Network planning: Covers the areas of capacity, contingency, and strategic planning. This function allows managers to improve system capacity and cost while planning for future events and growth [13].

Appendix C: Network Planning

Customers share network resources such as transmission facilities, buffer spaces, and switch memories. Key attributes of most of these services are data rate, performance requirements, and reliability. In order for the network to operate without problems, implementation of an efficient access capacity allocation is needed to provide to customers generating traffic with differing characteristics, e.g, data rate, packet length, performance objectives, routing requirements, and flow control/protocol procedures[2]. A bandwidth management becomes imperative in order to provide for a optimum use of network resources.

The network capacity planning mission is to design, analyze and implement services required by customers. The planning activity is driven by the quality of service being purchased by a customer and the necessity to minimize costs and maximize calls.

Network engineers and technical support groups interact with customers to provide the best solution for their needs and the best usage of the network resources. This interaction basically helps the network team to create an adequate network topology.

Every network owner performs a strategic network growth planning. This strategic planning normally covers long periods such as 3 to 5 years. This is based on marketing forecast for network activity growth. These marketing figures help network planners define the future location of hardware and software to attend the market demand. After the completion and approval of the deployment plan, the purchase orders are submitted and the new equipment is deployed.

Marketing forecasts become reality with the arrival of new customers. These new customers buy services based on their application requirements. The process of translating customer requirements into configuration tables is performed by the Translations Group.

This group generates configuration tables based on the quality of service that is being bought by customers. Tables contain performance parameters, constraints, thresholds, statistics and accounting information that has to be logged. Configuration tables are further sent to the Network Support Group which checks if the hardware and software required are compatible with what has been deployed.

After obtaining the Network Support's approval, tables are sent to network operators at Network Control Center. In this process operators have the responsibility to set up the new network by loading the tables in certain switches. The loading process is negotiated with the customer and, depending on customer's convenience, it can be static or dynamic.

In case of any incompatibility during the loading process, operators contact Network Support and Field Engineers to have them provide the necessary adjustments. As soon as adjustments are in place and tables are loaded, the network operation is started.

The configuration of a data network is the result of a sequence of development steps. These steps are designed to assure the final network meets both present and future communication needs in the most cost effective way. The network planning process follows the traditional systems engineering approach for problem solving.

The following steps which are based on a major data network describe the network development process:

1. Document Network Requirements: This step involves collecting data such as location of the host processors, identification of terminal types, number and location, as well as such traffic data characteristics such as average transaction length and transaction rates.
2. Configure Logical Matrix: This step involves organizing the end point requirements into a matrix for the purpose of analyzing traffic volumes by location.

3. Identify Backbone Network: This step establishes an initial network topology by selecting certain locations as nodes on the basis of traffic volumes and geographical location.
4. Develop Physical Matrix: Development of the physical matrix involves organizing the traffic into a matrix using the Backbone locations identified in the previous step.
5. Determine Trunk line Requirements: This step uses the physical matrix data to determine the traffic volumes between Backbone Nodes.
6. Determine Line Access Requirements: Using the logical and physical matrices, this step determines the traffic volumes entering the Backbone Network from concentrators or hosts.
7. Select Node Hardware: Once the requirements for trunk lines and general access have been established and nodal traffic volumes have been assessed, the nodal hardware can be selected.
8. Select Concentrator Hardware: This step identified the concentrator hardware using the requirements for concentrator capacity, external access capacity and the port requirements for terminal access.
9. Configure for Redundancy: Perform a trade-off between availability and cost.
10. Plan for Growth: Consider the expected growth patterns and new requirements due to the introduction of new applications.
11. Provide for Network Control: Consider Network Management functions according to the service being provided.

The network states are defined based on network-capacity planning principles. In the planning process the network planner tries to achieve a state with previous defined network performance. The network engineer uses information based on traffic flow, routing algorithms, resource utilization, network performance data, networking requirements,

technological trade-offs, and estimated growth of present and future applications to perform network capacity planning. In many cases, however, optimal performance is not achieved. The following list describes some possible reasons for the deviations between projected and actual resource consumption in a network:

1. underestimation of the work load.
2. forecasting errors for the work load volumes.
3. planning errors.
4. inaccurate estimation of resource capacities.
5. underestimation of overhead.
6. inaccurate availability estimation for the resources.

These deviations are responsible for instability in quality of service and may require immediate reaction from operators at Network Control Center.

Once the network is up and running, the Network Management System at Network Control Center receives messages from concentrators and switches in a regular basis. The objective of this communication is to provide a feedback to Network Control Center on any event that may affect switch operation. A twenty four hour continuous network surveillance assures the customers of continuity in the quality of service that has been purchased.

Appendix D: Systems Approach to Network Management

In these times of intensifying competition and at the same time complexity of the telecommunications systems, network carriers, product vendors and users are searching for a way to take the best they can from the resources that are available.

New technologies and innovative management have shown to be the key answer for the challenges faced by the telecommunications industry. The fundamental application of systems engineering in this area is to understand the dynamics of the process as a whole, by identifying the user, service provider and product vendors needs and limitations. Through data collection and analyses the deficiencies or problems to implement a telecommunications system are made evident. These problems and limitations define the systems baseline by establishing a set of requirements, constraints, and design criteria. Based on the results, functional analyses and allocations are generated to apportion the appropriate system-level requirements down to the subsystem, unit, and lower levels of the system.

In order to come up with this baseline, during the preliminary system analysis the system engineer tends to act on the following questions:

1. Define the problem: It begins with the clarification of objectives, defining the issues of concern, and limiting the problem so that it can be studied in an efficient and timely manner [23].
2. Identify feasible alternatives: Identify alternative solutions to the problem. A set of possible candidates are considered and an evaluation criteria will select the alternative to be pursued.

3. Select the evaluation criteria: For instance, it may include expected system performance, cost effectiveness, logistics, operational availability, expandability, reusability, etc.
4. Apply modeling techniques: The model may be simple or complex. The extensiveness of the model will depend on the nature of the problem relative to the number of variables, input-parameter relationship, complexity of the operation, etc.
5. Generate input data: Specify the requirement for appropriate input data. Specific data requirements are identified from the evaluation criteria. This is a very critical step and in many cases it dictates the validity of a proposed solution. In the cases where data is not available, it can be generated using criteria that resemble the real systems operation.
6. Manipulate the model: It conveys the application of data to the model and analysis of its results. It will lead to the identification of the best alternative as defined by the evaluation criteria.

Trade-offs are performed during each step of the process in order to narrow the options to the best and most feasible alternative. Figure 39 presents a classical view of the systems analysis process.

The network planning requires the methodology presented for systems analysis. An efficient network planning may prevent several operational problems from happen and may allow easily further expansion. Many organizations are utilizing data networks to link powerful computers together and to provide access to these computers from remote locations. Such networks serve a variety of needs which include [14]:

1. Access to corporate and public data bases
2. Collection of information from branch locations

3. Access to powerful computer resources
4. Rapid distribution of administration messages
5. Complete transactions such as banking, commodities shipment and payment, securities trades and travel reservations
6. Improved customer service by processing customer requests and service problems promptly.

By assuming the classical systems approach to problem solving, the first step to implement a new network is to perform Requirements Analysis. In a network environment this phase covers the following steps:

1. Documenting the Requirements
2. Building the Logical Matrix
3. Identifying the Backbone Node Locations.

Each of the steps above can be further broken down into more specific tasks. A proposed break down of such steps is as follows:

1. Documenting the Requirements: Document the existing facilities.
    a. Number of terminals supported
    b. Speed/Code/Protocol of each terminal
    c. Physical location of each terminal
    d. Physical location of the hosts
    e. Speed/code protocol of the hosts.
2. Documenting the Requirements: Traffic statistics data.
    a. Average number of terminals in use at any one time,
    b. Average length of input inquires

c. Average length of output responses

d. Peak hours in different time zones.

3. Documenting the Requirements: Calculate average input/output.

   a. Average number of inquires input from the terminals versus time.

   b. Average number of responses output from the hosts versus time.

   c. Average percent of time terminal is in output state.

   d. Average percent of time terminal is in input state.

4. Logical Matrix Development: Once the basic traffic requirements are known, the next step is to organize the data. The first part of this organization is to construct a logical to/from matrix. This matrix indicates the possible source-to-destination connections without consideration of physical paths available.

5. Network Topology: Once the requirements have been documented and the logical matrix constructed, the next step is to determine the topology of the Backbone Network.

   a. Physical considerations: Includes existing or planned facilities and geographical preferences as well as considerations such as access to communications facilities.

   b. Cost considerations: Includes the physical facilities, labor and communications costs.

   c. Requirements of the network: Includes the pattern and volume of the network traffic as well as considerations of redundancy and network growth.

Other steps in the systems analysis follow the already established baseline.

Appendix E: Analogies Used

E.1. Analogy with Sensitivity Analysis

The approximation provided by this model is based on concepts borrowed from network reliability sensitivity analysis [11]. We could say that trends or fluctuations in quality of service could be observed by analyzing the network through its sensitivity to message codes, topology of faulty switch, and time of the day when an event is reported. It is certainly a gross approximation, but the results provide us with meaningful information that could help network operators to start appropriate actions to prevent the network from migrating into critical states.

The following arguments identify some of the principles for such approximation:

1. Consider that a specific network has sensitivity *"S" as a function of individual sensitivities* s_i of each processor/switch in this network.
2. Consider also β_i *as the sensitivity of "S" with respect to each independent variable* s_i. By definition,

$$\beta_i = \frac{\partial \mathrm{S}}{\partial \mathrm{s_i}}$$

3. Consider now each *elementary sensitivity as a function of some local measure* reported by a message code

$$s_i = s_i\left(mc_i\right)$$

These assumptions lead us to conclude that

$$\varphi_i = \frac{\partial \mathrm{S}}{\partial \mathrm{mc_i}}$$

is the instantaneous variation of the global sensitivity "S" with respect to the event generating the message code at switch i.

Assuming that the sensitivity to a switch depends on the message code generated, we have:

$$\varphi_i = \beta_i \frac{d\mathbf{s_i}}{d\mathbf{mc_i}}$$

The above result only takes into account the message code generated by a trouble switch. Individual sensitivity, s_i, could be described as a function of topology:

$$s_i = s_i(topology)$$

At network level, the fluctuations in sensitivity could be described as:

$$\frac{d\mathbf{S}}{d(topology)} = \sum_i \beta_i \frac{ds_i}{d(topology)}$$

If we now replace variable topology by variable time of day, local and network sensitivities to time of day could be identified as well.