

SIMPLE NETWORK MANAGEMENT PROTOCOL -SIMULATION-

A DISSERTATION

*Submitted in partial fulfilment of the
requirements for the award of the degree*

of

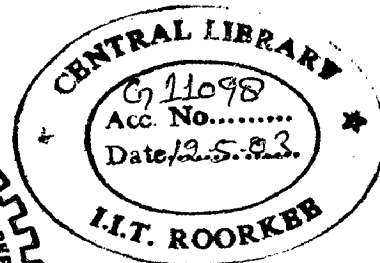
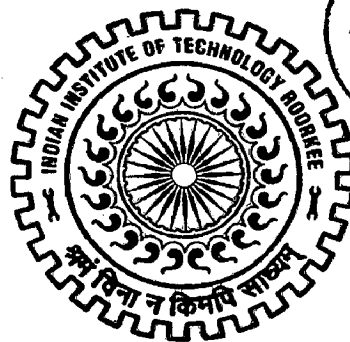
MASTER OF TECHNOLOGY

in

INFORMATION TECHNOLOGY

By

AYITHEPALLI DEEPAK BABU



**ER & DCI
NOIDA**

**IIT Roorkee-ER&DCI, Noida
C-56/1, "Anusandhan Bhawan"
Sector 62, Noida-201 307**

FEBRUARY, 2003

IP

CANDIDATE'S DECLARATION

I hereby declare that the work presented in this dissertation titled “**SIMPLE NETWORK MANAGEMENT PROTOCOL - SIMULATION**”, in partial fulfillment of the requirements for the award of the degree of **Master of Technology in Information Technology**, submitted in **IIT, Roorkee – ER&DCI Campus, Noida**, is an authentic record of my own work carried out during the period from August 2002 to February, 2003 under the guidance of **Dr. P.R. Gupta**, Reader, Electronics Research and Development Centre of India, Noida.

The matter embodied in this dissertation has not been submitted by me for award of any other degree or diploma.

Date: 25-2-2003

Place: Noida


A. Deepak Babu.
(Ayithepalli Deepak Babu)

CERTIFICATE

This is to certify that the above statement made by the candidate is correct to the best of my knowledge and belief.

Date: 25.2.2003

Place: Noida


(Dr. P. R. Gupta)
Reader,
ER&DCI, Noida

ACKNOWLEDGEMENT

I hereby take the privilege to express my deepest sense of gratitude to **Dr. Prem Vrat**, Director, Indian Institute of Technology, Roorkee, and **Mr. R.K. Verma**, Executive Director, ER&DCI, Noida for providing me with this valuable opportunity to carry out this work. I am also very grateful to **Dr. A.K. Awasthi**, Programme Coordinator and **Dr. R.P. Agrawal**, course coordinator for providing the best of the facilities for the completion of this work and constant encouragement towards the goal.

I have no words to thank, **Mr. V.N. Shukla**, Director Special Applications, ER&DCI, Noida for his invaluable suggestions during the entire course of this work.

I am highly indebted to my guide **Dr. P.R.Gupta**, Reader, ER&DCI, Noida for the inspiration, constant support and invaluable timely suggestions she provided me throughout the course of this dissertation. I am also grateful to **Mr. Munish Kumar**, Project Engineer, ER&DCI, Noida for the cooperation extended by him in the successful completion of this work.

It is impossible to mention the names of all those persons who have been involved, directly or indirectly, with this work and I extend my gratitude to all of them. However, I feel, I owe special thanks to all my friends who have helped me formulate my ideas and have been a constant support. I find myself short of words to thank my father, mother and sister who have always been by my side throughout my life.

A. Deepak Babu.
(Ayithepalli Deepak Babu)
Enrollment. No. 019016

CONTENTS

| | Page. no |
|--|-------------|
| Candidate's Declaration | (i) |
| Acknowledgement | (ii) |
| Abstract | 1 |
| 1. Introduction | 3 |
| 1.1. Overview | |
| 1.2. Objective of Dissertation | 3 |
| 1.3. Scope of Dissertation | 4 |
| 1.4. Organization of Dissertation | 4 |
| 2. Literature Survey | |
| 2.1. Introduction to Network Management | 5 |
| 2.2 . Fundamentals of Network Management | 6 |
| 2.3 . Types of Network Management | 8 |
| 2.3.1 OSI Network Management | 8 |
| 2.3.2 Internet Management | 9 |
| 2.4. SNMP | 9 |
| 2.4.1 Background | 9 |
| 2.4.2 Descriptions and Definitions | 9 |
| 2.4.3 Introduction to SNMP Protocol | 10 |
| 2.4.4 SNMP Architecture | 15 |
| 2.4.5 SNMP Protocol specification | 17 |
| 2.4.6 Functions in SNMP | 18 |
| 2.4.7 Underlying Communication Protocols | 20 |
| 2.4.8 Strengths and Shortcomings of SNMP | 23 |
| 3. Analysis and Design | 25 |
| 3.1 Problem Analysis | 25 |
| 3.2 Problem Specifications | 25 |
| 3.3 Architectural Design | 26 |
| 3.4 Structural Design | 27 |

| | |
|---|-----------|
| 4. Implementation Details | 29 |
| 4.1. Simulation Model | 29 |
| 4.2. Design & Algorithms of Proposed Architecture | 30 |
| 4.3 Algorithms | 31 |
| 5. Results and Discussion | 37 |
| 6. Conclusion | 43 |
| References | 45 |
| Appendix A | 47 |
| Appendix B | 49 |

ABSTRACT

Computers and the networks that connect them are highly pervasive and have infiltrated almost every aspect of our daily lives. Technology, when it works, becomes transparent and invisible to the end-user. When technology doesn't work, or stops functioning, complaints and serious consequences are sure to follow.

Due to the move towards global open networks, with the number of hosts on the network in the hundreds of thousands, and the number of individual networks in the thousands, it is no longer possible to rely on a small group of network experts to solve management problems. What was required was a standardized protocol with far more functionality than ping and yet one that could be easily learned and used by a wide variety of people with network management responsibilities.

This project aims in simulating the Simple Network Management Protocol (SNMP) which is a communication Protocol that has gained widespread acceptance and has become the defacto standard for internetwork management. SNMP was developed by IETF(Internet Engineering Task Force) and is applicable to any TCP/IP network, as well as other types of networks.

1.3 Scope of the work

As the initial versions of SNMP did not thoroughly address security of network management, the security framework can be implemented to the existing SNMP standards. Though the first versions of SNMP did address some authentication and access control issues based on the MIB concept, However, one can extend this by also providing integrity, confidentiality, more sophisticated authentication and remote configuration and administration capabilities[2].

Thus the security in network management can be achieved with two different proposed models of security:

- The User-based Security Model (USM)
- The View-based Access Control Model (VACM)

where USM defines the SNMP message-level security while VACM brings group based access control to the authenticated user[3].

1.4 Organization of Dissertation

This Report is structured as follows: In Chapter 2, Introduction to Network Management and it's fundamental issues are introduced initially before going into the SNMP Protocol. The actual protocol SNMP is discussed from chapter 2.4 which gives a brief introduction to SNMP protocol, SNMP basics, definitions and description. The architecture of SNMP, it's functions and Underlying Protocols are discussed in the same chapter 2. Chapter 3 named Analysis and Design gives the practical details in implementing the protocol. This Chapter starts with the actual analysis of the problem and covers various design models of SNMP. The actual implementation of the SNMP which forms the basis of our Dissertation is discussed in Chapter 4. The proposed Architecture and Algorithms used in Implementation are discussed in Chapter 4 followed by the actual results and discussion in chapter 5. Chapter 6 concludes the report.

LITERATURE SURVEY

2.1 Introduction to Network Management

Network management is about the complex issues of remote control, monitoring and configuration of the various devices and resources in a communication network. Network management means different things to different people[1]. In some cases, it involves a solitary network consultant monitoring network activity with the use of a protocol. In other cases, network management involves a distributed database, auto polling of network devices, and high-end workstations generating real-time graphical views of network topology changes and traffic. In general, network management is a service that employs a variety of tools, applications, and devices to assist human network managers in monitoring and maintaining networks.

A Historical Perspective

The early 1980s saw tremendous expansion in the area of network deployment. As companies realized the cost benefits and productivity gains created by network technology, they began to add networks and expand existing networks almost as rapidly as new network technologies and products were introduced. By the mid-1980s, certain companies were experiencing growing pains from deploying many different (and sometimes incompatible) network technologies.

The problems associated with network expansion affect both day-to-day network operation management and strategic network growth planning. Each new network technology requires its own set of experts. In the early 1980s, the staffing requirements alone for managing large, heterogeneous networks created a crisis for many organizations. An urgent need arose for automated network management (including what is typically called network capacity planning) integrated across diverse environments.

2.2. Fundamentals of Network Management

The fundamentals of network and system management have changed a lot during the last ten years. Today the networks are global and open multivendor client-server networks that must provide communication services to a broad range of different applications having specific requirements. Network management is vital for optimized, controlled, and cost efficient utilization of network resources. It covers issues like performance analysis, error reporting, configuration and accounting. Network management can be divided into five function areas as [1]

(FCAPS):

- Fault management
- Configuration management
- Accounting
- Performance management
- Security management

Fault management

The goal of fault management is to detect, log, notify users of, and (to the extent possible) automatically fix network problems to keep the network running effectively. Fault management handles fault detection, isolation and correction. To monitor the network, and to detect and perform error recovery tasks, different event, alarm and reporting mechanisms are used. Because faults can cause downtime or unacceptable network degradation, fault management is perhaps the most widely implemented of the ISO network management elements.

Fault management involves first determining symptoms and isolating the problem. Then the problem is fixed and the solution is tested on all-important subsystems. Finally, the detection and resolution of the problem is recorded.

Configuration Management

The goal of configuration management is to monitor network and system configuration information so that the effects on network operation of various versions of hardware and software elements can be tracked and managed. Configuration management handles configuration of the network. Hence, it handles parameterization and configuration of network devices and manages the dynamic changes of the network setup. subsystems of configuration Management store this information in a database for easy access. When a problem occurs, this database can be searched for clues that may help solve the problem.

Accounting Management

The goal of accounting management is to measure network utilization parameters so that individual or group uses on the network can be regulated appropriately. Such regulation minimizes network problems (because network resources can be apportioned based on resource capacities) and maximizes the fairness of network access across all users. As with performance management, the first step toward appropriate accounting management is to measure utilization of all important network resources. Analysis of the results provides insight into current usage patterns, and usage quotas can be set at this point. Some correction, of course, will be required to reach optimal access practices. From this point, ongoing measurement of resource use can yield billing information as well as information used to assess continued fair and optimal resource utilization.

Performance Management

Performance management is rather network technology dependent. However, generally it handles performance analyzes and management of traffic parameters to achieve maximized utilization of the network resources. The goal of performance management is to measure and make available various aspects of network performance so that internetwork performance can be maintained at an acceptable level. Examples of

performance variables that might be provided include network throughput, user response times, and line utilization.

Performance management involves three main steps. First, performance data is gathered on variables of interest to network administrators. Second, the data is analyzed to determine normal (baseline) levels. Finally, appropriate performance thresholds are determined for each important variable so that exceeding these thresholds indicates a network problem worthy of attention.

Management entities continually monitor performance variables. When a performance threshold is exceeded, an alert is generated and sent to the network management system. Each of the steps just described is part of the process to set up a reactive system. When performance becomes unacceptable because of an exceeded user-defined threshold, the system reacts by sending a message. Performance management also permits proactive methods: For example, network simulation can be used to project how network growth will affect performance metrics. Such simulation can alert administrators to impending problems so that counteractive measures can be taken.

Security Management

Security management provides means to support and manage the security policy and mechanisms of the network. This means alarm handling, event management and different mechanisms to handle authentication, access control, key management and logs. It's goal is to control access to network resources according to local guidelines so that the network cannot be sabotaged (intentionally or unintentionally).

2.3 Types of Network Management

2.3.1. OSI network management

OSI network management defines the Common Management Information Protocol (CMIP) and Common Management Information Service(CMIS). CMIS is the network management service of every network element while CMIP is the corresponding protocol. CMIP is a connection-oriented protocol that works on the application layer.

2.3.2 Internet management

The Internet management protocol is the Simple Network Management Protocol (SNMP) which defines a framework that provides means to organize system elements into managing system entities and managed agents. It also defines a format for representing management information (Management Information Base – MIB) and the protocol used for exchange of management data. SNMP is a connectionless protocol that works above the transport layer.

2.4 SNMP

2.4.1 Background

Network management system contains two primary elements: a manager and agents. The Manager is the console through which the network administrator performs network management functions. Agents are the entities that interface to the actual device being managed. Bridges, Hubs, Routers or network servers are examples of managed devices that contain managed objects. These managed objects might be hardware, configuration parameters, performance statistics, and so on, that directly relate to the current operation of the device in question. These objects are arranged in what is known as a virtual information database, called a management information base, also called MIB.

2.4.2 Descriptions/Definitions

SNMP (Simple Network Management Protocol)

The internet standard protocol developed to manage nodes on an IP network. The protocol has a consistent set of monitoring and control variables, and format for any monitored device: a structure of management information (SMI) and a management information base (MIB).

MIB (Management Information Base)

A database of managed objects accessed by network management protocols. A SNMP MIB is a set of parameters which a SNMP management station can query or set in the SNMP agent of a network device (e.g. router).

SMI (Structure of Management Information)

The rules used to define the objects that can be accessed via a network management protocol. The SMI identifies the data types that can be used in the MIB and specifies how resources within the MIB are represented and named.

ASN.1 (Abstract Syntax Notation One)

A formal language used to define syntax. In the case of SNMP, ASN.1 notation is used to define the format of SNMP protocol data units and of objects.

PDU (Protocol Data Unit)

A packet of data passed across a network.

BER (basic encoding rules)

An encoding specification developed and standardized by CCITT (X.209) and ISO (ISO 8825). It describes a method for encoding values of each ASN.1 type as a string of octets.

2.4.3 Introduction to SNMP

The model of network management with SNMP includes the following: management station, management agents, management information base, and network management protocol.

A management station is typically a stand-alone device which serves as the interface for the human network manager into the network management system. The station must be equipped with management applications, an interface for the human network manager, capability to transmit the network manager's commands by use of the network management protocol, and a data base of information extracted from the MIBs. The management agents are network devices, such as hosts, bridges, routers, and hubs,

that are capable to communicate with and are monitored by the management station. An agent responds to the station's requests, performs an action (on itself) commanded by the management station, and may send messages to the station without a request.

A management information base (MIB) is a collection of objects, which represent the resources or devices' aspects in the network that may be managed. Each object is, essentially, a data variable that represents one aspect of the managed agent.

The fourth piece of a SNMP network management system is the network management protocol, or SNMP. The basic functions of SNMP are: get, set, and trap. get allows the management station to retrieve object information from an agent. set enables the management station to set the value of objects of an agent. trap allows an agent to notify the management station of unplanned events.

MIB & Objects

A MIB contains a list of objects that are related by an OBJECT IDENTIFIER type, which serves as the name of the object. MIBs are arranged in a hierarchical or tree structure. There are 2 ways to access the value of a MIBs object.

Example of a sysObjectID:

short way: 1.3.6.1.2.1.2;

long way: iso(1) org(3) dod(6) internet(1) mgmt(2) mib-2(1) system(2).

Every object within an SNMP MIB is defined in a format way: the definition specifies the data type of the object, its allowable forms and value ranges, and its relationship to other objects within the MIB. The ASN.1 notation is used to define each individual object and also to define the entire MIB structure.

There are 2 different object data types: universal and application. Universal data types are: integer, octetstring, null, object identifier, and sequence, sequence-of. The application types are: networkaddress, ipaddress, counter, gauge, timeticks, and opaque. To define the objects themselves, ASN.1 form is used. The basic building block of an ASN.1 specification is the module. Modules have the basic form

```
<modulereference> DEFINITIONS ::=
    BEGIN
        EXPORTS
        IMPORTS
        AssignmentList
    End
```

The modulereference is a module name followed optionally by an object identifier to identify the module. The EXPORTS construct indicates which definitions in this module may be imported by other modules. The IMPORTS construct indicates which type and value definitions from other modules are to be imported into this module. The assignment list consists of type assignments, value assignments, and macro definitions. Type and value assignments have the form

```
<name> ::= <description>
```

Architecture

- All TCP/IP layers below the application layer have their own SNMP components
- Manufacturers also implement the corresponding management components in their hardware product
- Extensions have been made by manufacturers to cover proprietary protocols and non-TCP/IP equipment
- Many number of MIBs and one server

Server Possessing

- requests are limited to simple fetch / store operations
- can have multiple operations in a single request

(as shown in Fig 2.4.1)

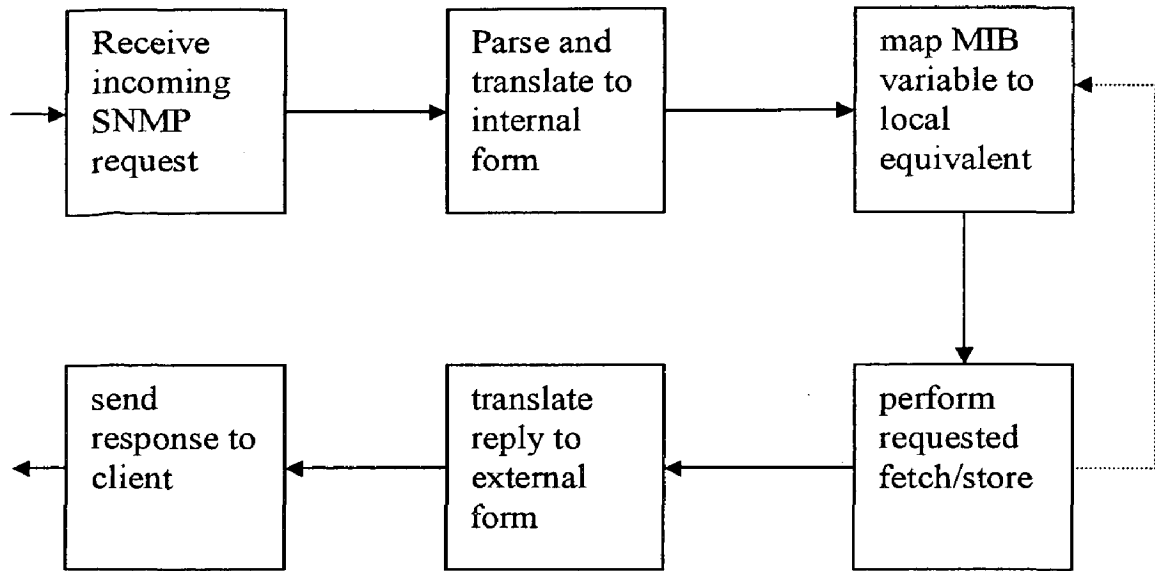


Fig 2.4.1 Flow of an SNMP message through a server

ASN.1

- Abstract Syntax Notation (ASN.1) is a high level computer language used to to define MIB objects
- SNMP uses a subset of the ASN.1 basic types: integer, octet, string, object identifier, sequence, and null
- the Basic Encoding Rules (BER) are used to compile the ASN.1 language into a sequence of octets used for communication between SNMP agents
- ASN.1 defines a lexicographic ordering among variable names and defines a hierarchical namespace
- lexicographic ordering allows the server to answer requests without knowing the size of the variable requested
- all variable names contain the suffix: iso.org.dod.internet.mgmt.mib or 1.3.6.1.2.1

The above is illustrated in Fig 2.4.2

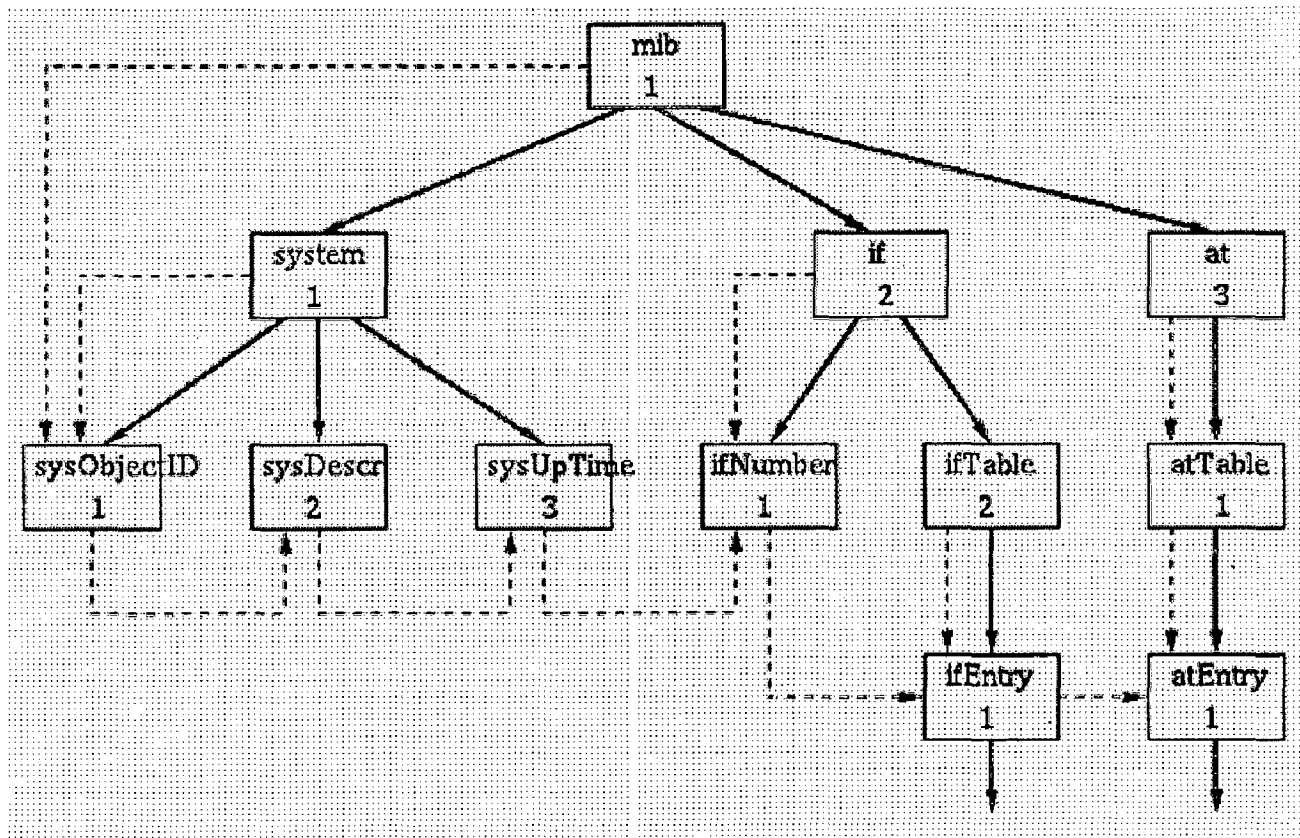


Fig 2.4.2. Lexicographic Ordering of a partial MIB Tree

The SNMP Standard

SNMP can be viewed many different ways. One perspective is to regard SNMP as three distinct standards[4].

A Standard Message Format.

SNMP is a standard communication protocol that defines a UDP message format. This part of the standard is highly involved, and is of little consequence to users (but of great interest to SNMP programmers.)

A Standard Set Of Managed Objects.

SNMP is a standard set of values (referred to as SNMP “objects”) that can be queried from a device. Specifically, the standard includes values for monitoring TCP, IP, UDP, and device interfaces. Each manageable object is identified with an official name, and also with a numeric identifier expressed in dot notation.

A Standard Way Of Adding Objects.

Certainly, one reason that SNMP has become popular and the industry standard is that a method was originally defined so that the standard set of managed objects (above) could be augmented by network device vendors with new objects specific for a particular network.

2.4.4 SNMP Architecture

The model of network management architecture given in the Fig 2.4.3 below shows two Network Management Stations(NMS) , connected in the network to SNMP (Agents). The SNMP agent comprise of an SNMP Agent Protocol engine and Management Information Base (MIB).

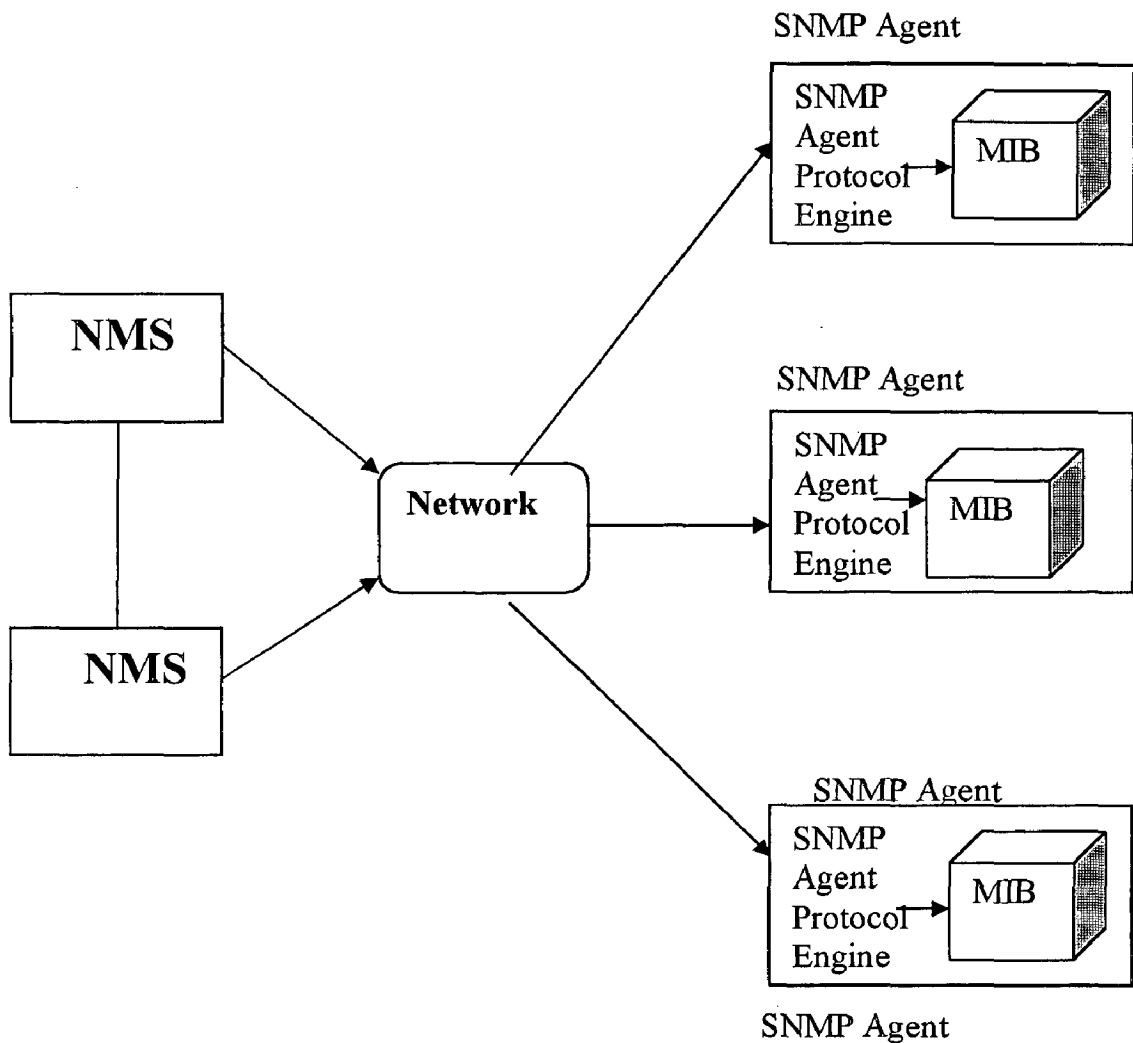


Fig 2.4.3: The model of network management architecture

A typical agent does:

- Implement full SNMP protocol.
- Store and retrieve management data as defined by the Management Information Base - MIB
- Signals asynchronously an event to the manager
- Works as a proxy for some non-SNMP manageable network node.

A typical manager is:

- Implemented as a Network Management Station (the NMS)
- Implements full SNMP Protocol
- Able to Query agents
- Get responses from agents
- Set variables in agents
- Acknowledge asynchronous events from agents

2.4.5. SNMP Protocol Specifications

The peer processes which implement the SNMP, and thus support the SNMP application entities, are called protocol entities. Communication among protocol entities is accomplished using messages encapsulated in UDP datagrams. An SNMP message consists of a version identifier, an SNMP community name, and a protocol data unit (PDU). FIG 2.4.4 shows encapsulated SNMP message. The Version and community name are added to the data PDU and along with the application header the entire message is passed on to the transport layer as SNMP PDU. The UDP header is added at the transport layer, which then forms the transport PDU for the network layer. Addition of an IP header to the transport PDU forms the network PDU for the data link layer. The network or data link layer (DLC) header is added before the frame is transmitted on to the physical medium[5].

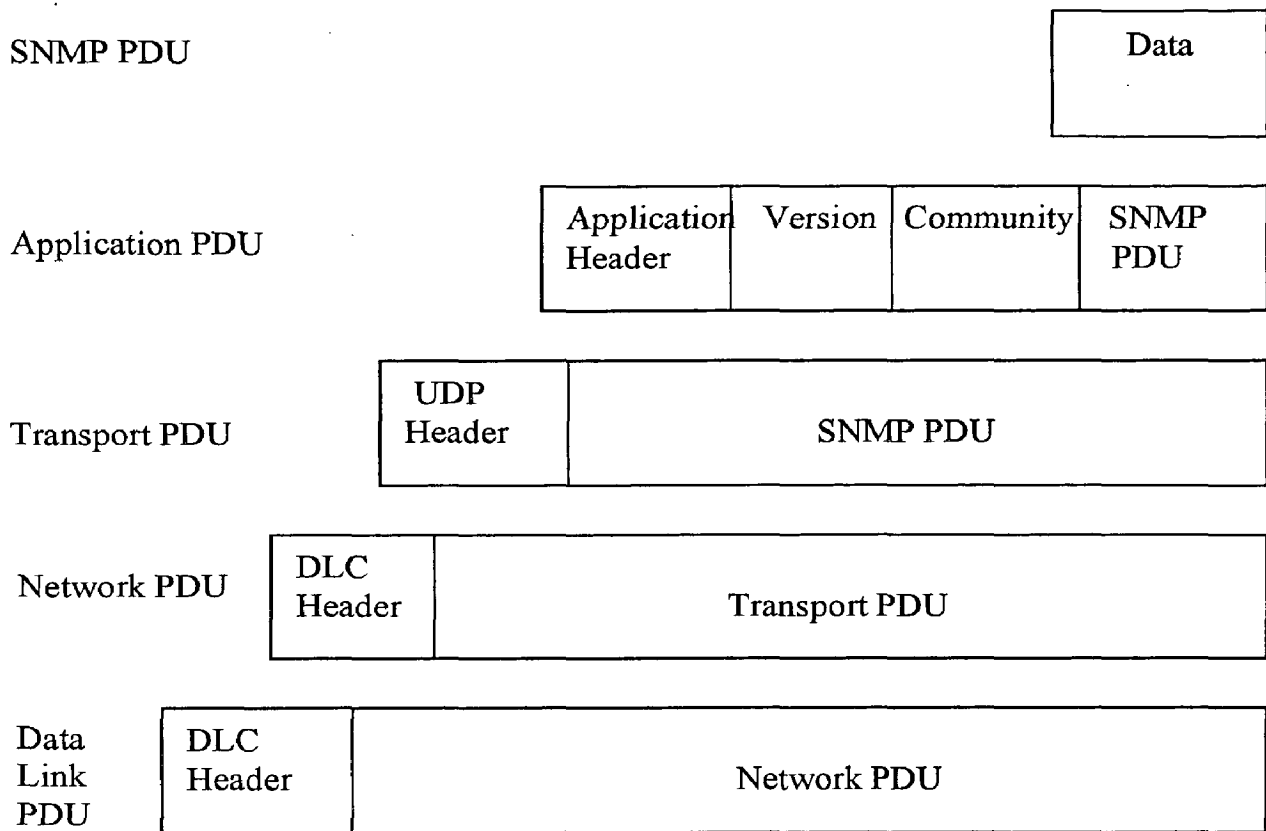


Fig 2.4.4. Encapsulated SNMP Message

2.4.6 Functions in SNMP

SNMP is based on the manager/agent model. SNMP is referred to as "simple" because the agent requires minimal software. Most of the processing power and the data storage resides on the management system, while a complementary subset of those functions resides in the managed system.

To achieve its goal of being simple, SNMP includes a limited set of management commands and responses. The management system issues Get, GetNext and Set messages to retrieve single or multiple object variables or to establish the value of a single variable. The managed agent sends a Response message to complete the Get, GetNext or Set. The managed agent sends an event notification, called a trap to the

managements system to identify the occurrence of conditions such as threshold that exceeds a predetermined value. In short there are only five primitive operations:

- get (retrieve operation)
- get next (traversal operation)
- get response (indicative operation)
- set (alter operation)
- trap (asynchronous trap operation)

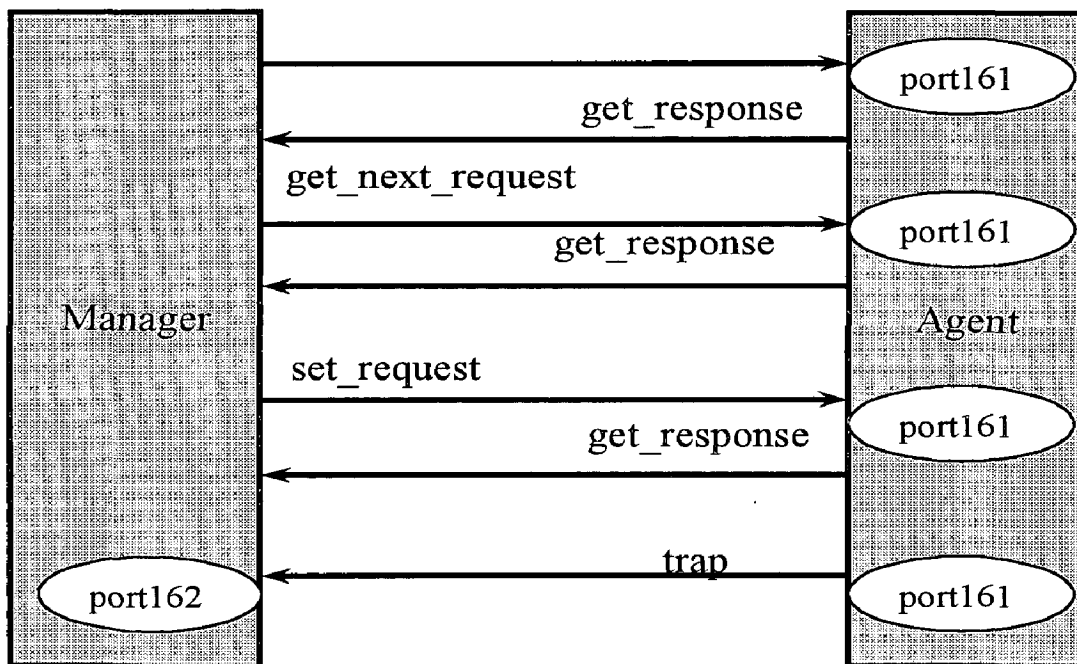


Fig 2.4.5. SNMP Functions

The functions in the above Fig 2.4.5 works as follows:

- a client can issue three basic commands: set-request, get-request, and get-next-request
- the set-request and get-request causes the server to perform a straightforward mapping

- the get-next-request does not specify a variable name to retrieve, instead it specifies a variable name and the server returns the value of the next variable in lexicographical order
- the get-next-request command is useful for accessing values in a table of unknown size (e.g. arp table and routing table)
- the process of stepping through entries one at a time is called "walking the tree"
- the get-bulk-request is intended to reduce network traffic by requesting large amounts of data instead of multiple get-requests or get-next-requests
- a client can also send snmp-trap message asking the server to send an alarm to the client when the specified variable has changed

all trap messages are sent to port 162 while all other PDUs use port 161.

2.4.7. Underlying communication protocols

SNMP assumes that the communication path is a connectionless communication sub network. In other words, no prearranged communication path is established prior to the transmission of data. As a result , SNMP makes no guarantees about the reliable delivery of the data. Although in practice most messages get through , and those that don't can be retransmitted. The primary protocols that SNMP implements are the User Datagram Protocol (UDP) and the Internet Protocol (IP). SNMP also requires Data Link Layer protocols such as Ethernet or Token Ring to implement the communication channel from the management to the managed agent.

SNMP's simplicity and connectionless communication also produce a degree of robustness. Neither the manager nor the agent relies on the other for its operation. Thus, a manager may continue to function even if a remote agent fails. When the agent resumes functioning , it can send a trap to the manager, notifying it of its change in operational status. The connectionless nature of SNMP leaves the recovery and error detection up to

the NMS(Network Management Station) and even up to the agent. However keep in mind that SNMP is actually transport independent (although original design was connectionless transport function, which corresponds to the UDP protocol) and can be implemented on other transports as well:

- TCP (Connected approach)
 - Direct mapping onto Ethernet MAC level
 - Encapsulation onto X25 protocol
 - Encapsulation onto ATM Cell
- and so on.....

The following fig 2.4.6 describes the Transport Mechanism used in SNMP over UDP.

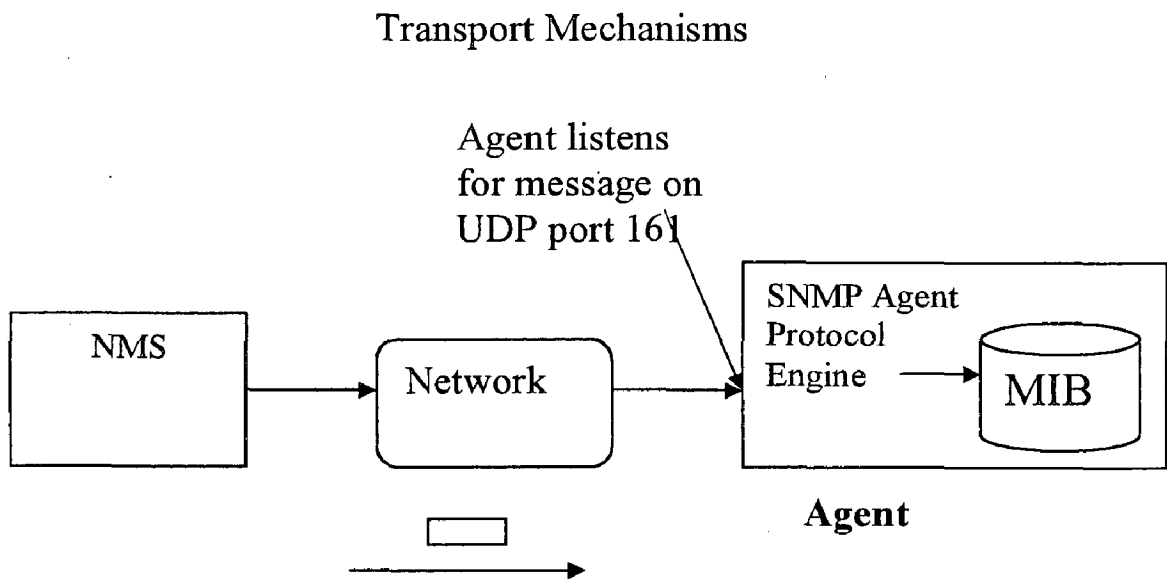


Fig 2.4.6 SNMP Protocol Data Units encapsulated within UDP Datagrams

The following diagrams Fig 2.4.7 & 2.4.8 shows the architecture of UDP transport.

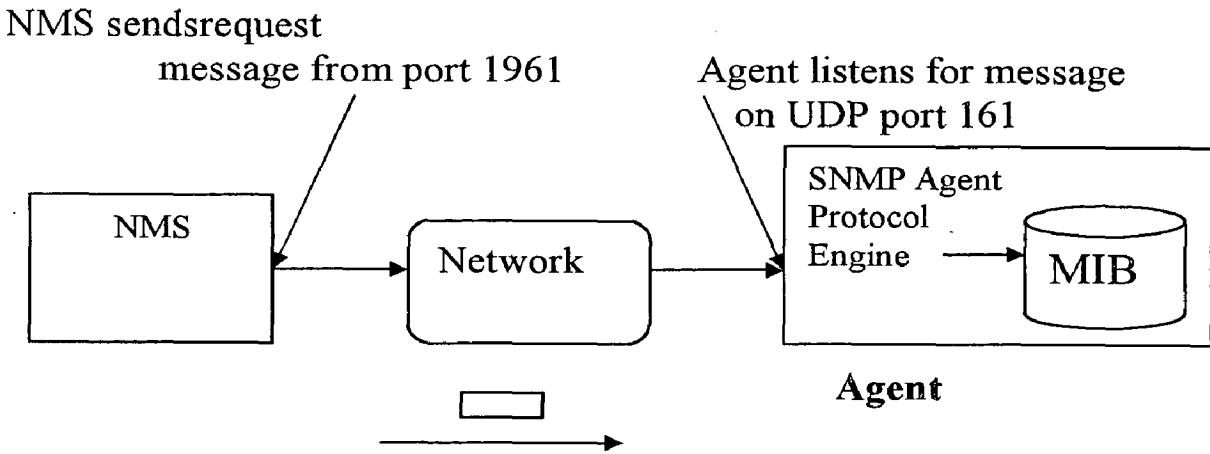


Fig 2.4.7 SNMP Protocol Data Units encapsulated with in UDP Datagrams

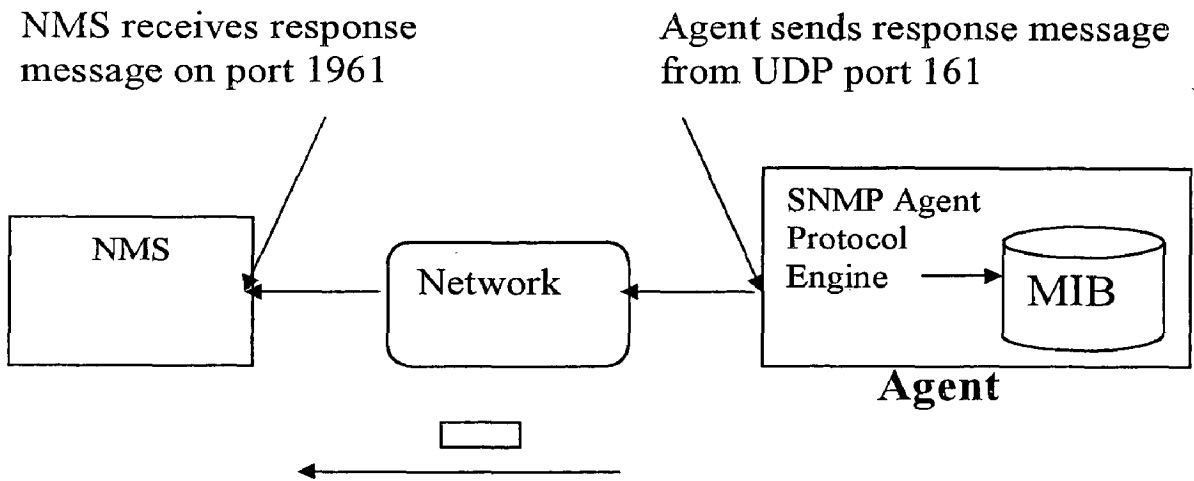


Fig 2.4.8 SNMP Protocol Data units encapsulated within UDP Datagrams

UDP Transport

- Agent listen on UDP port 161
- Responses are sent back to the originating NMS port from a dynamic port , although many agents use port 161 also for this target.
- Maximum SNMP message size is limited by maximum UDP message size (i.e. 65507 octets)
- All SNMP implementations have to receive packets at least 484 octets in length
- Some SNMP implementation will incorrectly or not handle packets exceeding 484 octets
- Asynchronous Traps are received on port 162 of the NMS
- UDP more suitable than TCP when dynamic route changes occur often (e.g. when there are problems in the network)
- UDP packets minimize the demands placed on the network(no resource tied up as with connection mode)
- Agent and NMS are responsible for determining error recovery [5]

2.4.8. Strengths and Shortcomings of SNMP

SNMP has several strengths. Its biggest strength is arguably its widespread popularity. SNMP agents are available for network devices ranging from computers, to bridges, to modems, to printers. The fact that SNMP exists with such support gives considerable credence to its reason for existence; SNMP has become interoperable. Additionally, SNMP is a flexible and extensible management protocol. Because SNMP agents can be extended to cover device specific data, and because a clear mechanism exists for upgrading network management client programs to interface with special agent capabilities (through the use of ASN.1 files), SNMP can take on numerous jobs specific to device classes such as printers, routers, and bridges, thereby providing a standard mechanism of network control and monitoring. Several weaknesses of SNMP can be identified. Despite its name, (i.e. "Simple" Network Management Protocol), SNMP is a highly complicated protocol to implement. By the admission of its designers, a more apt

name might be “Moderate Network Management Protocol”, although even this seems generous in light of SNMP's highly complicated encoding rules. Also, SNMP is not a particularly efficient protocol. Bandwidth is wasted with needless information, such as the SNMP version (transmitted in every SNMP message) and multiple length and data descriptors scattered throughout each message. The way that SNMP variables are identified (as byte strings, where each byte corresponds to a particular node in the MIB database) leads to needlessly large data handles that consume substantial parts of each SNMP message.

The above complaints, while fair, cannot reasonably be used to detract from the aforementioned strengths of SNMP. While complicated encoding rules frustrate programmers who must deal with them, it is easily argued that end users of the protocol aren't concerned with the complexity of the algorithms that handle their data. As for complaints about SNMP efficiency, it has been shown to work well enough, given the nature of network administration activities. In short, SNMP has been shown to work as a management protocol, making criticisms (however well deserved) irrelevant.

ANALYSIS AND DESIGN

3.1 Problem Analysis

In the late 80's, computer networks had grown from a simple layout of small separate networks that were not connected to each other to larger networks that were interconnected.

These larger networks were called internets and their size grew at an exponential rate. The larger the networks became, the more difficult they became to manage (i.e. monitor and maintain) and it soon became evident that the network management protocol need to be developed.

The growing complexity of networks has posed a serious problem to network managers and administrators. Several standards have been proposed to assist managers to monitor and "tune" networks from a central location.

3.2 Problem Specification

The problem is to design and implement Network Management Station(Clients) and Agents, the relationship between whom is as given below in Fig 3.1.1

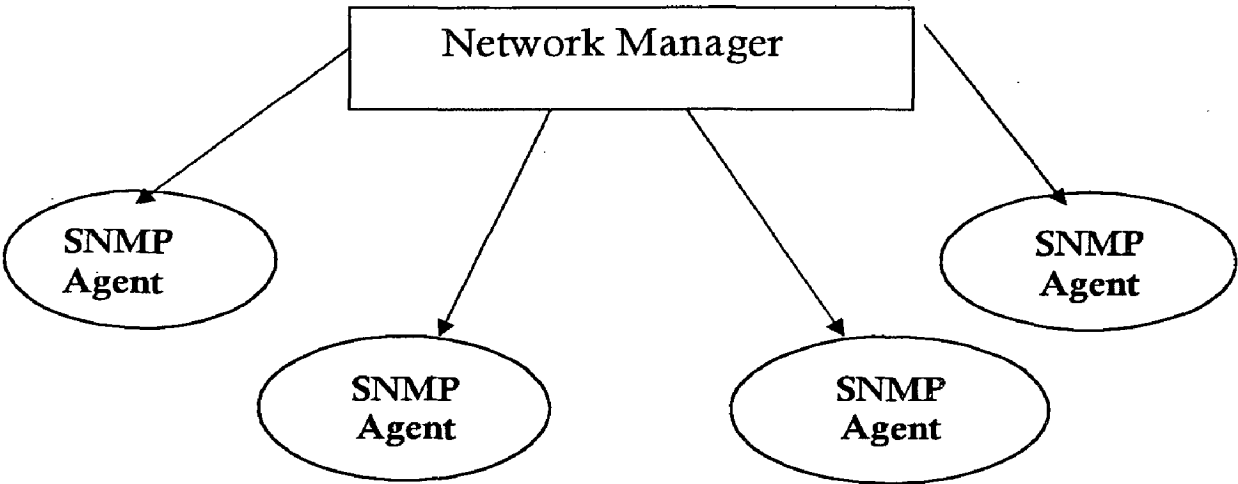
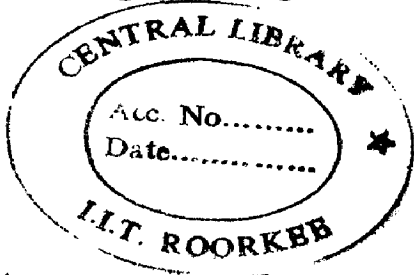


Fig 3.1.1 Relationship between Network Manager and Agents



The primary task is to design and implement a protocol that would facilitate the communication between the network manager and agents.

3.3 Architectural Design

The SNMP architecture gives a solution to the network management problem in terms of

- 1) The representation of management information communicated by the protocol (ASN.1)
- 2) Operations on management information supported by the protocol (Set, Get).
- 3) The form and meaning of exchange among management entities (UDP)

SNMP MIB OBJECTS:

SNMP management Information Base (MIB) is the one that defines all the values that SNMP is capable of reading or setting.

The SNMP MIB is arranged in a tree-structured fashion (as shown in the fig below) similar in many ways to a disk directory structure of files. The top level SNMP branch begins with the ISO "internet" directory, which contains four main branches:

The "mgmt" SNMP branch contains the standard SNMP objects usually supported (at least in part) by all network devices;

The "private" SNMP branch contains those "extended" SNMP objects defined by network management vendors; the "experimental" and "directory" SNMP branches, also defined within the "internet" root directory, are usually devoid of any meaningful data or objects.

3.4. Structural Design

The “tree” structure is an integral part of the SNMP standard. However, the most pertinent parts of the tree are the “leaf” objects of the tree that provide actual management data regarding the device. Generally, SNMP leaf objects can be partitioned into two similar but slightly different types that reflect the organization of the tree structure. They are,

- 1) Discrete MIB objects – Which contain one precise piece of management data.
- 2) Table MIB objects - Which contain multiple pieces of management data.

which are shown below in Fig 3.4.1

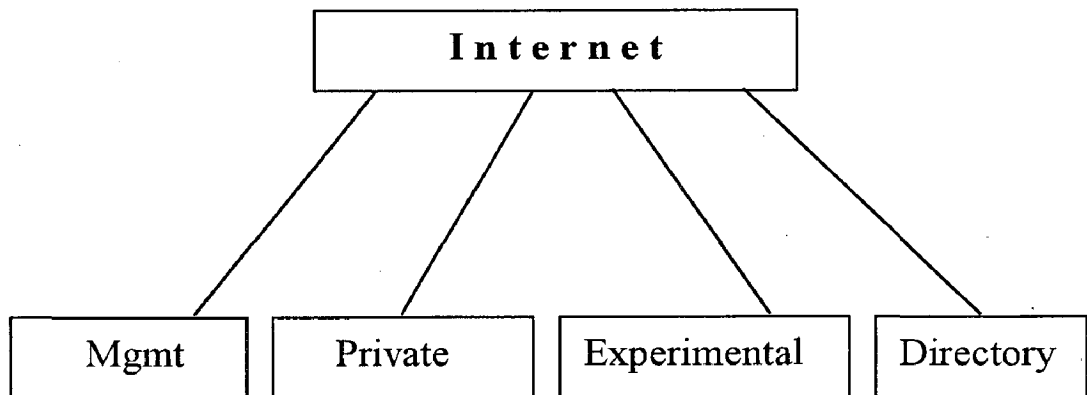


Fig 3.4.1 SNMP MIB Tree Structure

In this Dissertation, the overhead of using MIB in the tree structure through ASN.1 notation is reduced by the use of simple files for MIB.

IMPLEMENTATION DETAILS

4.1 The Simulation Model

In this project the following modules were implemented:

---A Chat Server was programmed which is considered analogous to a device to be managed.

---A Chat client program was written which is considered analogous to the users of the device.

---An agent was designed which gives necessary information about the device to be monitored whenever queried by the management station.

---A management station has to be implemented to query the status of the device through the agent residing on the device.

4.2 Design and Algorithms of Proposed Architecture

The following (as shown in Fig 4.2.1) is the proposed Design which is implemented. The Procedure/Algorithm of the design is discussed below.

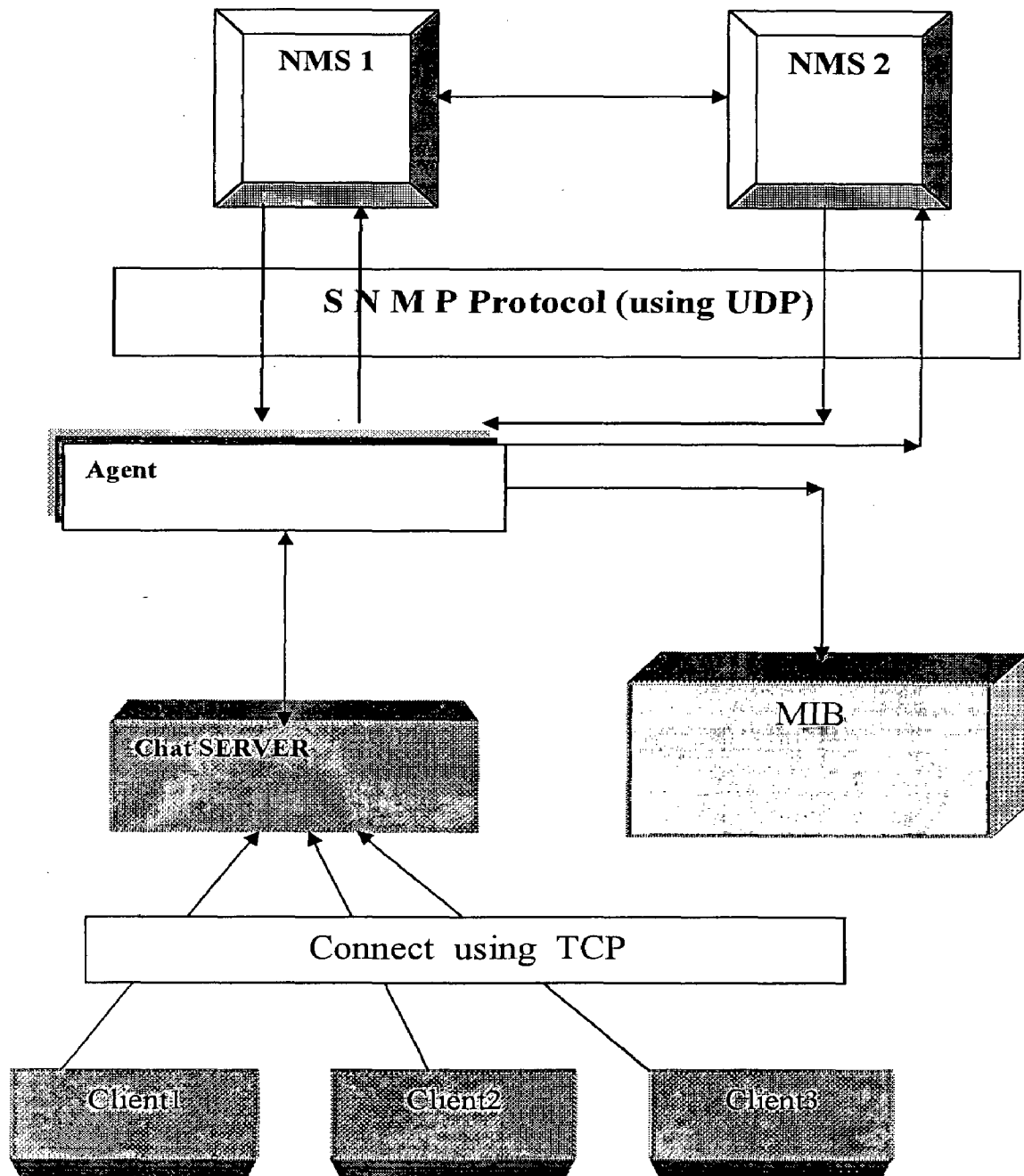


Fig 4.2.1 Proposed Design

4.3 Algorithms (for the Proposed Design):

Step 1. Chat Server is started (the device to be monitored).

Step 2. Chat Server creates a database file (Management Information Base(MIB)).

Step 3. Chat Clients are connected to the server using the Connection oriented TCP transport. (connected as per the procedure mentioned for Client Server(TCP) connection.

Step 4. The MIB of the device to be monitored is updated with the present status of the device.

Step 5. Network Management Station (NMS) is started

Step 6. The agent is started (to fetch the status values of the network) and

is connected to the NMS through the SNMP Protocol (UDP Transport).

Step 7. NMS queries the status of the device being monitored(i.e. Chat Server)

Step 8. Agent opens the MIB of the Chat Server and gets the status values of the required device reported to the NMS again through the SNMP Protocol i.e., UDP Transport.

The procedure for connection-oriented transfer[6] i.e. TCP connection which is established between chat clients and chat server is illustrated(in Fig 4.3.1) as follows.

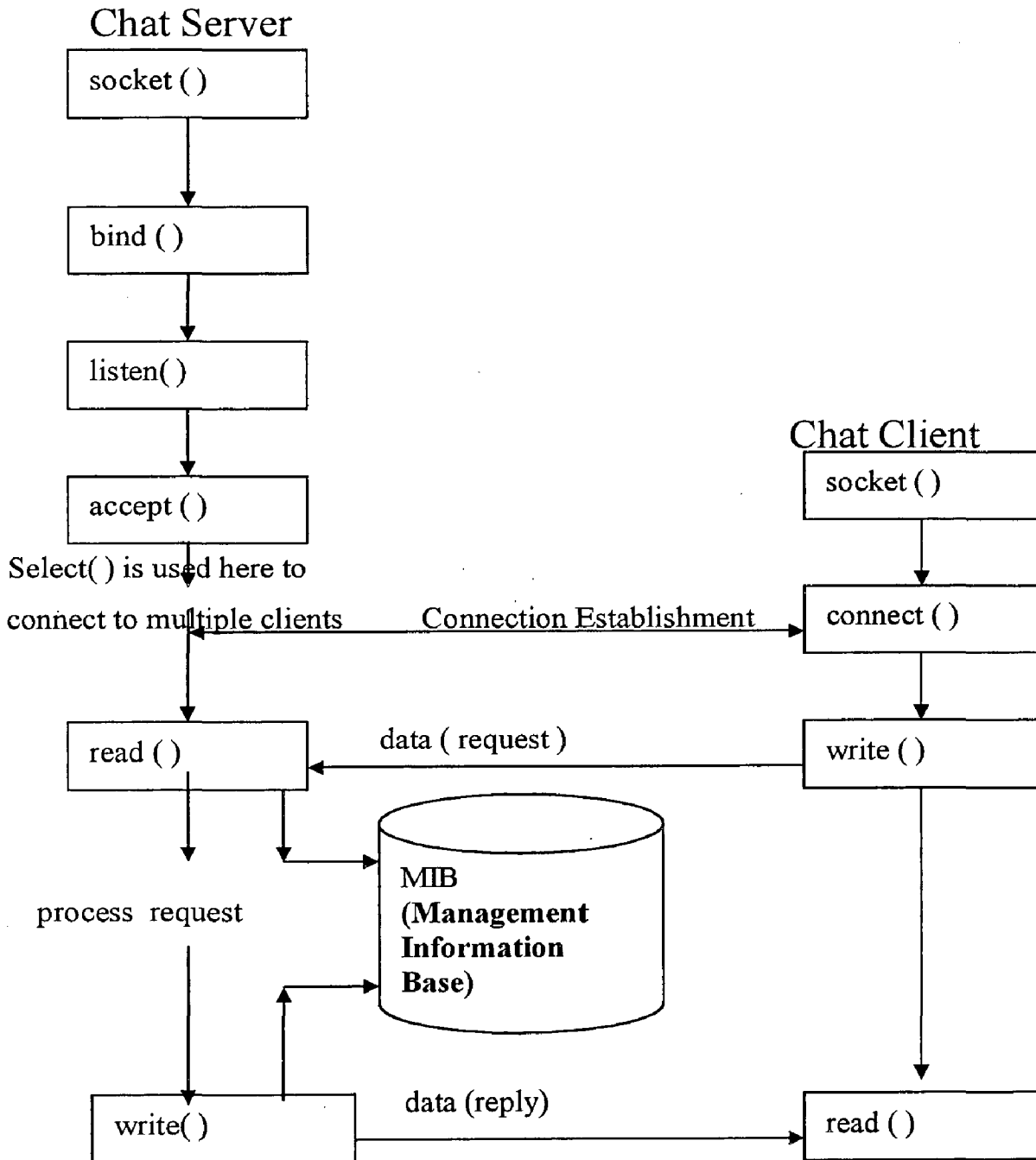


Fig 4.2 Connection oriented Transport(TCP)

Chat Server:

Algorithm: For establishing a connection-oriented transfer between chat client and chat server.

First the Chat server is started, then sometime later a chat client is started that connects to the server.

Step 1 : Allocate space

Step 2 : Create end point (socket())

Step 3 : Bind the address (bind())

Step 4 : Specify Queue (listen())

Step 5 : Wait for connection (accept())

Step 6 : get new fd

In this algorithm, multiple Chat clients are connected to the Chat Server using the Select Command[7]

Chat Clients:

Step 1 : Allocate space

Step 2 : Create end point (socket())

Step 3 : Bind the address (bind())

Step 4 : Connect to the Chat Server(connect())

Step 5 : Transfer data (read (), write ())

Step 6 : Terminate.

Thus multiple clients are connected to the server using Connection-Oriented TCP/IP Transfer.

Illustration of Connectionless Transfer(UDP)[8] established between Network Management Station and Agents (as shown in Fig 4.3 [10]) as follows:

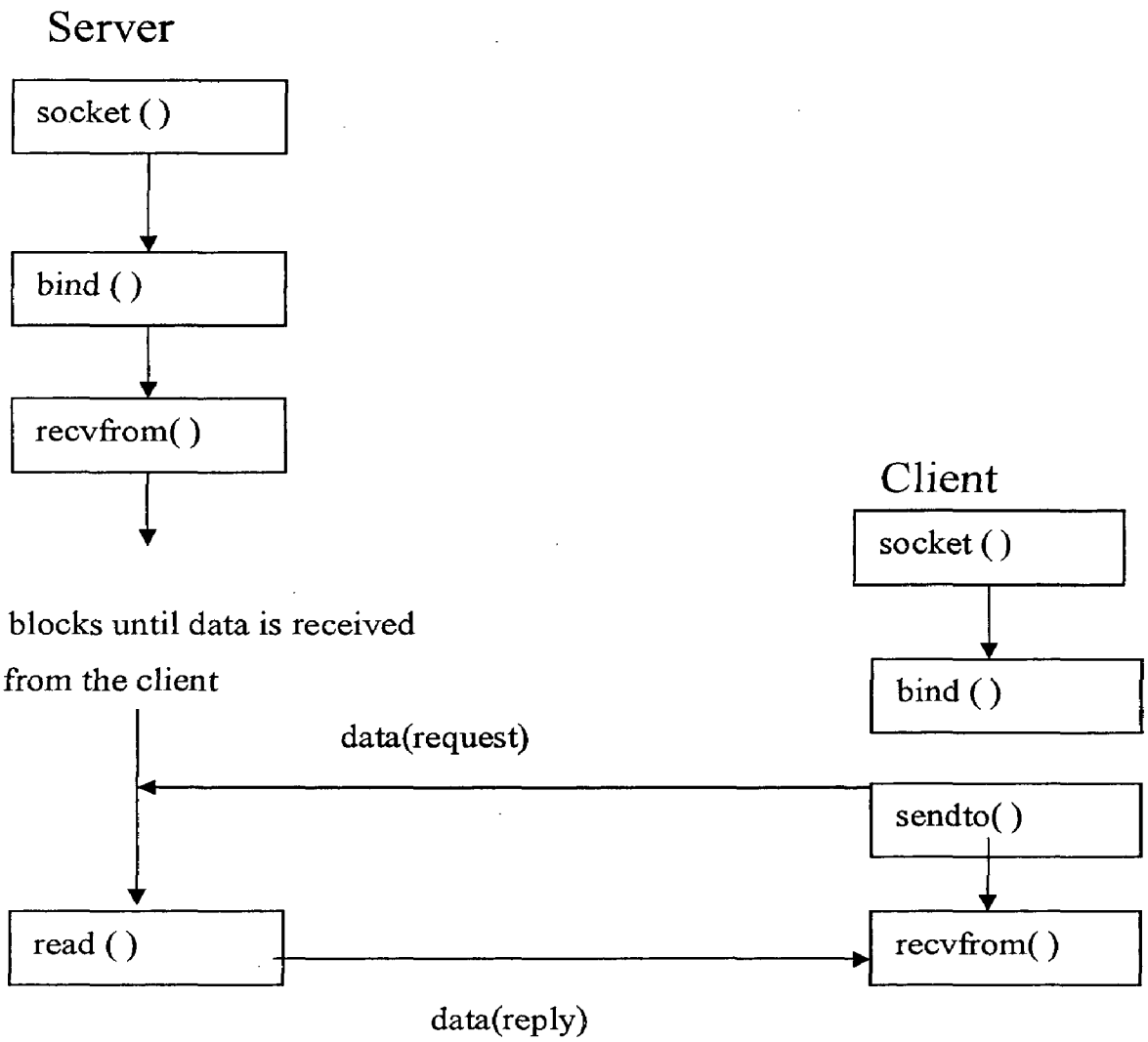


Fig 4.3: Connection less transport (UDP)

Algorithm:

For establishing a connectionless (UDP) transfer between Network Management station and the agent.

First the Agent (which acts as server now) is started, then the management station(as client)is started that connects to the agent to query for the status of the device to be monitored.

Agent :

Step 1 : Allocate space

Step 2 : Create end point (socket())

Step 3 : Bind the address (bind())

Step 4 : Transfer datagrams with the client

Management Station :

Step 1 : Allocate space

Step 2 : Create end point (socket())

Step 3 : Bind the address (bind())

Step 4 : Transfer datagrams with the server

RESULTS AND DISCUSSION

Initially the Chat Server is compiled, and it is made listen to port e.g. 1234. The output Screen of the Chat Server is given below.

[sridhar@localhost sridhar]\$ **cc -o server server.c**

[sridhar@localhost sridhar]\$ **./server 1234**

Chat Server is started..
MIB Database Created Successfully
File Descriptor is created
Bind Is Successful

Now the Chat Clients are connected to the Server with Connection – oriented transfer(TCP/IP) at the port no 1234 as follows:

[sridhar@localhost sridhar]\$ **cc -o myclient myclient.c**

[sridhar@localhost sridhar]\$ **./myclient 1234**

Connected to the Server

As soon as the Chat client gets connected to the Chat server, the MIB (Management Information Base) of the Device being monitored (i.e., Char Server) is Updated with the 'System Details' and 'TCP Details' as follows:

```
before listen
Accept :Successful
Client ConnFd : -1
Client fd :5
Updating DataBase for New
client
Writing Sys Obj to database
Writing Tcp Obj to database
```

Now another Chat Client is connected to the Chat Server

```
[sridhar@localhost sridhar]$ cc -o myclient myclient.c
[sridhar@localhost sridhar]$ ./myclient 1234
```

```
Connected to Server
```

As soon as another Chat client gets connected to the Chat server, the MIB (Management Information Base) of the Char Server) is Updated with the latest 'System Details' and 'TCP Details' as follows:

```
before listen
Accept :Successful
Client ConnFd : -1
Client fd :6
Updating DataBase for New
client
Writing Sys Obj to database:
Writing Tcp Obj to database:
```

Now the Agent, which acts as the SNMP server over UDP connection , is activated on the network at a different port eg.1235.

```
[sridhar@localhost sridhar]$ cc -o agent agent.c
[sridhar@localhost sridhar]$ ./agent 1235
```

```
Agent is active in the network
```

Now we operate from the Network Management Station , which now acts as the SNMP client in over UDP Transport, to query the Agent regarding the Status values of the Device to be monitored i.e., Chat Server.

[sridhar@localhost sridhar]\$ cc -o mgmt mgmt.c

[sridhar@localhost sridhar]\$./mgmt 1235

```
Know system details ---> 1
Know TCP details ---> 2
Exit ---> 3
*****

Enter your choice ---> :
```

```
Enter your choice ---> :1
SysUptime =Sun Feb 23 23:55:15 2003
SysContact =Contact: Deepak, Telephone : 9811627344
SysName =ER&DCI
Know system details ---> 1
Know TCP details ---> 2
Exit ---> 3

*****

Enter your choice ---> :
```

The 'System Details'[Appendix A] of the network and the Chat Server are displayed as above for option 1.

Enter your choice ---> :2

```
Max conn      =20
Curr conn     = 2
TCPInSegs    =77
TcpOutSegs   =77
Know system details ---> 1
Know TCP details ---> 2
Exit          ---> 3
*****
Enter your choice ---> :
```

The current Status values of the Network and Chat Server like its TCP details such as 'Total no of current connections', 'no. of incoming TCP packets' and 'no. of outgoing TCP 'packets etc.. are displayed[Appendix B].

```
sridhar@localhost sridhar]$ cc -o mgmt mgmt.c
```

```
[sridhar@localhost sridhar]$ ./mgmt 1235
```

```
Know system details ---> 1
Know TCP details ---> 2
Exit ---> 3
*****
Enter your choice ---> :3
```

Thus, the choice '3' stops the UDP Communication between SNMP Manager and SNMP Agent with a trap .

```
[sridhar@localhost sridhar]$
```

CONCLUSION

In this project Simple Network Management Protocol is successfully simulated. The Client-Server relationship that is defined by the protocol has been clearly shown in the model. The model developed uses the same communication protocol used by SNMP for the purpose of communication between the agent and management station via, UDP.

Through this project, a low-overhead solution is provided for short term network management needs with the help of simple files for MIB instead of using a complex ASN.1 notation which is used in the actual protocol. Client-Server applications over TCP and UDP are created that implement objects present in the TCP and Systems Group of SNMP with the help of sockets API for the network communication part.

REFERENCES

- [1] Stevenson D., Network Management, What it is and what it isn't, April 1999,
available via WWW at URL:
<http://netman.cit.buffalo.edu/Doc/Dstevenson/>, 1999
- [2] Blumenthal U., et al., User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3), RFC-2574, IETF Network Working Group, April 1999.
- [3] Wijnen B., et al., View-based Access Control Model (VACM) for the Simple Network Management Protocol, RFC-2575, IETF Network Working Group, April 1999.
- [4] RFC 1157, Simple Network Management Protocol (SNMP), 1990
- [5] Manisubramanyam., Georgia Institute Of Technology
Network Management : Principles and Practices,
Pearson Education, ISBN 81-7808-595-X, May 2002.
- [6] Comer D., Internetworking with TCP/IP, Volume 2: Principles, Protocols, and Architecture, 3rd Edition,
Prentice Hall, ISBN 0-13-227836-7, 2000.
- [7] Case J., et al., Introduction to version 3 of the Internet-standard Network Management Framework,
RFC-2570, IETF Network Working Group, April 1999.

[8] Andrew S. Tanunbaum, “ Computer Networks “,
3rd edition, Prentice Hall of India, 2001.

[9] Network Management & Monitoring with Linux, by David Guerrero
<http://www.develnet.es/~david/papers/snmp>

[10] W. Richard Stevens., “ Unix Network Programming”
Prentice-Hall India, ISBN-81-203-0749-6, August 2001.

Appendix A

Table 1. The System Group

| Entity | OID | Description |
|-------------|----------|--|
| sysDescr | system 1 | Textual description |
| sysObjectID | system 2 | OBJECT IDENTIFIER of the entity |
| sysUpTime | system 3 | Time(in hundredths of a second since last reset) |
| sysContact | system 4 | Contact person for the node |
| sysName | system 5 | Name of the system |
| sysLocation | system 6 | Physical location of the node |
| sysService | system 7 | Value designating the layer services provided by the entity |

Appendix B

Table 2. The TCP Group

| Entity | OID | Description(brief) |
|--------------|--------|---|
| tcpMaxConn | tcp 4 | Maximum no of TCP connections |
| tcpCurrEstab | tcp 9 | No. of connections for which the current state is either ESTABLISHED or CLOSED-WAIT |
| tcpInSegs | tcp 10 | Total number of segments received including with errors |
| tcpOutSegs | tcp 11 | Total number of segments sent excluding retransmission |

