Business Education Journal
Volume II Issue II
Email: *jeditor@cbe.ac.tz*
Published Online January, 2019 in *http://www.cbe.ac.tz/bej*

CONCEPTS ON DEVELOPING NETWORK MONITORING SOFTWARE

Ahmed Kijazi, Assistant Lecturer, Department of Mathematics & ICT, College of Business, Education,
P.O. Box 1968, Dar es Salaam-Tanzania, Tel: +255712307240,
Email: kijaziahmed@gmail.com, a.kijazi@cbe.ac.tz

ABSTRACT

Network Monitoring involves using different hardware and software tools (Systems) or both to continuously observe the status of network devices or hosts, and notify the network administrator through email, SMS or other alarms in case of error or failure. This may happen when network monitoring software observe the status of network devices or host speaks with their corresponding protocols within the Open System Interconnection Model stack (OSI Layer).The aim of this paper is to explain important concepts used in implementing software for monitoring hosts and network devices to inexperienced programmers and researchers. This paper explains how a Simple network Management Protocol (SNMP), Internet Control Message Protocol (ICMP), Dynamic Host Control Protocol (DHCP), Domain Name Service (DNS), Hypertext Transfer Protocol (HTTP), Management Information Base (MIB) and Port Scanning concept can be implemented to form up a network monitoring software. During implementation, the following software have been suggested, Java NetbeansIDE for the Java programming platform, Manage engine for Identifying SNMP Object Identifier (OID) numbers and their meaning from network devices and hosts, and SNMP for Java (SNMP4J) Application Program Interface (API) for providing SNMP Libraries for the Netbeans IDE.

Keywords: *SMTP, Port Scanning, MIB, SNMP Agent, DHCP, DNS, HTTP, Service, SNMP Manager, ICMP.*

1.0 INTRODUCTION

Normally network monitoring software are used for monitoring various network parameters in our networks. Even researchers sometimes utilize network monitoring software for analysis purposes depending on the nature of the project. However, the concepts in developing this monitoring software remain to be a secret of the respective companies. For example, the researcher can determine the current operating temperature of a device or availability of a particular network service using a certain network monitoring software without knowing the exact techniques used in the software to obtain such parameters. Consequently, this lack of knowledge prohibits researcher in going deeper in their findings. For example, (Khan, 2013) in his study on efficient network monitoring and management system, he used Nagios for the SNMP monitoring part however few lines of Java program could do the same task. Similarly, most of the studies do not disclose basic concepts for developing network monitoring software rather than explaining the improvements of these tools. The same problem happens to inexperienced programmers while engaged in network monitoring software development industry. In general, it can be said to be a challenge to obtain information regarding developing network monitoring tools. This paper tries to disclose implementations of important techniques which most of the network monitoring software should have. Network Monitoring is not only concerned with Monitoring of the physical network and host device such as bridges, routers, computers and hubs, but also it deals with monitoring of services which are running on these devices such as data storage services, data manipulation services, data presentation services and data communication services.

All these services are running in the network Layer and above. In fact, this paper explains how to develop a software for monitoring the application layer services, which are Domain name service (DNS), Dynamic host control protocol (DHCP), Simple Message Transfer protocol (SMTP) and Hypertext transfer protocol (HTTP). The application layer services are executed on most important servers such as mail servers, web servers, name resolution servers and IP addresses servers. In this paper, the monitoring process is divided into three forms as follows (Figure 7): (1). Monitoring of application services (2). Monitoring of hosts and network devices, and (3). Monitoring of other parameters. However, all these categories are implemented using the Java programming to form a network monitoring software.

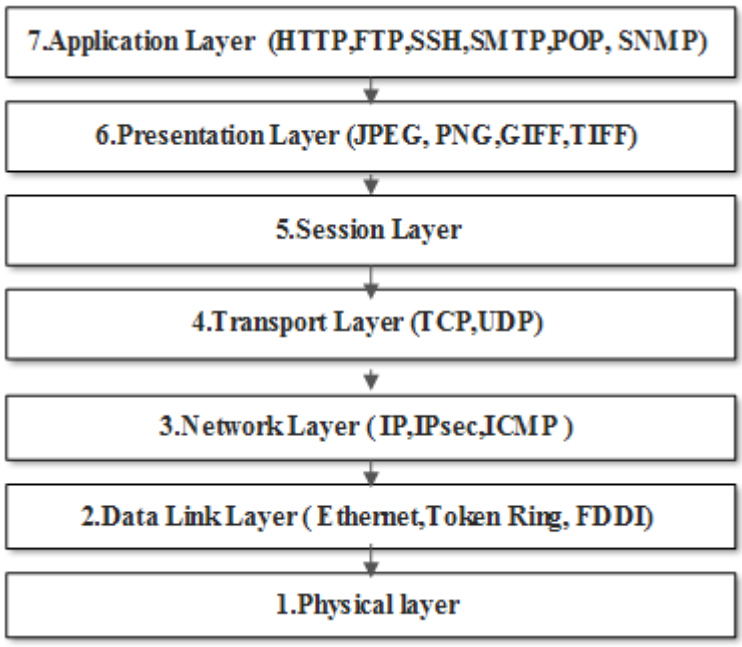

Figure1. A diagram for OSI 7 layers and services.

METHODOLOGY

In order to demonstrate the concept of developing a network monitoring software consider a waterfall model of software development whereby the process of developing the following modules are explained in details: - (1). Monitoring of application services (2). Monitoring of hosts and network devices, and (3). Monitoring of other parameters.

2.1 Requirement Gathering and analysis

Requirements were identified by exploring functionalities of different network monitoring software such as Nagios, PRTG (Nagios, 2018) (Paessler, 2018) network monitor and Opsview. Thereafter, common functionalities where categorized into three groups namely; monitors applications and services, host and network devices, and other parameters. The concepts utilized on implementations of these functionalities were analyzed and compared with those used in other application through literatures. It was observed that, the skills used for implementation of these functionalities were borrowed from network troubleshooting using command line interface, e.g.Ping and trace root, port programming using TCP and UDP protocols and Management information base (MIB) browser utilities. Although, these concepts are being implemented in network monitoring software which are sold at a high prices.

2.2 System Design

In order to understand the network monitoring software development concept, consider a simple network model (Figure 2) which consist of different networking devices, Management station and servers installed with different applications such as web, email and domain name resolution which execute HTTP, SMTP and DNS service respectively. All network devices, servers and services running on them were monitored by our network monitoring software which was installed on the network monitoring station (NMS).The network monitoring software is the one going to be developed.

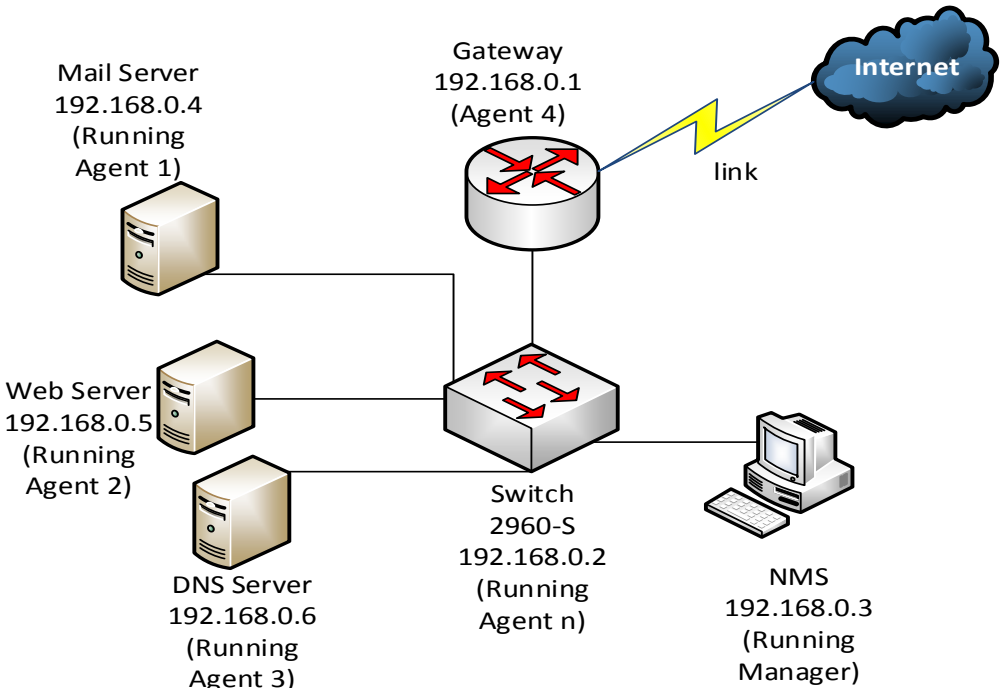

Figure 2. Monitored Local Area Network (LAN)

In a nutshell, this network is composed of five key elements which are:-

i. Managed device – Sometimes called network elements, the managed devices can be any type of device, including, but not limited to, routers, access servers, switches, hubs, IP video cameras, bridges, IP telephones, printers and computer hosts.
ii. Running agents - software which runs on the managed device (Firmware and OS)
iii. Network services – These are applications software running on the Mail, DNS and Webserver
iv. Network management system (NMS) - A computer which host our network monitoring software
v. Running Manager – Our network monitoring software installed on the NMS.

Therefore, the running manager is the one monitor devices availability, services availability and other parameters of devices via their running agents. The monitoring process is done through three modules of the running manager as follows:

2.2.1 Devices availability monitoring module

This module monitors device availability in the network. However, it does not care about the services running on them. Monitoring of device availability is done through the ICMP protocol. This module utilizes the ICMP ping packets to monitor availability of any devices in the network by periodically sending the ICMP echo request to each device using their respective IP addresses. If the device echoes reply indicates not reachable means the device is down the vice versa device is up (Ahsan *et al*, 2003) (Figure 3). In order for the program to send the ICMP ping the programmer does not need to have deep knowledge about the ICMP packet structure. Fortunately, Java provides libraries for sending a ping packet to the devices using their IP addresses (David & Michael, 2002).

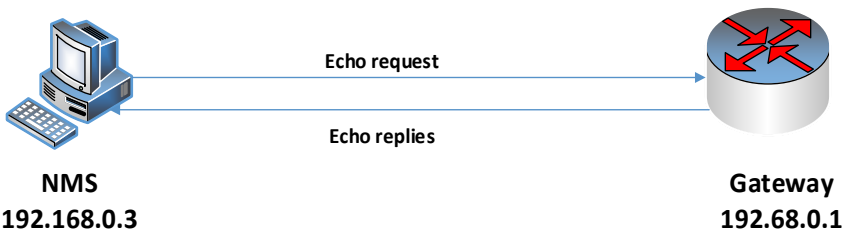

Figure 3. Management Station (NMS) detects the device availability

2.2.2 Service availability monitoring module

This specifically monitors the availability of services running on the servers. These Services are the ones providing different functionalities in the network such as mailing, address resolution and Web site hosting. Monitoring of Service availability is performed by using port scanning technique. A service monitoring module of the running manager, periodically establishing connection to the servers using Socket programming, see (Figure 4). A socket consists of a port number of a particular service and IP address of the server hosting that service which a running manager wishes to establish a connection with. By default Netbeans IDE does not support socket programming. In order to enable Socket programming java.net.* library should be imported into the Netbeans IDE before programming. The socket should be enclosed within the try and catch block so that any disconnection will be caught while connection remains in the try block (David & Michael, 2002).

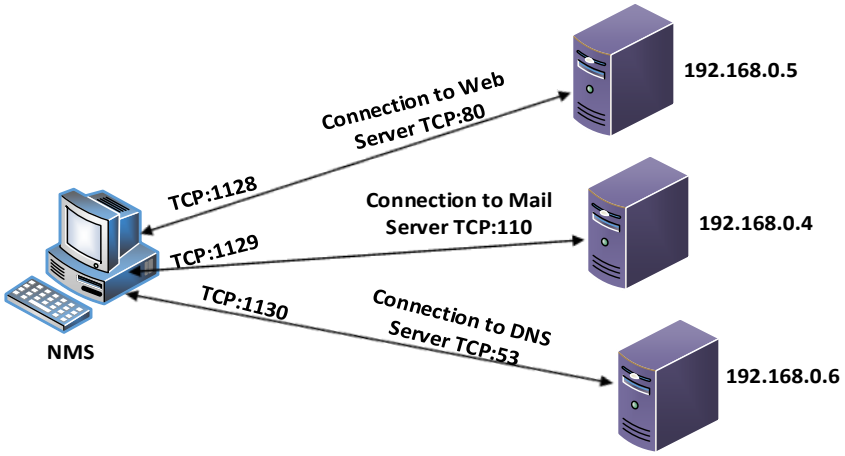

Figure 4. NMS Monitor Service availability by establishing a TCP connection with them

2.2.3 Other parameters monitoring module

Other parameters are those apart of services availability and device availability. These are all parameters which can be determined via SNMP protocols such as the location of all devices in the network with their names, the elapse time since the devices are up, status of all ports of the switches (on or off), bandwidth allocated for all switches ports, number of users configured on each server, fan status of the switches (on or off), OS version which is running on devices, current operational temperature of switches, amount of physical memory available in each server, etc. Within a running agent each of these parameters are stored in the database called a management information base (MIB) in a tree form (Figure 6) and they are identified through their unique identifiers known as SNMP Object Identifiers (OIDs).For example, (Table 1) shows parameters of Cisco switch 2960-S and their corresponding OIDS. In this case, a programmer is required to read the SNMP device manual of the device before further development processes of software in order to understand the SNMP Object Identifies (OIDs) structure, and their meaning in the management information base (MIB). The running agent can access particulars of mentioned SNMP services with the help of SNMP for Java (SNMP4J) libraries imported in the Netbean IDE during development (Figure5). The libraries are imported because the IDE does not know anything about SNMP. The libraries act as translator between running manager and agent. Requests can be either a retrieve or change management information of the desired

running agent (Rane, 2007). Remember, SNMP operates in port number 161 and disabled in the devices which support this protocol, so it should be enabled.

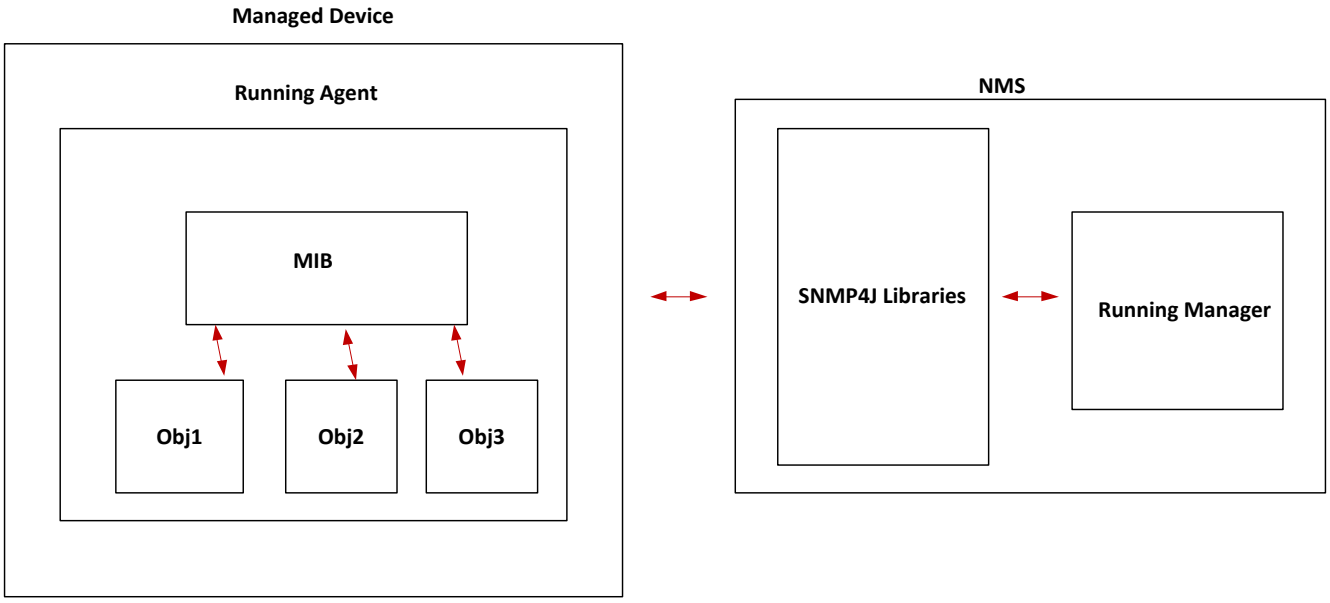

Figure 5. Running manager and agent communicate using SNMP4J Library

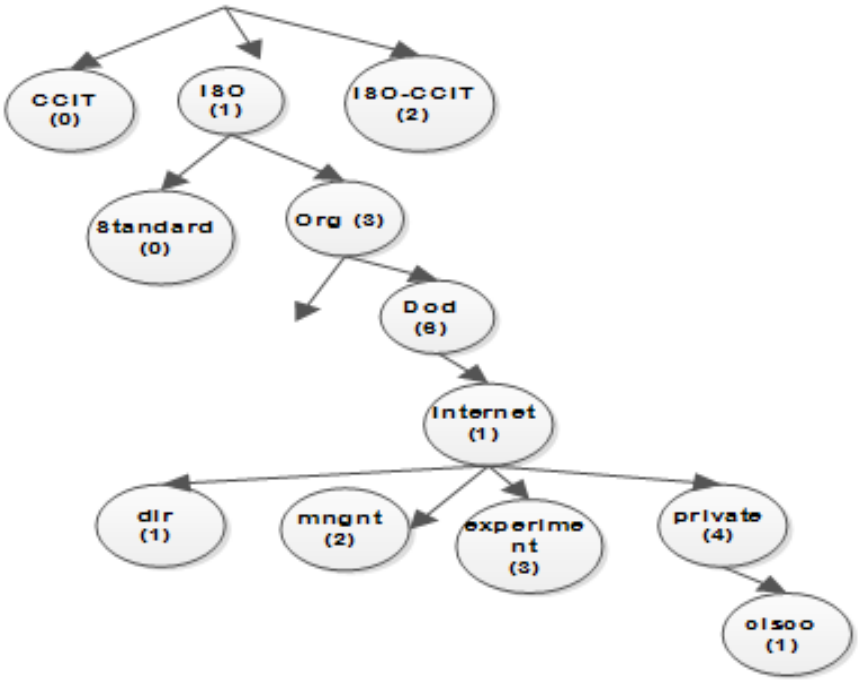

Figure 6. MIB configuration organization structure

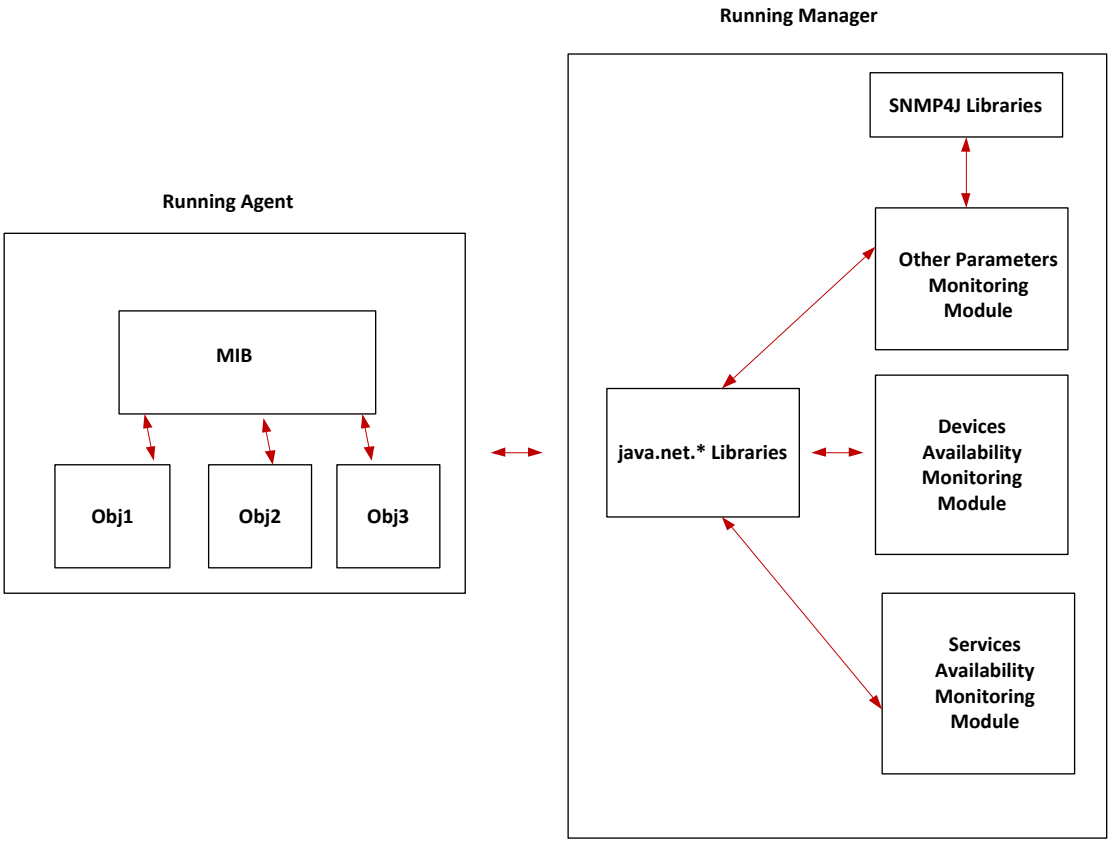

Figure 7.System Architecture of the network monitoring software (Running manager)

Table (1).OIDs number and their meaning for CISCO switch 2960-S

OID number	Configuration	Device
.1.3.6.1.2.1.2.2.1.8	Shows Switch port status	CISCO (2960-S)
.1.3.6.1.2.1.2.2.1.5	Switch ports bandwidth	CISCO (2960-S)
.1.3.6.1.2.1.1.6.0	Device, Location	Any
.1.3.6.1.2.1.1.3.0	Time elapse since the device is up	Any
.1.3.6.1.2.1.25.1.5.0	Current Number of Users in the System	Any
.1.3.6.1.2.1.25.2.2.0	Amount of Physical Memory	Any
.1.3.6.1.4.1.9.9.13.1.4.1.3	Fan status	CISCO (2960-S)
.1.3.6.1.4.1.9.9.13.1.3.1.3	Device Current working Temperature	CISCO (2960-S)
.1.3.6.1.2.1.1.1.0	OS Description	Any

2.3 System implementation

During implementation Netbeans IDE has been used as a Java programming platform. This is because Java supports network programming through networking libraries. Most of basic networking libraries are available in Netbean however; more libraries may be imported whenever they are needed. In our case, java.net.* and SNMP4J libraries were downloaded and imported into the Netbean IDE (Figure 8,9 & 10). SNMP4J libraries have been used by a network monitoring software (running manager) for monitoring other parameters.

Figure 8: SNMP for Java Libraries' (SNMP4J,2018)

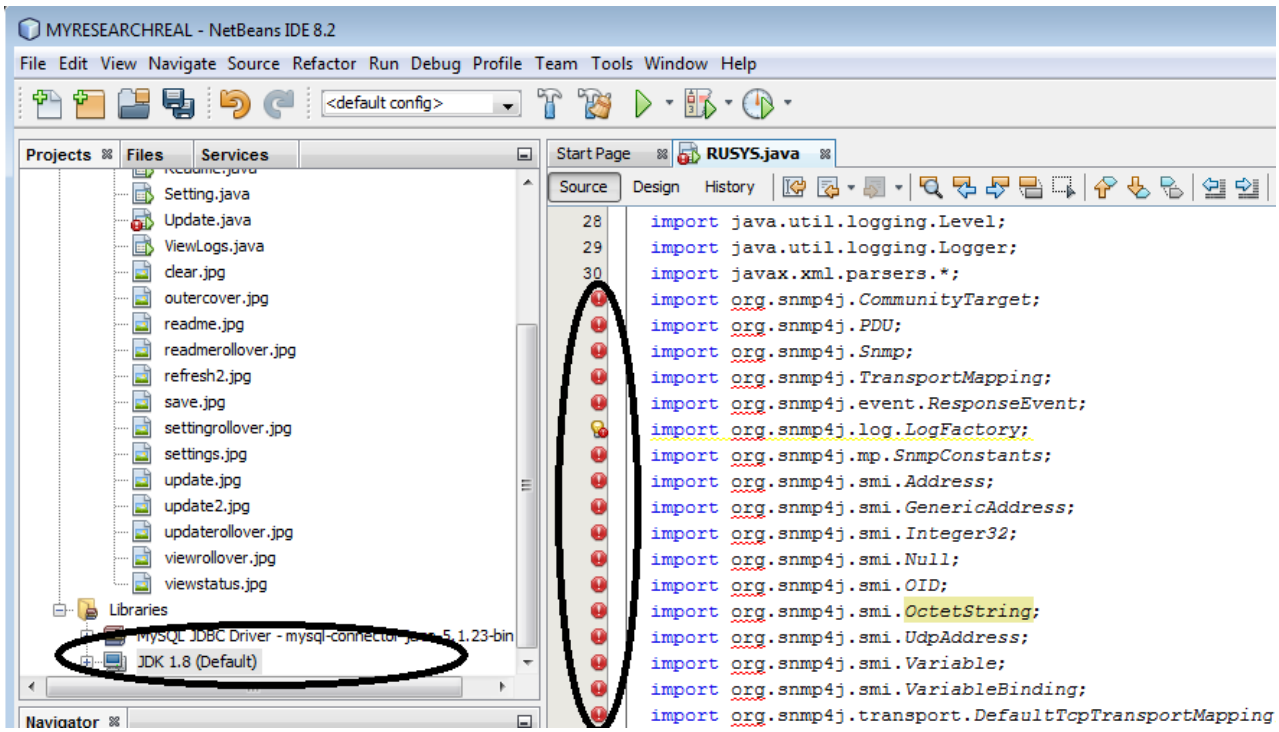

Figure 9: SNMP4J libraries before being imported in Netbean IDE

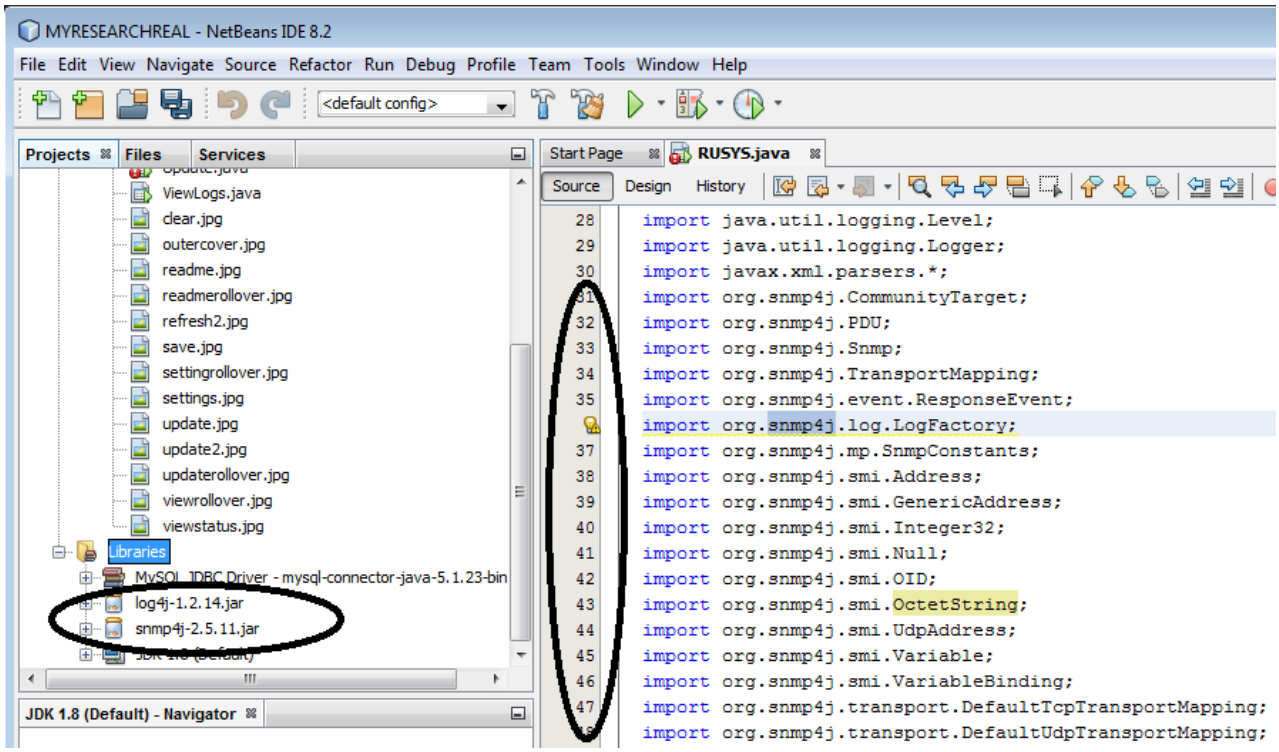

Figure 10: SNMP4J libraries after being imported in Netbean IDE

2.3.1 Implementation of device availability monitoring module

The implementation of this module borrows the PING concept of network troubleshooting using Command line interface (CMD). In contrast, this concept was implemented in the Java programming language for detecting device availability. The module is implemented in such a way that it periodically sends ICMP ping packets to all devices in the network in order to detect their availability. The (Figure 11) shows a sample class of a monitoring device availability module that detects web server availability using its IP address 192.168.0.5. The class sends ping request to the web server using r.exec() function in order to determine its availability. Similarly, the module does the same for detecting the availability of other devices.

```
class WEBServer_availability_test
{
   String a[]=new String[100];
   String webipaddress="192.168.0.5"
   //Don't forget to leave a space after ping
   String pingcmd = "ping " + webipaddress;
   Runtime r = Runtime.getRuntime();
   Process p;
   try
   {
     //Ping Webserver using a ping function
     p = r.exec(pingcmd);
     BufferedReader in = new BufferedReader(new InputStreamReader(p.getInputStream()));
     String inputLine;
     int k=0;
     while ((inputLine = in.readLine()) != null)
     {
        a[k]=inputLine;
        k=k+1;
     }
     String ad=a[2].concat(a[3]);
     boolean ad1=ad.toLowerCase().contains("ttl");
     if(ad1==true)
         {
           //Display Message if server is Up
           System.out.println("Web Server Machine is Up");
         }
         else
         {
         //Display Message if web server is down
          System.out.println("Web Server Machine is down");
         }
   }
   catch(Exception ex)
   {
     //Display Message if web server is down
     System.out.println("Web Server Machine is down");
   }
}
```

Figure 11: A class of web server availability monitoring module

2.3.2 Implementation of service availability monitoring module

Just like in device availability detection module, this module borrows concepts of port programming using TCP or UDP protocols. This module detects service availability by periodically trying to establish connection to the services running on servers using socket programming. A socket consists of an IP address and port number of the service that a module needs to establish connection with. Sometimes, the firewall prevents connections from outside services, so it should be enabled during implementation. Similarly, the same concept has been used for developing charting applications. Figure 12. Shows a Java class trying to establish TCP connection with web server service running in port number 80. The connection could be TCP or UDP depending on the nature of application layer service which the running manager wish to establish connection (Alotaibi, et al. 2017).

```
class WEB_service_availability
{
  String webipaddress="192.168.0.5";
  String webportno=80;
  try
  {
    Socket s=new Socket(webipaddress,webportno);
    s.setSoTimeout(5*1000);
    System.out.println("***********Web service is on***********");
    System.out.println("");
    s.close();
  }
  catch(Exception ex)
  {
    System.out.println("***********Web service is off***********");
  }
}
```

Figure 12: A class of web services availability monitoring module

2.3.3 Implementation of other parameters monitoring module

Implementation of other parameters monitoring modules borrows concept from management information base browser applications. This application software is used to find the correct OID number in a given MIB configuration file regardless of its complexity structure. With the help of MIB browser software and correct IP address you can access information on any device such as device name, the elapse time since the device is up, port status of switch etc. with their OIDS. Table1, shows a sample of information extracted from CISCO switch 2960-S using a free MIB browser tool named MIB browser engine (Manage engine, 2017) (Figure 14). In this paper devices information extracted using MIB browser engine were the ones called other parameters. With the help of SNMP4J, the extracted OIDS were implemented in other parameters monitoring module of a running manager to trace their corresponding parameter's value. On the other hand, this is the reverse process. Remember, SNMP4J were imported in Netbean IDE before (Figure 9 & 10).

```
pdu.setType(PDU.GET);f
ResponseEvent event = snmp.send(pdu, getTarget(), null);
if(event != null) {
return event;
}
throw new RuntimeException("GET timed out");
}
//This method returns a Target, which contains information about
private Target getTarget() {
Address targetAddress = GenericAddress.parse(address);
CommunityTarget target = new CommunityTarget();
target.setCommunity(new OctetString("public"));
target.setAddress(targetAddress);
target.setRetries(2);
target.setTimeout(1500);
target.setVersion(SnmpConstants.version2c);
return target;
}
}
```

Figure 13: A class of other parameters monitoring module

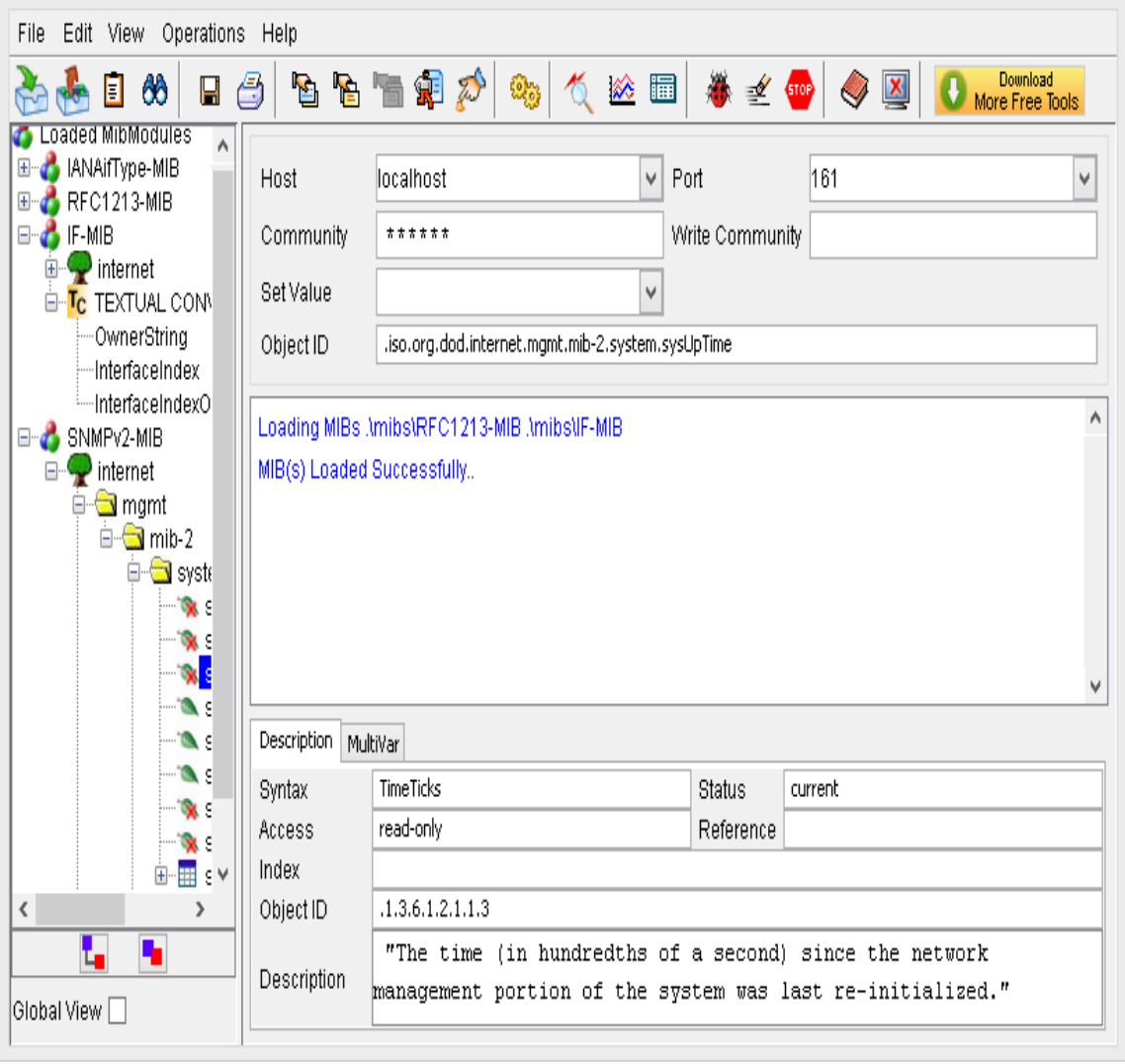

Figure 14.MANAGEENGINE Free Software for browsing devices OIDs

2.4 System Testing

Testing was done by setting a configured network (Figure 2).The network monitoring software was installed on the Network monitoring station (NMS).

2.4.1 Testing of device and services availability

The program was execute while all devices and services in the network are switched on and their status was observed through the program. The same procedure was repeated when some of the devices and services are switched off. The system was successfully indicating the status in all cases.

2.4.2 Testing of other parameters

This testing phase involves SNMP. The OIDS of each parameter were inserted in a program one after another and the feedback observed. The system successfully returned the status of all parameters requested from each device.

CONCLUSION

This paper provides concept on how network monitoring software is developed. This paper will enable beginner network programmers to acquire basic concepts on how to implement different network monitoring techniques in network monitoring software. Not only that, this paper will provide a footstep to anybody who wishes to engage in network monitoring software development. Similarly, this paper will enable researchers to understand the basic concepts regarding working principles and implementation of network monitoring software, so that

they can go further miles in their research. However, Netbean IDE and SNMP API for Java are proposed as tools for network monitoring software development, this does not limit developers to use other tools with the same capabilities.

REFERENCES

Ahsan H., Mohamed M., Hefeeda, and Bharat K., (2003). Detecting Service Violations and DOS Attacks. *In Proc. of Network and Distributed Systems Security Symposium (NDSS'03),pages 177-189.*

Alotaibi, A. M., Fahaad Alrashidi, B., Naz, S., & Parveen, Z. (2017). Security issues in Protocols of TCP/IP Model at Layers Level. *International Journal of Computer Networks and Communications Security*, *5*(5), 96–104. Retrieved from www.ijcncs.org

Khan, R. (2013). An Efficient Network Monitoring and Management System. *International Journal of Information and Electronics Engineering*, *3*(1). https://doi.org/10.7763/IJIEE.2013.V3.280

David R. and Michael R., (2002). Network programming and distributed computing. *Addison Wesley, Pub.* ISBN: 0-201-71037-4.

Manage engine, (2017). A Free MIB browser software, (2017). Retrieved from http:// www.manageengine.com

Nagios, (2018). Server and application monitoring tools. Retrieved from https://www.nagios.org/

Rane, (2017). Simple Network Management Protocol. Retrieved from http://www.rane.com

Paessla, (2018). PRTG Network Monitoring Software. Retrieved from https://www.paessler.com

Sivakumar S.R and Mangaiyarkarasi. R., (2012). Network Monitoring Using SNMP Protocol.*International Journal of Power Control Signal and Computation (IJPCSC)*, Vol3. No1. Jan-Mar 2012 ISSN: 0976-268X.

SNMP4J, (2018). Classes and interfaces for creating, sending, and receiving SNMP messages. Retrieved from http://www.snmp4j.org/doc/index.html