VLAN CONFIGURATION USING SNMP

AT VSSC-ISRO TRIVANDRUM

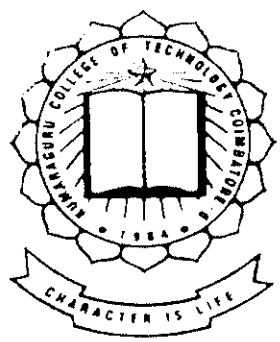
D_480

A Project Report

SUBMITTED IN PARTIAL FULFILMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF ENGINEERING IN
COMPUTER SCIENCE AND ENGINEERING
OF BHARATHIAR UNIVERSITY

SUBMITTED BY
JESTIN RAJAMONY
Reg. No. 9937K0006

Under the Guidance of
Mrs. L.S. Jayashree M.E, MISTE.



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
KUMARAGURU COLLEGE OF TECHNOLOGY
COIMBATORE - 641 006.

2000 - 2001

Department of Computer Science and Engineering
Kumaraguru College of Technology
(Affiliated to the Bharathiar University)
Coimbatore – 641 006

Project Work

CERTIFICATE

Bonafide record of the project work

VLAN Configuration Using SNMP

Done by

JESTIN RAJAMONY
(Reg. No. 9937K0006)

Submitted in partial fulfillment of the requirements
For the degree of Master of Engineering
in Computer Science and Engineering
Of the Bharathiar University

Faculty Guide | **Head of the Department**

Submitted for Viva-Voce Examination held on _____

Internal Examiner | **External Examiner**

साराभाई अंतरिक्ष केन्द्र
म - 695 022
स्पेस

GOVERNMENT OF INDIA
DEPARTMENT OF SPACE
VIKRAM SARABHAI SPACE CENTRE
THIRUVANANTHAPURAM - 695 022
TELEGRAM : SPACE
TELEPHONE :
TELEX :
FAX :

CERTIFICATE

This is to certify that this project work entitled **VLAN Configuration Using SNMP** is a bonafide record of the work done by *Mr. Jestin Rajamony* during the year 2000 under my guidance in partial fulfilment of the requirements for the award of the Degree of Master of Engineering in Computer Science and Engineering of the Bharathiar University.

Mr. Anil S.
Scientist / Engineer 'SD'
FCSD – VSSC
ISRO Trivandrum.

Dr. Omana Mammen
Head
FCSD – VSSC
ISRO Trivandrum.

11/12

Declaration

*I, **Jestin Rajamony** hereby declare that this project work entitled **"VLAN CONFIGURATION USING SNMP"** submitted to **Kumaraguru College of Technology, Coimbatore** (Affiliated to Bharathiar University) is a record of original work done by me under the supervision and guidance of **Mrs. L.S. Jayashree M.E**, Department of Computer Science and Engineering.*

Name of the Candidate	Register Number	Signature of the Candidate
Jestin Rajamony	***9937K0006***	***(Jestin Rajamony)***

Countersigned by: Staff in Charge

Mrs. L.S. Jayashree M.E
Lecturer
Department of Computer Science and Engineering
Kumaraguru College of Technology
Coimbatore – 641 006

Place : Coimbatore
Date : 11-04-2001

ACKNOWLEGEMENT

I express my deep sense of gratitude to ***Dr.K.K. Padmanaban. Ph.D,*** *Principal, Kumaraguru College of Technology, Coimbatore, for providing permission to carry out this project work.*

With profound sense of gratitude and regards, I acknowledge with great pleasure the guidance and support extended by ***Prof. Dr. S. Thangaswamy Ph.D.,*** *Head of the Department of Computer Science and Engineering, Kumaraguru College of Technology, Coimbatore, for his valuable and continuous guidance, suggestions, constructive criticisms and persistent encouragement.*

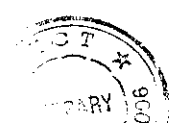
I express my deep sense of respectful gratitude to my guide ***Lecturer. Mrs. L.S. Jayashree M.E.,M I S T E,*** *Department of Computer Science and Engineering for her valuable guidance, keen suggestions, innovative ideas, inspiration, discussions, helpful criticisms and kind encouragement in all the phases of this project work. It had been indeed a great pleasure to work under her guidance.*

It is my duty to express my thanks to ***Asst. Prof. Mr.R. Kannan M.E.,*** *Department of Computer Science and Engineering who has been an all time encouragement not only throughout the project but also during the entire course.*

Also I express my deep sense of gratitude to ***Dr. Mrs. Omana Mammen Ph.D,*** *Head, Flight Computer Checkout System Division, ISRO for her encouragement in this project work. I express my sincere gratitude to the generous help and proper guidance rendered by my guide* ***Mr.S.Anil,*** *SD/Engg, FCSD, ISRO towards my project.*

I extend my sincere thanks to ***Mr. James Xaviour,*** *ISRO who has laid the path and foundation to do this project here successfully. Also I extend my gratitude to* ***Mr. Srinivasan.P.K,*** *SC/Engg. FCSD, ISRO in invoking with some fascinating ideas in doing this project.*

I like to express my special thanks to all persons in the FCSD lab who helped me for the successful completion of the project. Finally I express my deep sense of gratitude to my parents, friends and all other persons who directly or indirectly involved with this project, for their invaluable help and consideration towards us.



SYNOPSIS

By virtually networking some of the network elements it is possible to communicate among these network elements. So it is possible to transfer data among these network elements. The rest of the network elements, which are not virtually networked, remain static temporarily in the Local Area Network. So the speed of response is increased.

Since only some network elements are virtually networked data can be transferred with nearly no any loss of bytes/packets. The chance of duplication of packets is also much less

In order to effectively manage and to monitor the performance of the various elements connected to a network managerial software was developed according to the International standard of Internet Architecture Board. This managerial software could send and receive message from the other computers in the virtual network. The managerial software was developed with a standard protocol, since the agent software, which already existed, was developed, with the same standard protocol. The standard protocols used here are SNMP, UDP, and IP. These protocols are referenced in the Request for Comment, which is available with the IAB. So it is easier for reference. Also Simple Network Management Protocol is used in this product so it is easier to modify the product further, since SNMP is universally accepted and is wide spread.

This product uses DOS environment because it is built for real time systems and DOS operating system is most appropriate for developing real time systems.

The internal designs have been done in object-oriented style, so if any further enhancement is needed calling from the class library can do it. So the time is saved and also the work is reduced. The language used here is C++, a general-purpose object oriented language supporting system-oriented designs.

CONTENTS

Acknowledgement	i
Synopsis	ii
Contents	iii
1. Introduction	01
2. Network Management Protocol	04
2.1. Network Management Protocol Architecture	05
2.2. MIB	07
2.3. UDP	11
2.4. SNMP Format	12
3. Construction	15
3.1. Flow Chart	17
3.2. System Flow	21
3.3. Objects	25
4. Program Design	31
5. Conclusion	
Bibliography	
Appendixes	
Appendix A:	
Appendix B:	

1. INTRODUCTION

A computer network means an interconnected collection of autonomous computers. The primary goals of networking are resource sharing, reliability enhancement and economy. But depending upon the requirement, specialized architectures and protocols are being developed. Basic functionality of computer communication is transfer of data between two systems having their own addresses, where multiple computers exist.

Network management is required in order to maintain proper operation of the complex network. So care must be taken that the system as a whole or individual component are working properly. The model of network management that is used for UDP/IP network management include the following key element:

- ❖ Management station
- ❖ Management agent
- ❖ Management Information Base
- ❖ Network Management Protocol

❖ **Management station:**

It is a stand-alone device and also has the capability of sharing systems. It will have

- ▸ A set of management applications for data analysis, faults recovery, and so on.
- ▸ An interface by which the network manager may monitor and control the network.

- The capability of translating the network manager's requirements into the actual monitoring and control of remote elements in the network
- A database of information extracted from the MIBs of all the managed entities in the network.

❖ **Management agent:**

The active element in the network management system is the management agent. SNMP agent lies in the host, bridges, routers and hubs and makes these devices be managed from a central management station. The management agent responds to the requests for information and actions from the management station.

❖ **Management Information Base:**

Resources in the network could be managed by representing these resources as objects. Each object is a data variable that represents one aspect of the managed agent. The collection of objects is referred to as management information base (MIB).

❖ **Network management protocol:**

The management station and the agents are linked by a network management protocol. For this the management protocol has a set of commands.

WHY NETWORK MANAGEMENT IS NEEDED?

Network components are added to provide users with an increasing range of services. For example as shown in the figure1:

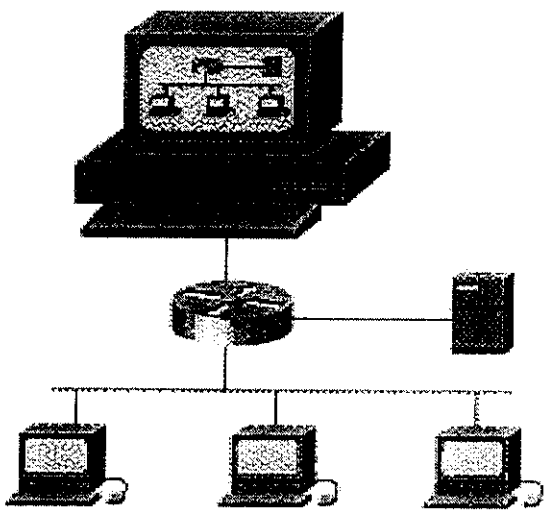


Figure 1

In an organization, suppose if there is only one server and many computers then the server has to be shared by the computers in the network. So this is an advantage of the network.

Suppose, if a network fault has being generated means it will create disruption and generate high costs for the operators. So a proper network management is required to trace out and diagnose the fault.

2. NETWORK MANAGEMENT PROTOCOLS

Protocols are the specified set of rules used for network management. There are several network management protocols used for managing the network terminals. Among these very popular network management protocols SNMP has established top position in the network management. CMIP is another protocol but is not so popular even thou it has many features than the SNMP. SNMPv1 has being used for this project.

SNMP:

This is an enhanced version of Simple Gateway Management Protocol (SGMP). SNMP is abbreviated as Simple Network Management Protocol. It defines the protocol used to manage the objects. It serves as a mechanism to provide and transport management information between network components. It permits interactive network administration via parameter checks or supervision of certain network conditions.

An SNMP agent is a device running some software that understands the language of SNMP. Almost any network device like routers, hubs, switches, and bridges could run SNMP. Some end stations like Unix boxes or network servers may also be running an SNMP agent. If a device is not running an SNMP agent, that device is referred to as *unmanageable*. SNMP protocol lies in the application layer of the UDP/IP network management.

SNMP protocol has some specific format and rules in accordance to the ISO. All management information transferred via the SNMP protocol is shown as non-aggregate object types. These object types are gathered in one or more Management Information Base (MIB).

NETWORK MANAGEMENT PROTOCOL ARCHITECTURE:

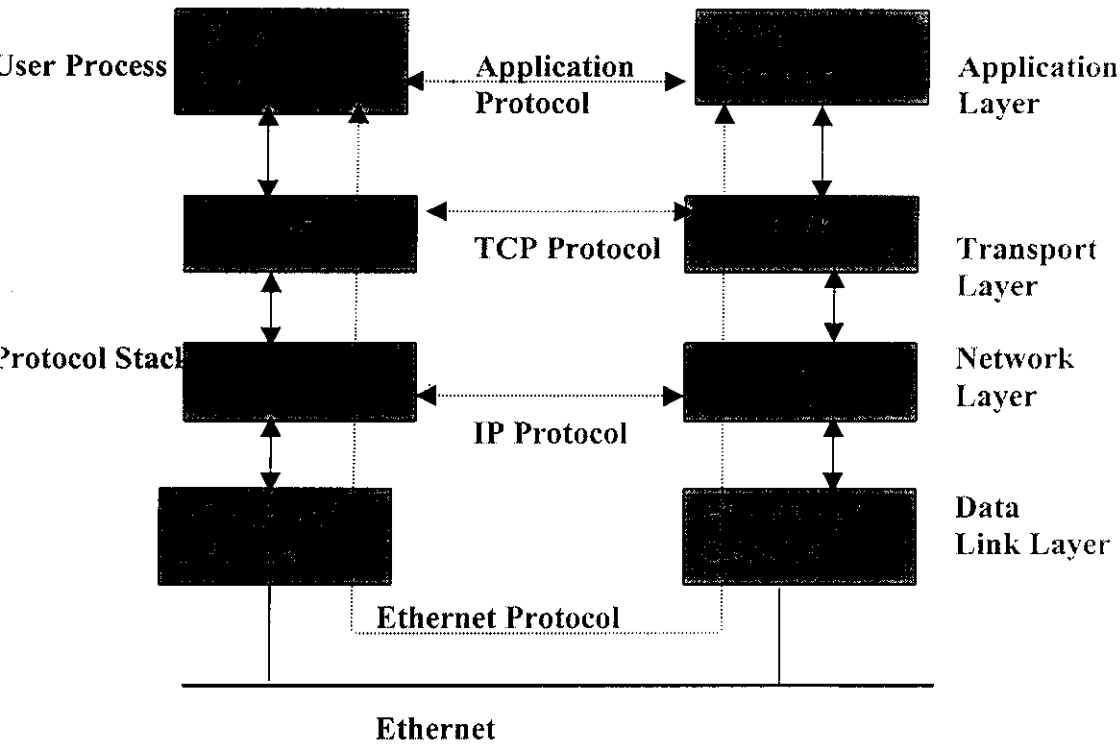


Figure 2

SNMP is an application level protocol that is a part of the UDP/IP protocol suite. It is intended to operate over the UDP. As shown in figure 2 the manager process achieves network management by using SNMP, which is implemented on the top of UDP, IP and the relevant network-dependent protocols (for example, Ethernet, FDDI, and X.25). Each agent also has SNMP, UDP, and IP. An agent process interprets the SNMP messages and controls the agent's MIB.From the management station, four types of SNMP commands are issued on behalf of a management application. They are,

- GetRequest:

  This enables the management station to retrieve the value of the objects at the agent.

- GetNextRequest:

  This enables the management station to retrieve the value of the next objects at the agent.

- SetRequest:

  This enables the management station to set the value of the objects at the agent.

- SetNextRequest:

  This enables the management station to set the value of the next objects at the agent.

All these messages are acknowledged by the agent in the form of a GetResponse message, which is passed up to the management application. In addition, an agent may issue trap message in response to an event that affects the MIB and the underlying managed resources.

- Trap:

  This enables the agent to notify the management station of significant events.

2.2. MIB:

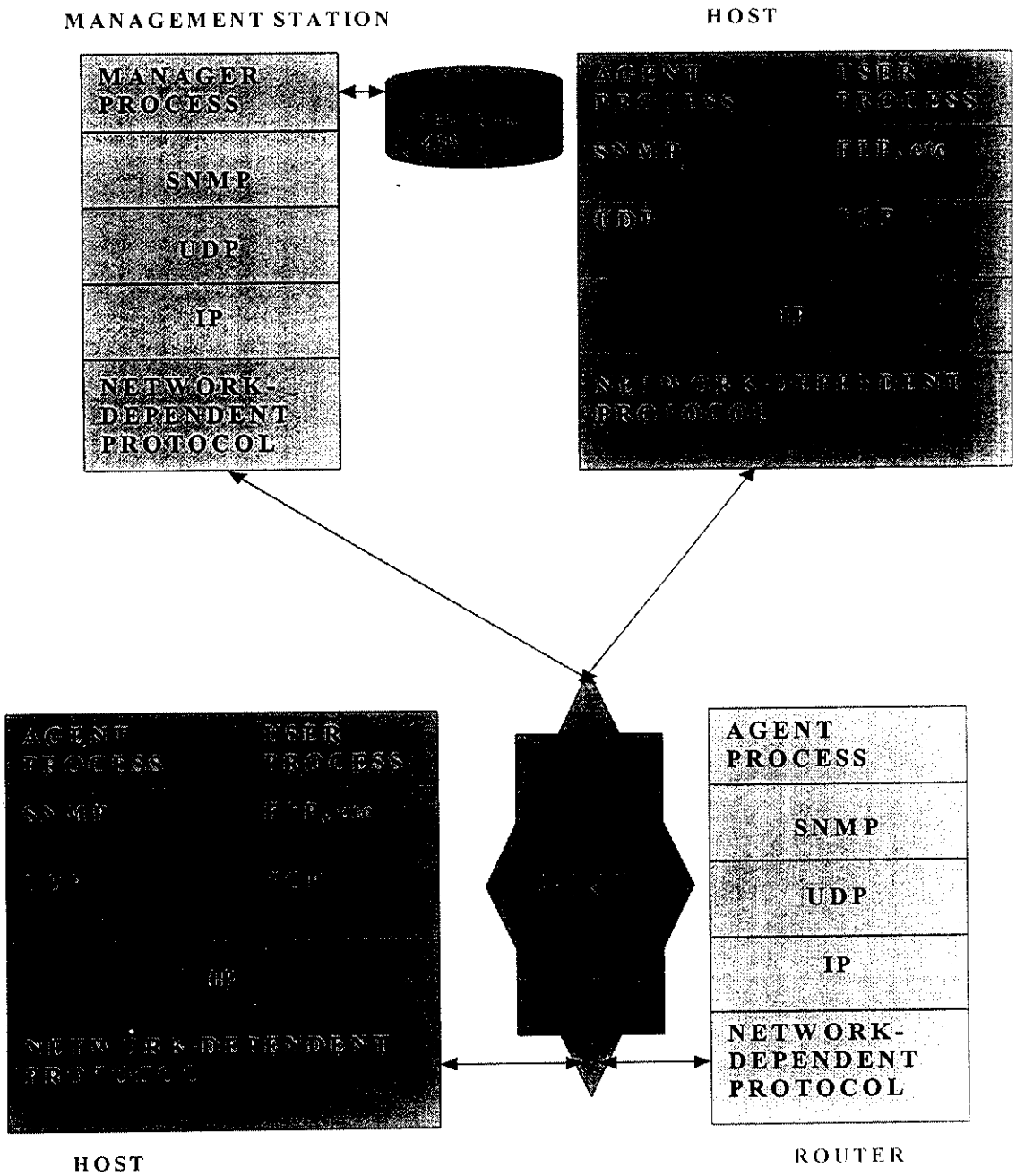


Figure 3

MIB is abbreviated as Management Information Base. This is where he object variables are stored as shown in figure 3. The MIB can be invoked by means of an object identifier. The management tree identifies the managed information and the data. The management tree is a hierarchical structure starting from the root as shown in figure 4. It then branches of into branches and leaves. This is represented numerically and alphabetically. The numerical code is machine-readable and the alphabetical display is for humans to read.

The top level SNMP branch begins with the ISO "internet" directory. This contains four main branches.

Currently the "directory" SNMP branch contains no directory objects and remains reserved for future applications.

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 equipment vendors.

The "experimental" and "directory" SNMP branches, also defined within the "internet" root directory, are usually devoid of any meaningful data or objects.

MIB-I contains 8 groups with 100 objects. But this proved to be less in the growing network population. So the MIB-II was used, which consists of 11 group with 180 objects.

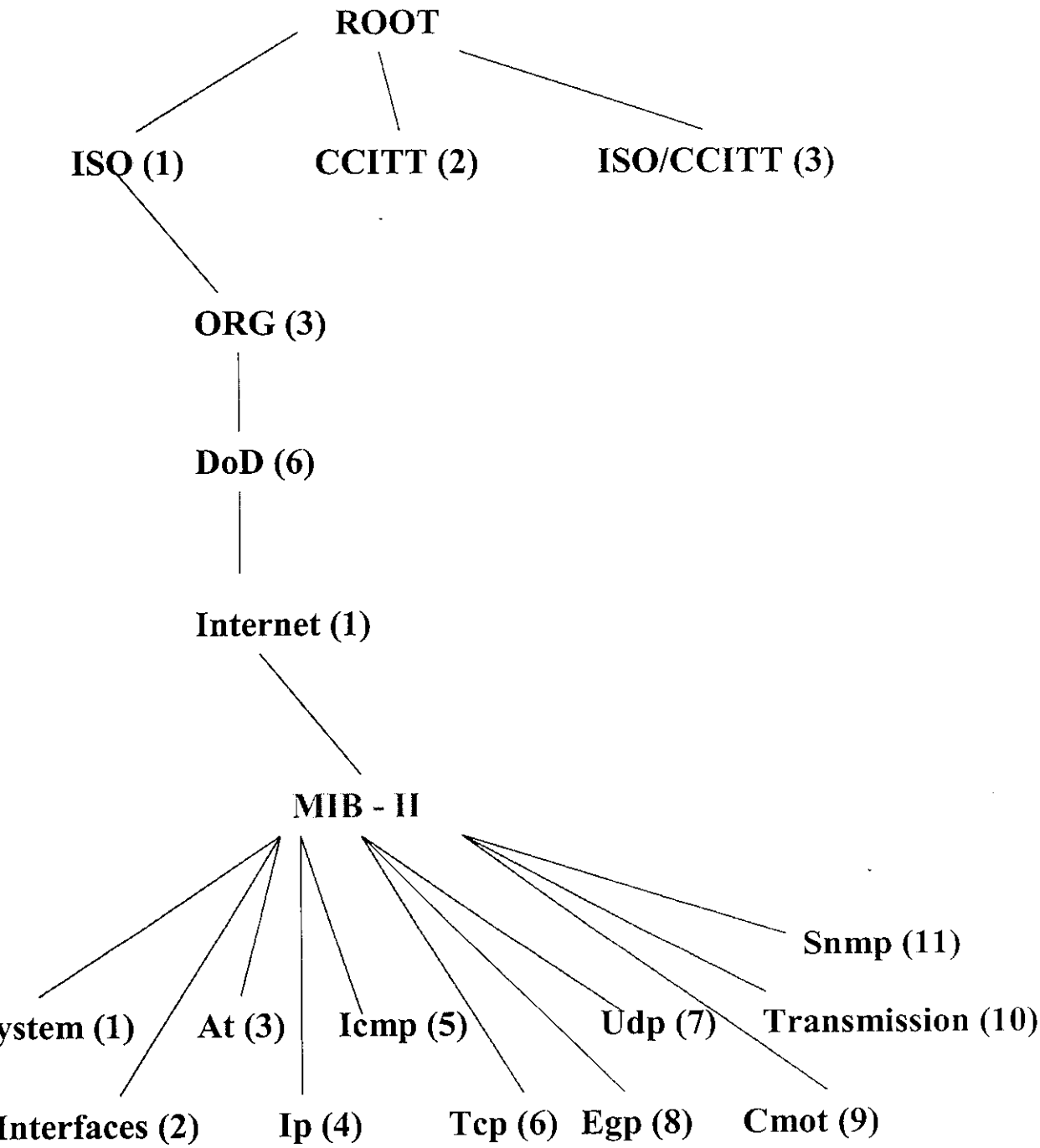

ROOT
ISO (1)
CCITT (2)
ISO/CCITT (3)
ORG (3)
DoD (6)
Internet (1)
MIB - II
Snmp (11)
ystem (1)
At (3)
Icmp (5)
Udp (7)
Transmission (10)
Interfaces (2)
Ip (4)
Tcp (6)
Egp (8)
Cmot (9)

Figure 4

OBJECT SYNTAX:

Every object within an SNMP MIB is defined in a formal way. The ASN.1 notation is used to define each individual object and also to define the entire MIB structure. They are,

❖ **Universal Types:**

The universal class of ASN.1 consists of application independent data types that are of general use. Within the UNIVERSAL class only the following data types is permitted to be used to define MIB objects:

- Integer (UNIVERSAL 2)
- Octetstring (UNIVERSAL 4)
- Null (UNIVERSAL 5)
- Object Identifier (UNIVERSAL 6)
- Sequence, Sequence of (UNIVERSAL 16)

The first four types are primitive types. The last one is of constructor type.

❖ **Application-Wide Types:**

The application class of ASN.1 consists of data types that are relevant to a particular application. The following types are defined:

- Networkaddress
- IpAddress
- Counter
- Gauge
- Timeticks
- Opaque

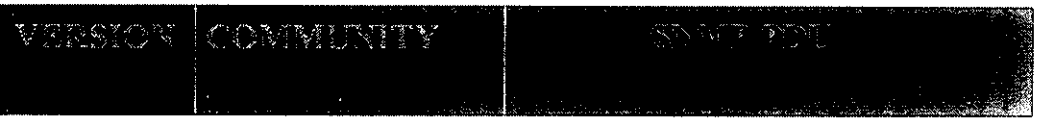
2.3. UDP:

SNMP requires the use of a transport service for the delivery of SNMP messages. Most implementations of SNMP use the User Datagram Protocol (UDP).

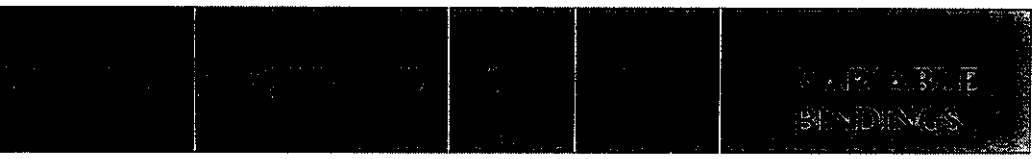
UDP is a connection less transport service. UDP segments are transmitted in IP datagrams. The UDP header includes source and destination port fields, enabling application level protocols such as SNMP to address each other. It also includes an optional checksum that covers the UDP header and user data. If there is a checksum violation, the UDP segment is discarded. No other services are added to IP. Two ports have being assigned to SNMP. Agents listen for the incoming commands through the port 161. Management stations listen for the incoming traps on port 162.

2.4. SNMP FORMATS:

NMP MESSAGE:

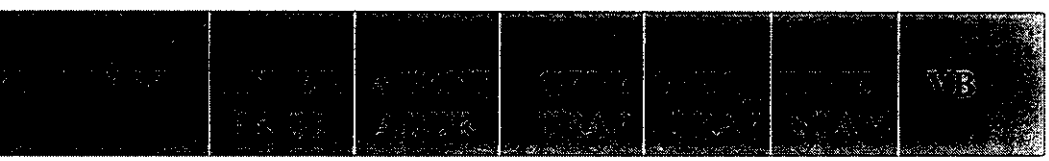


ETREQUEST PDU, GETNEXTREQUEST PDU, SETREQUEST PDU:



ETRESPONSE PDU:

RAP PDU:



ARIABLE BINDINGS:

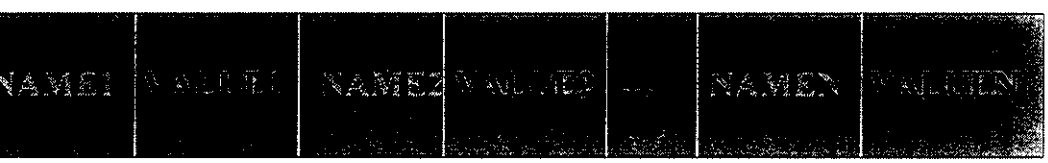


Figure 5

With SNMP, information is exchanged between a management tation and an agent in the form of an SNMP message. Each message, as nown in figure 5, includes a version number indicating the version of NMP, a community name to be used for this exchange, and one of the five ypes of protocol data units. The GetRequest, GetNextRequest, and etRequest PDUs have the same format as the GetResponse PDU, with the rror-status and the error-index fields always set to 0.

NMP message format is

- Version :

  SNMP version as described in RFC1157

- Community:

  This acts as a password to authenticate the SNMP message.

- Request_id:

  This distinguishes among outstanding requests by providing each request with a unique ID.

- Error_status:

  This is used to indicate that an exception has occurred while processing a request.

- Error_index:

  When the error_status is nonzero, this may provide additional information by indicating which variable in a list caused this exception.

- Variablebinding:

  A list of variable names and its corresponding value.

- Enterprise:

This describes the type of object generating the trap.

- Agent_addr:

  This describes the address of the object generating the trap.

- Generic_trap:

  This trap describes a one of the predefined trap types.

- Specific_trap:

  This trap describes a code that indicates more specifically the nature of the trap.

- Time_stamp:

  The time between the last re-initialization of the network entity that issued the trap and the generation of the trap.

3.CONSTRUCTION:

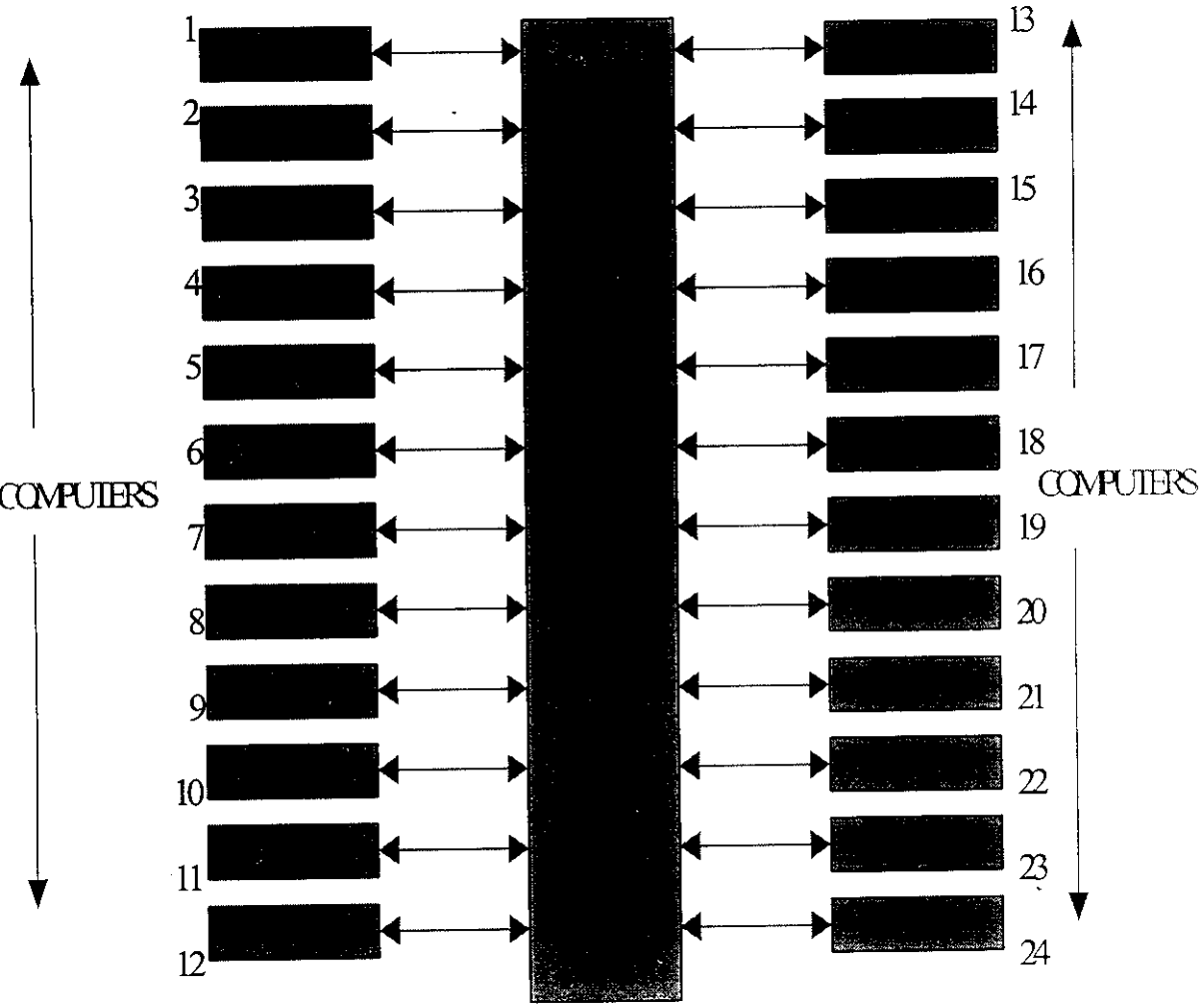


Figure 6

A common switch hub was used to establish connection for 24 network elements as shown in figure 6. The switch acts as a fore end of an intelligent device.The cables used for networking these networking elements were copper coaxial cables. The cables were laid for 500 meters for networking different network elements. The switch established connection between each computer. The switch hub had 24 ports to connect each computer.

An Ethernet type of networking technology was done in the physical layer. Star type of topology was used to network the different computers. The one end of the cable connects to the end if the Network Interface Card or the adapter card of the computer and the other end to the port of the switch.

The manager software and the agent software were installed in all the computers connected to the network. All the workstations had the SNMP, UDP, IP, and ARP/RARP protocols. So the design was also done according to the IEEE standards.

The whole set up was done for a local area network with star topology. A point-to-point connection was established among the different computers. Since it establishes a star topology even though if one machine or a cable fails will not affect the rest of the computers in the networks. But the disadvantage of the star topology is that if the switched hub fails then the whole network collapse.

3.1 FLOW CHART:

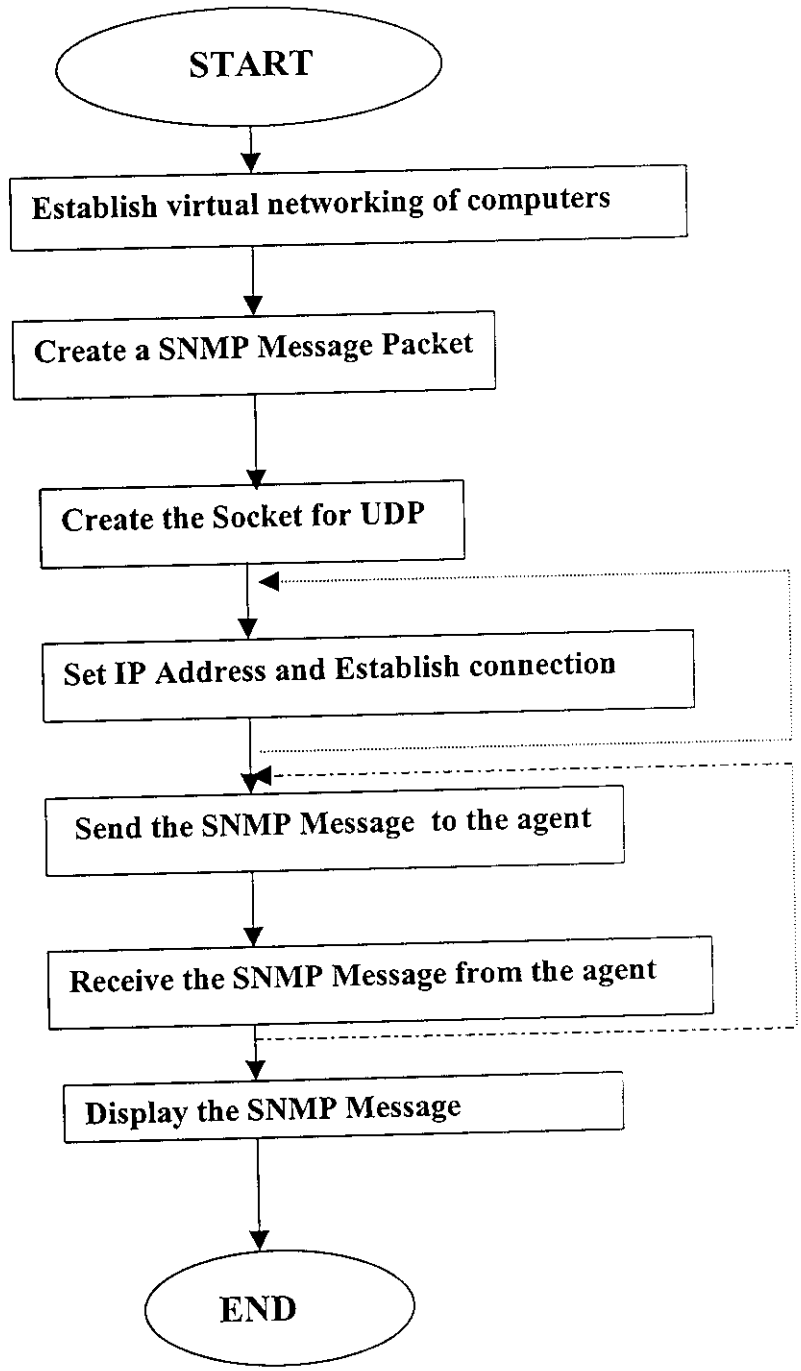


Figure 7

STABLISHING TEMPORARY NETWORKING:

The number of computers, which has to be networked, as mentioned. The IP addresses of the computers were ientioned and they are temporarily networked virtually as shown n figure 7. If any computer, which has, been entered for etworking more than once was indicated as an error message nd terminated; even if the computers exceeded the number of omputers to be networked then an error message was displayed nd the process terminated.

NMP MESSAGE CREATION:

The message to be sent was designed with the SNMP rotocol. This protocol has a header, data and a trailer part. The leader part consists of the source and the destination address. he complete design of the message was been described in napter 2. The packets in the SNMP can vary according to the ze of the message. The message consisted of the version umber.

This version number was given as 0, which indicates ie version of the SNMP as one. This version number was ndicated in the integer format. The integer was designed and oded as a separate class. So the integer class was called when ie version number was to be designed. The integer class had the lass, format, length, and data field specified in it. The version

number was taken as a separate field as version inside the class CTarget.

The CTarget was the class where the version field and the community field were set. The community was the next field in the packet. This acts as a password for establishing the connection. The community field (public) was six characters long. The community name was indicated in the octet string type. The octet string was designed and coded as a separate class called Octetstr. So when the community field was mentioned the octet string class was called. This octet string class had the class, format, length, and data field specified in it. The version number and the community name along with the destination address formed the header.

Then was the Protocol Data Unit, which is commonly referred as the message part. This was where the SNMP commands were being set. There were four commands set namely Get, GetNext, Set, and SetNext. These commands were given standard specified numbers set by the IEEE standard as 0 for Get, 1 for GetNext, 3 for Set and 4 for SetNext. So in the PDU the first field was Pdu type. This pdu type specified the type of commands.

Then the next field was the request identifier, which was represented as request id. This specifies the number of the packet for identification. Then the next fields were the Error status and the error index. The error status specified the type of error. The error index specified at which place the error had

occurred. The error index and the error status were the integer type so the integer class was called there. The pdu types, request ID, error status, error index were put in a class Pdu.

The variable binding section which is also a part of the pdu format. This variable binding consisted of the object and the value field. The object identifier specified the object. So the Object identifier class was called. The object identifier finds the object in the MIB tree of the MIB database. This object identifier class consisted of class, format, length and the data field.

The value of the object can be octet string, integer, time ticks etc depending upon the type of object specified. So octet string, integer, timeticks classes etc were called. This variable binding value field was the place to be filled by the agent and returned to the manager. Rest all the parts of the message were filled by the manager. The object and the value field are put in the variable binding class.

The version number, community name, pdu data type, and variable binding together combine to form a message. So all the separate classes were concatenated together by an operator overloader. This gave the full design of the message packet. Now the packet is ready for sending.

3.2 SYSTEM FLOW:

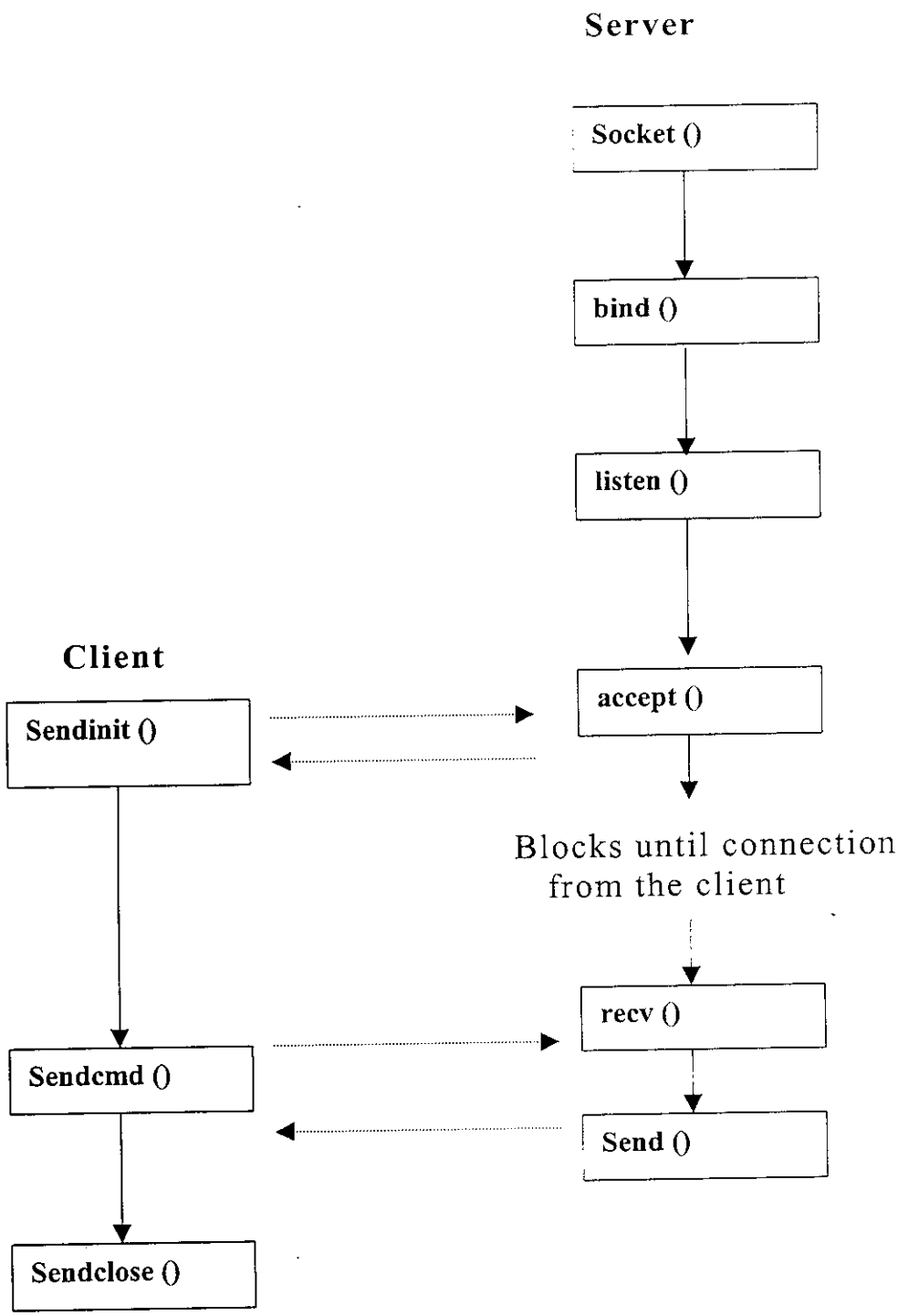


Figure 8

SOCKET CREATION:

Before sending a message, a socket has to be created and a connection has to be established as shown in figure 8. A socket was created in the manager computer, which was intented to send the message to agent. The socket is a communication end point between the two layers.

The socket here was created using the function

sendinit (host).

The host is where the IP address was passed. This socket if created, returned a value 0, else a negative value. If it had returned a negative value it indicates some error has occurred in creation of the socket. So again it has to be newly created until the socket gets created.

ESTABLISHING CONNECTION:

After the socket has being created, it has to establish a connection. But establishing a connection is optional since it is a connectionless style of protocol. The sendinit (host) function also established the connection between the computers. The IP Address was given for establishing connection. The family name, port numbers are internally designed default.

SENDING AND RECEIVING SNMP MESSAGE:

The message, which was already created, was sent through the function

sentcmd (sendbuf, Strlen (sendbuf), flag).

This sends the message along the socket, which was created. The size of the message to be sent was checked. The destination address along with its size was also mentioned. This was done because UDP is a connectionless style of transport protocol. So with each packet there is a need for the destination address. The message was sent through a port 161 of the UDP.

After the message was sent the client waited for receiving the message. The message, which was sent, was checked for error if any. The community name should match with the agent's community name. The message was sent until a response from the agent was received. The sending and the receiving of the message were done by the same function. The function returned a value 0 and indicated that the message has been received. If it was other than the 0 then it indicates that an error had occurred.

When the message was received it was split into the three portions. The front portion of the message consists of the version and the community. The middle portion of the message consists of the PDU data type, which is also separated. These two portions are not modified or filled in the agent. The variable binding part is the third portion of the message where the agent fills and sends back the message. The data field of the variable portion was only altered and filled.

DISPLAY MESSAGE:

The variable binding portion of the message, which was received from the agent, was put forth on the screen for the display. This is in the the human understandable form. The message, which was received, varies for different objects. So the size of the fields was made to vary flexibly. The class Snmp gave this.

SOCKET CLOSE:

After all the messages, which were, need to be sent and received are done the socket was then closed. This disconnected the connection.

3.3 OBJECTS:

There were seven object types used in this project. They can be extended. These objects were SysDescr, SysObjectID, SysUpTime, SysContact, SysName, SysLocation, and SysServices. These objects described about the information of the system. These were grouped under a common object group called system.

In order to reach these objects the MIB-II tree were used. The MIB-II consisted of a nearly 11 groups with 180 objects. But the project was done for 1 group with 7 objects because extension of the rest of the objects was relatively easy with the same set of codes. Each object in the group was reached through a path called the object identifier. The different objects are described briefly below.

SysDescr:

This was the first object of the system group. This gives the description of the system. The object identifier for this in human readable is

iso.org.dod.internet.management.mib-II.system.sysdescr

and in numeric value is 1.3.6.1.2.1.1.1.0

The numeric value is what the machine could read. This object was obtained with the command as Get SysDescr. So the description about the system was displayed on the screen. The Set command couldnot be used here because it was only a readable value. It was not possible to write over it. So when a set

command was given with this object, the message displaying that it was a readable object was displayed.

SysObjectID:

This was the second object of the system group. This described the path of the object in the MIB tree. The object identifier for this object was (in human readable) as

iso.org.dod.internet.management.mib-II.system.sysobjectid

and (in the numeric value) as 1.3.6.1.2.1.1.2.0

This object was obtained with the command as Get SysObjectID. So the complete object identity about the system was displayed on the screen. The Set command cannot be used here because it has access as only as read only. It was not possible to write over it. So when a set command was given with this object, the message displaying that it was a readable object, was displayed.

SysUpTime:

This was the third object of the system group. This described the last updated time of the object of the system. The object identifier for this object was (in human readable) as

iso.org.dod.internet.management.mib-II.system.sysuptime

and (in the numeric value) as 1.3.6.1.2.1.1.3.0

This object was obtained with the command as Get SysUpTime. So the time at which the system was updated was displayed on the screen. The Set command cannot be used here because it has access as only as read only. It was not possible to write over it.

So when a set command was given with this object, the message displaying that it was a readable object, was displayed.

SysContact:

This was the fourth object of the system group. This described the person to be contacted together with how to be contacted for the information about the system. The object identifier for this object was (in human readable) as
iso.org.dod.internet.management.mib-II.system.syscontact
and (in the numeric value) as 1.3.6.1.2.1.1.4.0
This object was obtained with the command as Get SysContact. So the person to be contacted and the details of contacting the person was displayed on the screen. The Set command can be used here because it has access as read-write. It was possible to write over it. So changes were made in the system it was modified. Again it was retrieved means of Get SysContact command, it displayed the latest modified details.

SysName:

This was the fifth object of the system group. This described the administratively assigned name of the system. The object identifier for this object was (in human readable) as
iso.org.dod.internet.management.mib-II.system.sysname
and (in the numeric value) as 1.3.6.1.2.1.1.5.0
This object was obtained with the command as Get SysName. So the name of the system, which was given by the system administrator was displayed on the screen. The Set command

can be used here because it has access as read-write. It was possible to write over it. So the name of the system was changed. Again it was retrieved means of the Get SysName command; it displayed the latest modified details.

SysLocation:

This was the sixth object of the system group. This described the area of location of the system in the LAN. The object identifier for this object was (in human readable) as

iso.org.dod.internet.management.mib-II.system.syslocation

and (in the numeric value) as 1.3.6.1.2.1.1.6.0

This object was obtained with the command as Get SysLocation. So the area were the system was located was displayed on the screen. The Set command could be used here because it had access as read-write. It was possible to write over it. Some alteration of the systems area of location was done. Again it was retrieved means of Set SysLocation command; it displayed the latest modified details.

SysServices:

This was the seventh object of the system group. This described the set of services, which has being done in the system. The object identifier for this object was (in human readable) as

iso.org.dod.internet.management.mib-II.system.sysservices

and (in the numeric value) as 1.3.6.1.2.1.1.7.0

This object was obtained with the command as Get SysServices. So the services made for the system was displayed on the screen. The Set command cannot be used here because it has access as read. It was not possible to write over it. So when a set command was given with this object, the message displaying that it was a readable object, was displayed.

The GetNext command can be used along with an object name Which described the next object in the system group. The SetNext command modifies the next command in the system group.

LAUNCH VEHICLE USAGE:

The purpose of the project was for the launch vehicles onboard computer. There are three sets of onboard computers used in the launch vehicle. These computers are used to monitor and manage the launch vehicle. There are many devices connected to the onboard computer. These devices may be at different locations and made by different manufacturers.

Each device may have different names and different service centers. There may be a lot of descriptions for each device. So it is very difficult to manage them manually or without network management software. So each device details are collected by the agent software and made ready for the managerial software. The managerial software manages all the devices networked to the agent software. The project designed here describes the complete details of the onboard computer

systems information, i.e., the name, place of location, time at which the system was updated etc.

The project is extensible. So it has the capability of supporting more objects. Therefore the project is to be used for the real time system. This is the greatest advantage of the project. This project also supports non-real time system if the proper sockets are installed.

4. PROGRAM DESIGN:

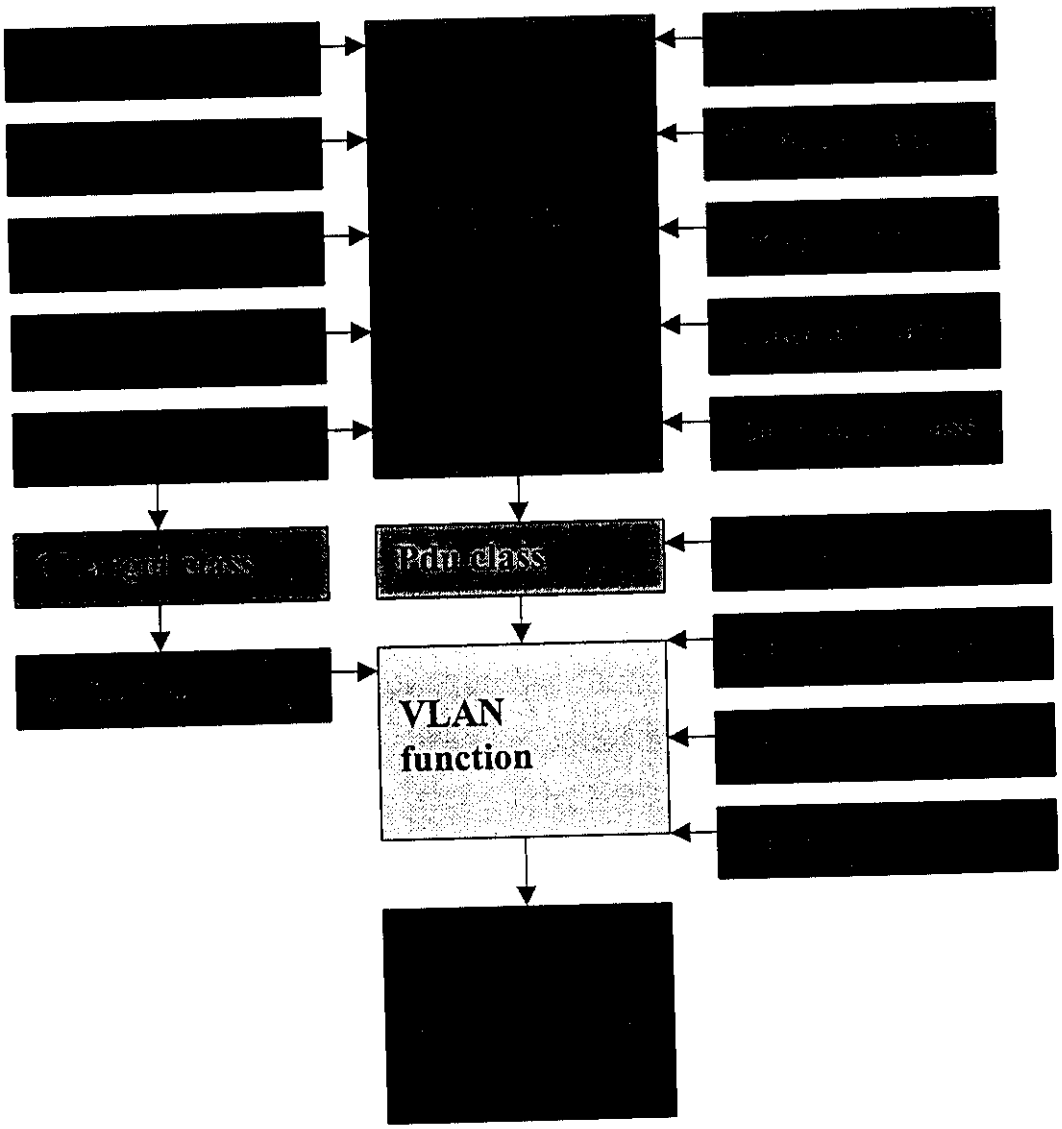


Figure 9

CLASSES:

They are the building blocks for eligible programming. They use the principles of OOPS. Ten base classes and two derived classes were used. Other two classes were not derived or base class. Altogether there were fourteen classes used in this project.

The base classes were
INTEGER, Oid, Octetstr, IpAddress, Null,
Sequence, Opaque, Counter, Gauge, and TimeTicks.

The derived classes used were Pdu, Vb class.
The other two classes were the Ctarget and Snmp.
These classes are described in brief below.

INTEGER Class:

INTEGER class was used for defining variables as integer. This integer was different from that of the normal integer data type. So this class was made as a user defined data type. Therefore variables could be declared as INTEGER. While receiving the data's from the agent too, the data's were declared as an INTEGER where this data type has to be applied. The length of the INTEGER varies, so the constructors were designed flexibly according to the varying length.

Oid Class:

Oid class was used for defining variables as object identifier. So this class was made as a user defined data type. Therefore variables could be declared as object identifier. While receiving the data's from the agent, the data's were declared as an object identifier for the display of the data variable. The length of the object identifier varies, so the constructors were designed flexibly according to the varying length.

Octetstr Class:

Octetstr class was used for defining variables as Octet string. It could be displaying it as character. But this character was different from the normal character data type. So this class was made as a user defined data type. Therefore variables could be declared as Octet string. While receiving the data's from the agent too, the data's were declared as an Octet string for the display of the data variable. The length of the Octet string varies, so the constructors were designed flexibly according to the varying length.

TimeTicks Class:

TimeTicks class was used for defining variables as the time settings. It could be displaying it in hours, minutes, and seconds. But this time data type was different from the normal time. So this class was made as a user defined data type. While receiving the data's from the agent too, the data's were declared as a TimeTicks for the display of the data variable. The length of the TimeTicks was constant, but the constructors were designed flexibly accordingly in order for the further development

IpAddress Class:

IpAddress class was used for defining variables as IP Address. It could be displaying it as unsigned character of the fixed length. But this character was different from the normal character data type. So this class was made as a user defined data type. Therefore variables could be declared as IpAddress. While receiving the data's from the agent too, the data's were declared as an IpAddress for the display of the data variable. This IpAddress class used for declaring the host name and foreign name of the network element. The length of the IpAddress does not vary, so the constructors were designed with out flexibility.

The first five base classes were used for the project. The rest five were designed but not used in this project. But they are to be implemented in the launch vehicle.There were certain keywords, which were used in the classes. They are

- Protected:

  In these classes there were some variables, which was protected. These variables were protected because it had to be derived by the rest of the classes.

- Public:

  Any thing declared under the public keyword could be used anywhere in the program. So the member functions and the constructors were declared here.

Constructors were used in these classes. Constructors were used to initialize a variable value. All the base classes used three constructors. They are

- Constructor with no argument.
- Constructor with a single argument.
- Constructor with the same class name as its argument

Constructor with no arguments:

These constructors carried no argument and the values here were initialized to zero or some constant value.

Constructor with a single argument:

The argument was either an unsigned character or a character or an integer type. It received an array of variable values. So in the constructor the argument was pointed to the first value by means of a pointer.

Constructor with same class variable:

Here the argument was received as the same data type, which was the class name. The received variable was checked for its maximum size. If the length was less than the maximum size then the variable was copied into another variable. If the variable was greater than the maximum size then an error warning was printed out. Here in the project this constructor was called only for IpAddress, where the Ipaddress was declared by its same name. The constructor does not return any value, so a member function was included in these classes because this gets the input and returns the value.

Vb class:

The variable binding class consisted of the object identifier and the value portion. The value can be any type of the base class. So all the base class was derived in this vb class. So this class became the derived class.

The variable binding class put the entire field of the vb class inside the buffer and made ready to be concatenated to the pdu class.

Pdu Class:

The pdu class defined the command field, the request identifier number, the error status, and the error index field. It puts every field in a buffer and made it into a single message. It derived the fields from the variable binding class. So the vb class became the base class of the pdu class and the pdu class became the derived class. The operator overload function inside this class concatenated the variable binding and the pdu format.

CTarget class:

The CTarget class defined the version number, and the community name. It also created the socket for the Snmp class and established connection to the remote network element. It checked the community name with the agent and if it matched only it could establish a connection. It puts every field in a single buffer.

Snmp class:

The CTarget buffer was concatenated with the pdu class buffer in the Snmp class. This was a main class, which checked the message packet. With the socket created by the CTarget it established the connection. It indicated that a connection had being established. After establishing a connection the message was send. The message, which was send, was verified at the agent and the response was send back. It kept sending the message until a response was obtained. The response, which the agent had sent, was received back by the Snmp class. It was then checked in the Snmp

class and the vb class data's were separated from the rest of the packets. The separated value was then displayed on the screen as output.

FUNCTIONS:

Four functions were called in the vlan () function. The four functions were get_function (), getnext_function (), set_function (), setnext_function (). The vlan () function was called in the main function of the project. All these functions are given in brief below.

get_function ():

This function was used to get the object and display it on the screen. This is a main part were all the classes are called and the program was built. Here the status of the socket creation and message reception was checked. Since constructors do not have the return value this was done here.

After reception of the message format the data field was separated from the header field and the result was displayed on the screen. It also checked whether the message was passed through the same computer that was networked before. If it was passed through the computer that was not networked before then an error was displayed and the process terminated. This is called virtual networking of the local area computer.

getnext_function ():

This function was used to get the next object and display it on the screen. This is generally used to check the next object in a table sequence. But here it was used for the next object to be displayed. Here also the status of the socket creation and message reception was checked. Since constructors do not have the return value this was done here.

After reception of the message format the data field was separated from the header field and the result was displayed on the screen. It also checked whether the message was passed through the same computer that was networked before. If it was passed through the computer that was not networked before then an error was displayed and the process terminated. All these functions were done by means of calling the get function into the program.

set_function ():

This function was used to set the object, which are read and write access. This was a main part were all the classes are called and the program was built. Here the status of the socket creation and message reception was checked as before. The additional data field, which has to be added, was attached to the variable binding and then passed as a message format.

But if the object had access as read only, then it displayed an error message. So no modification could be done there. It also checked whether the message was passed through the same computer that was networked before. If it was passed through the computer that was not networked before then an error was displayed and the process terminated. This does not display on the screen the data field that was modified. So it was retrieved back by the get function and displayed on the screen.

setnext_function ():

This function was used to set the next object. This is generally used to set the next object in a table sequence. But here it was used for the next object to be set. The additional data field, which has to be added, was attached to the variable binding and then

passed as a message format. But if the object had access as read only, then it displayed an error message. So no modification could be done there.

It also checked whether the message was passed through the same computer that was networked before. If it was passed through the computer that was not networked before then an error was displayed and the process terminated. This does not display on the screen the data field that was modified. So it was retrieved back by the get function and displayed on the screen. All these functions were done by means of calling the set function into the program.

vlan ():

This was the function, which called all the other functions. Here the computers were first temporarily networked virtually. It checked if a computer had being entered twice for virtual networking. If so then an error message was displayed. A maximum of 10 computers was networked. But it can also be extended depending up on the switched hub in the star topology. Here the switch had 24 ports but only ten computers were connected to it.

starline () :

The starline () function was used only for graphic works such as drawing the line. This was done in three types. It can draw line default without passing an argument. It can draw line according to the user wish in passing an argument. It drew the line from the first pixel to the end of the rows last pixel in a straight line. Actually it has nothing to do with the project but only for clarity and appearance.

Function of Main program:

The main function is the starting portion of the program. The C++ compiler starts its execution from the main function of the program. The main program called the vlan () function. Then the vlan function was executed fully. One timer was set at the start of main function. Another timer was set at the end of the main function. The difference in the time of execution was calculated at the end of the function. So the time of execution was also calculated in this main function.

5. CONCLUSION

The main advantage of this project is that it supports real time system. Since it virtually networks some of the network elements, communication is possible among these networked elements. Even though this project was not designed for all the 180 objects of the MIB-II,adding these objects can easily extend it.

Its design is simple and is easy for a user to program the variables. The SNMP protocol used here is easily obtainable through Internet and other media at free of cost. So the updating of the project is quite easy.

For this project the design of the SNMP message packet was done. This gave the foundation for full development of the project for future modification. Once the structure of the packet was known it is easy to develop further details on the project for real time system.

Some limitations are also there in this project. It is not well suited for retrieving large volume of data. But extending the size of the array in the message packet can rectify it. It does not support manager-to-manager communications. This can be rectified by means of the RMON protocol or the SNMPv3 protocol.

BIBLIOGRAPHY

[1]. Comer E. Douglas and Steven L.David,
Internetworking with TCP/IP Volume III, II & I,
Prentice Hall International, Inc, 1995

[2]. Steven Richard, *UNIX Network Programming Volume II & I.*
McGraw –Hill International Edition, 1991

[3]. Hein Mathias & Griffiths David, *SNMP Version III, II & I,*
Macmillan Computer Publication, 1993

[4]. Tannenbeum S.Andrew, *Computer Networks,*
McGraw-Hill International Edition, 1990

[5]. Stallings William, *SNMP – III Edition,*
Prentice Hall International, Inc, 1993

[6]. Wollongong Group,
Pathway from Wollongong – support services, 1999

[7]. Lafore Robert,*Borland C++,*
Galgotia Publication Pvt. Ltd.,1990

[8]. *www.Snmp++.com*
Describes the simple network management protocols.

[9]. *www.udp.com*
Describes the transport protocols.

[10]. www.oakland.edu
Describes the RFCs details.

```
ys Date: 24 - 11 - 2000                          The day of the Week is:
time before compilation is:                          10:55:15.33
*****************************************************************************
                    VLAN CONFIGURATION USING SNMP
*****************************************************************************
r the number of network elements to be connected: 5
ct the network elements to be connected
72.20.3.6   2:  172.20.3.200        3:  172.20.3.3    4:  172.20.3.1
72.20.3.10  6:  172.20.3.7          7:  172.20.3.12   8:  172.20.3.18
72.20.3.14  10:  172.20.3.15
-----------------------------------------------------------------------------
 the Network Elements: 1 2 3 5 7
-----------------------------------------------------------------------------
Commands:- 0.Get   1.GetNext       3.Set   4.SetNext
t Names:-  SysDescr        SysObjectID     SysUpTime       SysContact
           SysName         SysLocation     SysServices
-----------------------------------------------------------------------------
 the SNMP command & the Object Name: Get SysName
 the Computer to be communicated: 5
rror
et 60 has been created for host: 10
gent/server returned Value is:
CSD
-----------------------------------------------------------------------------
u want to continue(y or n):
```

```
-----------------------------------------------------------------------------
ommands:- 0.Get   1.GetNext       3.Set   4.SetNext
 Names:-  SysDescr        SysObjectID     SysUpTime       SysContact
          SysName         SysLocation     SysServices
-----------------------------------------------------------------------------
the SNMP command & the Object Name: Get SysDescr
the Computer to be communicated: 5
ror
t 60 has been created for host: 10
gent/server returned Value is:
are: x86 Family 5 Model 2Steppin 12 AT/AT compatible_software: Windows
a 4.0 ( Build Number: 1381 Uniprocessor Free )
-----------------------------------------------------------------------------
 want to continue(y or n):
```

```
 Commands:- 0.Get   1.GetNext       3.Set   4.SetNext
ct Names:-  SysDescr        SysObjectID     SysUpTime       SysContact
            SysName         SysLocation     SysServices
-------------------------------------------------------------------------
r the SNMP command & the Object Name: Get SysObjectID
r the Computer to be communicated: 5
Error
ket 60 has been created for host: 10
 Agent/server returned Value is:
.6.1.2.1.1.2.0
-------------------------------------------------------------------------
you want to continue(y or n): y
```

```
-------------------------------------------------------------------------
 Commands:- 0.Get   1.GetNext       3.Set   4.SetNext
ct Names:-  SysDescr        SysObjectID     SysUpTime       SysContact
            SysName         SysLocation     SysServices
-------------------------------------------------------------------------
r the SNMP command & the Object Name: Get SysUpTime
r the Computer to be communicated: 5
Error
ket 60 has been created for host: 10
 Agent/server returned Value is:
2:10.10
-------------------------------------------------------------------------
you want to continue(y or n): y
```

```
-------------------------------------------------------------------------
P Commands:- 0.Get   1.GetNext       3.Set   4.SetNext
ect Names:-  SysDescr        SysObjectID     SysUpTime       SysContact
             SysName         SysLocation     SysServices
-------------------------------------------------------------------------
er the SNMP command & the Object Name: Get SysContact
er the Computer to be communicated: 5
 Error
cket 60 has been created for host: 10
e Agent/server returned Value is:
il FCSD SC/Egr
-------------------------------------------------------------------------
 you want to continue(y or n): y
```

```
ommands:- 0.Get   1.GetNext      3.Set   4.SetNext
 Names:-  SysDescr        SysObjectID     SysUpTime       SysContact
          SysName         SysLocation     SysServices
---------------------------------------------------------------------
the SNMP command & the Object Name: Get SysLocation
the Computer to be communicated: 5
ror
t 60 has been created for host: 10
gent/server returned Value is:
left corner
---------------------------------------------------------------------
u want to continue(y or n): y
```

```
ommands:- 0.Get   1.GetNext      3.Set   4.SetNext
 Names:-  SysDescr        SysObjectID     SysUpTime       SysContact
          SysName         SysLocation     SysServices
---------------------------------------------------------------------
the SNMP command & the Object Name: Get SysServices
the Computer to be communicated: 5
ror
t 60 has been created for host: 10
gent/server returned Value is:
 floor FCSD room no: 110
---------------------------------------------------------------------
u want to continue(y or n): y
```

```
Commands:- 0.Get   1.GetNext      3.Set   4.SetNext
t Names:-  SysDescr        SysObjectID     SysUpTime       SysContact
           SysName         SysLocation     SysServices
-------------------------------------------------------------------
 the SNMP command & the Object Name: GetNext SysDescr
 the Computer to be communicated: 5
rror
et 60 has been created for host: 10
Agent/server returned Value is:
6.1.2.1.1.2.0
-------------------------------------------------------------------
ou want to continue(y or n): y
```

```
-------------------------------------------------------------------
Commands:- 0.Get   1.GetNext      3.Set   4.SetNext
t Names:-  SysDescr        SysObjectID     SysUpTime       SysContact
           SysName         SysLocation     SysServices
-------------------------------------------------------------------
 the SNMP command & the Object Name: GetNext SysObjectID
 the Computer to be communicated: 5
rror
et 60 has been created for host: 10
Agent/server returned Value is:
:10.10
-------------------------------------------------------------------
ou want to continue(y or n): y
```

```
-------------------------------------------------------------------
Commands:- 0.Get   1.GetNext      3.Set   4.SetNext
t Names:-  SysDescr        SysObjectID     SysUpTime       SysContact
           SysName         SysLocation     SysServices
-------------------------------------------------------------------
 the SNMP command & the Object Name: GetNext SysUpTime
 the Computer to be communicated: 5
rror
et 60 has been created for host: 10
Agent/server returned Value is:
 FCSD SC/Egr
-------------------------------------------------------------------
ou want to continue(y or n): y
```

```
Commands:- 0.Get   1.GetNext      3.Set   4.SetNext
t Names:-  SysDescr        SysObjectID     SysUpTime       SysContact
           SysName         SysLocation     SysServices
-------------------------------------------------------------------
 the SNMP command & the Object Name: GetNext SysName
 the Computer to be communicated: 5
rror
et 60 has been created for host: 10
Agent/server returned Value is:
 left corner
-------------------------------------------------------------------
ou want to continue(y or n): y
```

```
Commands:- 0.Get   1.GetNext      3.Set   4.SetNext
: Names:-  SysDescr        SysObjectID     SysUpTime       SysContact
           SysName         SysLocation     SysServices
-------------------------------------------------------------------
 the SNMP command & the Object Name: GetNext SysContact
 the Computer to be communicated: 5
rror
t 60 has been created for host: 10
gent/server returned Value is:
CSD
-------------------------------------------------------------------
u want to continue(y or n): y
```

```
Commands:- 0.Get   1.GetNext      3.Set   4.SetNext
: Names:-  SysDescr        SysObjectID     SysUpTime       SysContact
           SysName         SysLocation     SysServices
-------------------------------------------------------------------
 the SNMP command & the Object Name: GetNext SysLocation
 the Computer to be communicated: 5
rror
t 60 has been created for host: 10
gent/server returned Value is:
 floor FCSD room no: 110
-------------------------------------------------------------------
u want to continue(y or n): y
```

```
Commands:- 0.Get   1.GetNext      3.Set   4.SetNext
t Names:-  SysDescr        SysObjectID     SysUpTime       SysContact
           SysName         SysLocation     SysServices
-------------------------------------------------------------------
 the SNMP command & the Object Name: GetNext SysServices
 the Computer to be communicated: 5
rror
et 60 has been created for host: 10
Agent/server returned Value is:
ware: x86 Family 5 Model 2Steppin 12 AT/AT compatible_software: Windows
on 4.0 ( Build Number: 1381 Uniprocessor Free )
-------------------------------------------------------------------
ou want to continue(y or n): y
```

```
ommands:- 0.Get   1.GetNext      3.Set   4.SetNext
 Names:-  SysDescr        SysObjectID     SysUpTime       SysContact
          SysName         SysLocation     SysServices
-------------------------------------------------------------------
the SNMP command & the Object Name: Set SysDescr
the computer to be communicated: 5

is a Read Only Statement. You can't Write Over It
-------------------------------------------------------------------
u want to continue(y or n): y
*******************************************************************
-------------------------------------------------------------------
ommands:- 0.Get   1.GetNext      3.Set   4.SetNext
 Names:-  SysDescr        SysObjectID     SysUpTime       SysContact
          SysName         SysLocation     SysServices
-------------------------------------------------------------------
the SNMP command & the Object Name: Set SysObjectID
the computer to be communicated: 5

is a Read Only Statement. You can't Write Over It
-------------------------------------------------------------------
u want to continue(y or n): y
```

```
 Names:-  SysDescr          SysObjectID      SysUpTime        SysContact
          SysName           SysLocation      SysServices
-----------------------------------------------------------------------
the SNMP command & the Object Name: Set SysUpTime
the computer to be communicated: 5

is a Read Only Statement. You can't Write Over It
-----------------------------------------------------------------------
u want to continue(y or n):
```

```
-----------------------------------------------------------------------
ommands:- 0.Get   1.GetNext      3.Set   4.SetNext
 Names:-  SysDescr          SysObjectID      SysUpTime        SysContact
          SysName           SysLocation      SysServices
-----------------------------------------------------------------------
the SNMP command & the Object Name: Set SysContact
the computer to be communicated: 5
the data to be set in: Jestin KCT, Coimbatore.

-----------------------------------------------------------------------
```

```
-----------------------------------------------------------------------
ommands:- 0.Get   1.GetNext      3.Set   4.SetNext
 Names:-  SysDescr          SysObjectID      SysUpTime        SysContact
          SysName           SysLocation      SysServices
-----------------------------------------------------------------------
the SNMP command & the Object Name: Get SysContact
the Computer to be communicated: 5
ror
t 60 has been created for host: 10
gent/server returned Value is:
n KCT,Coimbatore
-----------------------------------------------------------------------
u want to continue(y or n): y
```

```
-----------------------------------------------------------------------
ommands:- 0.Get   1.GetNext      3.Set   4.SetNext
 Names:-  SysDescr          SysObjectID      SysUpTime        SysContact
          SysName           SysLocation      SysServices
-----------------------------------------------------------------------
the SNMP command & the Object Name: Set SysName
the computer to be communicated: 5
the data to be set in: ISRO Computer1
```

```
Commands:- 0.Get    1.GetNext       3.Set    4.SetNext
ct Names:- SysDescr          SysObjectID     SysUpTime       SysContac
           SysName           SysLocation     SysServices
-------------------------------------------------------------------
r the SNMP command & the Object Name: SetNext SysDescr
r the computer to be communicated: 5

 is a Read Only Statement. You can't Write Over It
-------------------------------------------------------------------
ou want to continue(y or n): y
```

```
Commands:- 0.Get    1.GetNext       3.Set    4.SetNext
t Names:- SysDescr          SysObjectID     SysUpTime       SysContac
          SysName           SysLocation     SysServices
-------------------------------------------------------------------
 the SNMP command & the Object Name: SetNext SysObjectID
 the computer to be communicated: 5

 is a Read Only Statement. You can't Write Over It
-------------------------------------------------------------------
ou want to continue(y or n): y
```

```
Commands:- 0.Get    1.GetNext       3.Set    4.SetNext
 Names;- SysDescr          SysObjectID     SysUpTime       SysContact
         SysName           SysLocation     SysServices
-------------------------------------------------------------------
the SNMP command & the Object Name: SetNext SysUpTime
the computer to be communicated: 5
the data to be set in: Jestin KCT, Coimbatore.
```

```
t Names:-  SysDescr          SysObjectID      SysUpTime        SysContac
           SysName           SysLocation      SysServices
-------------------------------------------------------------------------
 the SNMP command & the Object Name: SetNext SysLocation
 the computer to be communicated: 5

 is a Read Only Statement. You can't Write Over It
-------------------------------------------------------------------------
ou want to continue(y or n): y
```

```
Commands:- 0.Get   1.GetNext       3.Set   4.SetNext
t Names:-  SysDescr          SysObjectID      SysUpTime        SysContac
           SysName           SysLocation      SysServices
-------------------------------------------------------------------------
 the SNMP command & the Object Name: SetNext SysContact
 the computer to be communicated: 5
 the data to be set in: ISRO Computer1
```

```
-------------------------------------------------------------------------
Commands:- 0.Get   1.GetNext       3.Set   4.SetNext
 Names:-   SysDescr          SysObjectID      SysUpTime        SysContact
           SysName           SysLocation      SysServices
-------------------------------------------------------------------------
the SNMP command & the Object Name: Get SysName
the Computer to be communicated: 5
ror
t 60 has been created for host: 10
gent/server returned Value is:
Computer1
-------------------------------------------------------------------------
u want to continue(y or n): n
*************************************************************************
ime after complilation is:                              12:54:50.20
-------------------------------------------------------------------------
ifference in the time is:                               0:02:26.250
-------------------------------------------------------------------------
```

APPENDIX B:

DEFINITIONS:

- **ACCESS POLICY:**
  The comparison between SNMP community and SNMP profile. It gives the community profile required to gain access to an agent.
- **AUTHENTIC SNMP MESSAGES:**
  SNMP messages created by one NMS and received by an agent within the same community.
- **AUTHENTICATION SCHEME:**
  Rules that are used to check whether the SNMP message belong to the SNMP community.
- **LABEL:**
  The combination of object identifier and descriptive text is known as label.
- **PROTOCOL ENTITIES:**
  All peer processes on which the SNMP has been implemented and which consequently support SNMP application entities are described as protocol entities.
- **SNMP APPLICATION ENTITIES:**
  Management levels, which are, implemented both in the management station and network element (agent) in conjunction with SNMP protocol.
- **PASSIVE SOCKET:**
  Socket used by a server to wait for the incoming connection.
- **ACTIVE SOCKET:**
  Socket used by a client to initiate a connection
- **STANDARD APPLICATION PROTOCOL:**
  Application protocol documented in the RFC and adopted as part of the TCP/IP protocol suite.
- **NONSTANDARD APPLICATION PROTOCOL:**
  Protocols invented by programmers for private use.
- **FULLY PARAMETERIZED CLIENT:**
  Software that specifies protocol port number has more input parameters than other software.

ABBREVIATION:

SL. No.	Abbreviation	Description
1.	ASN.1	Abstract Syntax Notation One
2.	CCITT	Committee Consultant International Telegraphic and Telephones.
3.	DOD	Department Of Defense
4.	HEMS	High Level Entity Management System
5.	IAB	Internet Architecture Board
6.	IANA	Internet Assigned Numbers Authority
7.	ISO	International Standard Organization
8.	LAN	Local Area Network
9.	MIB	Management Information Base
10.	RFC	Request For Comment
11.	SGMP	Simple Gateway Monitoring Protocol
12.	SMI	Structure and Identification of Management Information
13.	SNMP	Simple Network Management Protocol
14.	WAN	Wide Area Network
15.	TCP	Transmission Control Protocol
16.	IP	Internet Protocol
17.	ANSI	American National Standards Institute
18.	IETF	Internet Engineering Task Force