Data Classifications in Network Monitoring System

Mr. Vishal Kumar
Sr. Software Engineer
Mind Tech Technologies Pvt. Ltd.
Electronics City , Bangalore-560100, India
Email : vishal.bhu.nitk@gmail.com

Mr. Pramod Verma
Sr. Software Engineer
Mind Tech Technologies Pvt. Ltd.
Electronics City , Bangalore-560100, India
Email : praver87@gmail.com

***Abstract*-In this ever-growing world, Internet plays a vital role in everyday life. We can access any resources at any time through the Internet. Internet is a reservoir of information that lets the user to browse and get any information. Using internet, we can send instant messages, send electronic mails to any one, host web pages and so on. The main aspect is to develop WAAS based(a specialized software designed for CISCO routers)monitoring system. Here,the objective is to identify the system information. The Network Monitoring will develop an operating system independent software where the Server will be able to identify all the clients and also it should maintain the security over the network and also to help the clients over the network to maintain their confidentiality and integrity of their data. This assists network administrator to efficiently manage whole network and thereby prohibiting the clients to perform illegal modifications**.

1.INTRODUCTION

Most applications that communicate over a network, where it's the Internet or small office network, use the same principles and functionality to perform their communication. One application sits on a computer, waiting for another application to open a connection. This application is "listening" connection request, much like you listen for the phone to ring if you are expecting someone to call. Mean while, another application, most likely on another computer (but not necessarily) try to connect to the first application. This attempt to open a connection is similar to calling someone on the telephone. You dial the number and hope that other person is listening for the phone on the other end. As the person making the call, you have to know the phone number of the person you are calling. If you don't know the phone number, you can look it up using the person's name. Likewise, the application trying to connect to the first application has to know the network location, or address of the first application.

Once the connection is made between the two applications, messages can pass back and forth between the applications, much like you can talk to the person on the other end of the phone. This connection is a two-ay connection channel, with both sides sending information.

Finally, once one or both side have finished their conversation, the connection is closed, much like you hang up the phone after you have finished talking to the person you called. Once the connection is closed from one side, the other side can detect it and close its side, just like you can tell if the person on the other end of the phone has hung up or if you've been disconnected by some other means. This is a basic explanation of how network communications work between two or more applications.

The basic object used by applications to perform most network communications is called socket. Socket was first developed on UNIX at the University of California at Berkley. Sockets were designed so that most network communications between applications could be performed in the same way that these same applications would read and write files. Sockets have progressed quite a bit since then, but the basics of how they work are still same.

During the days of windows 3.x, before networking was built into the windows operating system, you could buy the network protocols required for network communications from numerous different companies Many applications developers were not happy with this situation. As a result, all the networking companies, including Microsoft, get together and developed the Winsock (Windows Sockets) API. When you want to read or write a file, you must use a file object to point to the file.

A socket is similar to an object used to read and write messages that travel between applications. Making a socket connecting to another application does require a different set of information than opening a file. To open a file, you need to know the file name and location. To open a socket connection, you need to know the computer on which the application is running and the port on which it is listening.

A port is like a phone extension, and the computer address is like phone number. If you call someone at a large office building, you may dial the main office number, but then you need to specify the extension number. Likewise, ports are used to route network communications. As with the phone number, there are means of looking up the port number, if you do a different application like making phone call, someone other then the person you called may answer the phone call. You may not get an answer if there is no application listening at the other end.

2.CLIENT-SERVER MODEL

Most networking applications assume one side as the client and the other side as the server. The purpose of the application is for the server to provide some defined service for clients. Normally a client initiates the conversation while a server waits for clients to start conversations with it. In some cases, the same program may be both a client and a server.

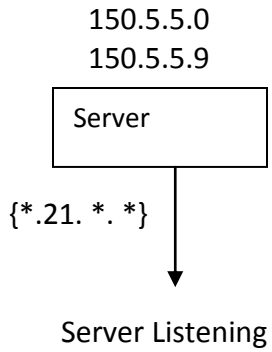
Server

Server always listens to a well-defined port for incoming request. There are two types of servers.

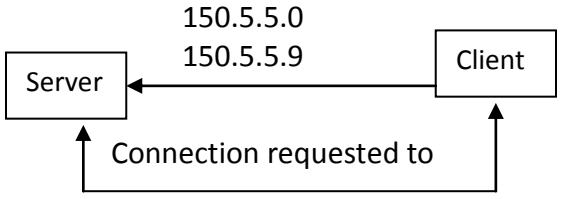
1. Iterative Server
2. Concurrent Server

1.Iterative server

a) Initially it waits for the incoming connection



b) Whenever the client initiates the request, it receives it and sends the request back to the client



150.5.5.0, port 21

not already know what it is, but this requires your computer to be configured with the information about which port the connecting application is listening on. If you specify the wrong computer address or port number, you may get a connection of

c) Then the server goes back to the first step for listening connection request from other clients.

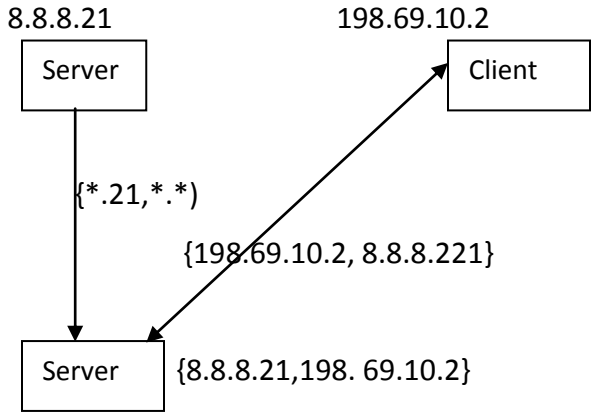
2. Concurrent server

a) Server waits for the incoming connection.

b) Start a new server to handle the new client request. This may involve creation of new process, task or thread, depending on what the underlying operation system supports.

c) The created new server handles the client request, when complete the server will be terminated.

d) Go back to the step 1 for listening



3.NETWORK PROGRAMMING

One of Java's great strengths is painless networking. The Java network library designers have made it quite similar to reading and writing files, except that the "file" exists on a remote machine and the remote machine can decide exactly what it wants to do about the information you're requesting or sending. The programming model you use is that of a file; in fact, you actually wrap the network connection (a "socket") with stream objects, so you end up using the same method calls as you do with

all other streams. In addition, Java's built-in multithreading is exceptionally handy when dealing with another networking issue: handling multiple connections at once.
Inet Address -This class represents an Internet Protocol (IP) address.

Socket:A socket is a connection between two hosts. It can perform seven basic operations:

1. Connect to a remote machine
2. Send data
3. Receive data
4. Close a connection
5. Bind to a port
6. Listen for incoming data
7. Accept connections from remote machines on the bound port.

Java's Socket class, which is used by both clients and servers, has methods that correspond to the first four of these operations. The last three operations are needed only by servers, which wait for clients to connect to them.

Each program reads from and writes to a socket in much the same way that you open, read, write to, and close a file. Essentially, there are two types of sockets:

- One is analogous to a telephone (a connection-oriented service, e.g., Transmission Control Protocol)
- One is analogous to a mailbox (a connectionless "datagram" service, e.g., User Datagram Protocol).

An important difference between TCP connection sockets and UDP datagram sockets is that TCP makes sure that everything you send gets to the intended destination; UDP, on the other hand, does not. Much like mailing a letter, it is up to you, the sender, to check that the recipient received it. The difference between the two protocols is very similar to comparing the differences between using the phone to talk to friends and writing them letters.

A socket is sometimes called a "pipe" because there are two ends (or points as we occasionally refer to them) to the communication. Messages can be sent from either end. The difference, as we will soon see, between a client and a server socket is that client sockets must know beforehand that information is coming, whereas server sockets can simply wait for information to come to them. It's sort of like the difference between being recruited for a job and actively seeking one. Sockets are a visualization mechanism for a software buffering scheme that is implemented deep in the bowels of the transport layer of the TCP/IP stack.

Java supports TCP and UDP sockets using different classes.

- **TCP Socket**

Java supports TCP sockets using ServerSocket and Socket class.

- Server Socket: This class implements server sockets. A server socket waits for requests to come in over the network. It performs some operation based on that request, and then possibly returns a result to the requester.
- Socket: This class implements client sockets (also called just "sockets"). A socket is an endpoint for communication between two machines.

- UDP Socket:

Java supports UDP socket through Datagram Socket and Datagram Packet class.

- Datagram Socket*:* This class represents a socket for sending and receiving datagram packets. A datagram socket is the sending or receiving point for a packet delivery service. Each packet sent or received on a datagram socket is individually addressed and routed. Multiple packets sent from one machine to another may be routed differently, and may arrive in any order. UDP broadcasts sends and receives are always enabled on a DatagramSocket.
- Datagram Packet*:* This class represents a datagram packet. Datagram packets are used to implement a connectionless packet delivery service. Each message is routed from one machine to another based solely on information contained within that packet. Multiple packets sent from one machine to another might be routed differently, and might arrive in any order.

4.SYSTEM ANALYSIS

System analysis is the phase or step of the systems approach to problem solving using computers. It is a process of gathering and interpreting facts, diagnosis of problems and using the information to recommended improvements to the existing system.

"Network monitoring" allows you to find out the client details. The only restriction is you must be an administrator. If this Network monitoring was not there, we can get the following information: -

1. System identification details.
2. Device identification details.
3. Drive identification details.
4. Process identification details.
5. File system details.
6. Systems Software details.
7. Interface details.
8. IP address details.
9. Partition details.
10. TCP connection details.
11. UDP details.
12. Charts.

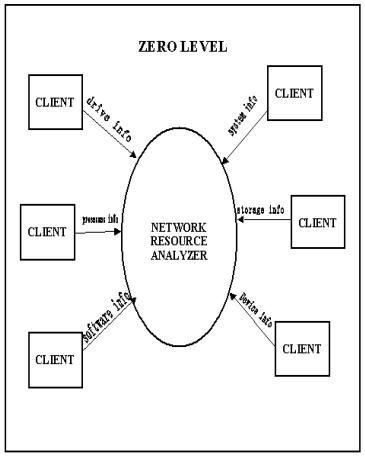


Figure 1:DFD for zero level functioning

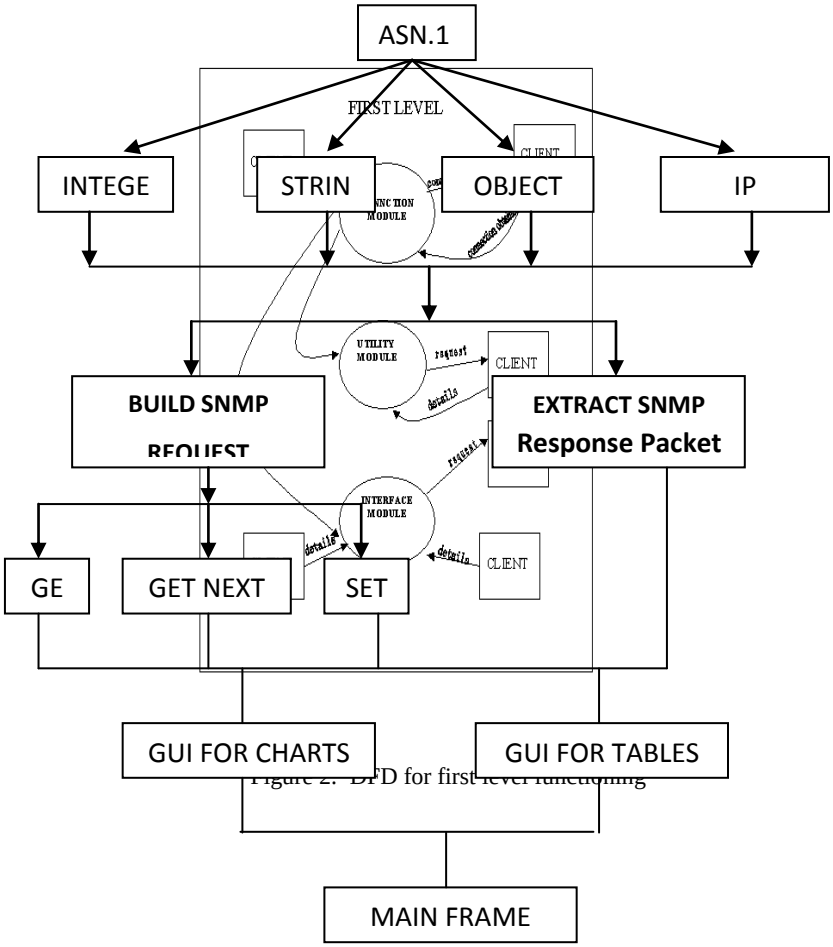


Figure 2: DFD for first level functioning

Figure 3:System Design

System design consists of following modules:

1. Defining the ASN.1 notation for various data types.
2. Building SNMP request and sending the packet.
3. Receiving SNMP response after receiving the packet.
4. Building GUI environment for online performance monitoring.
5. Building GUI environment for monitoring details of remote system in Table format.

5.EDUCATION AND TRAINING

The implementation of the proposed system includes the training of the system operators. Training the system operators includes not only instructions in how to use the equipment, but also in how to diagnose malfunctions and in what steps to take when they occur. So proper training should be provided to the system operations in the delivery information system. The success of the delivery information system depends on the way in which it is operated and used. Therefore the quality of the training given for the operating personnel effects the successful implementation of the system. The training must ensure that the personnel can handle all the possible operations. Training must also include data entry personnel. They must also be given training for the installation of new hardware, terminals, how to power the system, how to power it down, how to detect the malfunctions, how to solve the problems etc. The operators must also the provided with knowledge of trouble shooting which involves the determination of the case of the problem.The proposed system requires trained personnel for operating the system. The data entry jobs must be done utmost carefully in order to avoid errors. This will reduce the data entry errors considerably. It is preferable to provide the personnel with some kind of operating manuals that will explain all the details of delivery information.

7. CONCLUSION

The current network management systems are based on the SNMP protocol. Most of the commercial network components have embedded SNMP agents. Because of the universality of the Internet with TCP/IP protocol, the transport of management information for SNMP management, which is TCP/IP based is resolved automatically. In addition, most of the popular host operating systems come with the TCP/IP suite and thus are amenable to SNMP management.

This is satisfactorily works for Network Resource Analyzer in which we can retrieve the clients overall information just by selecting the clients IP address or domain name of the specified client from output. The main drawback of this system is it cannot identify the software installed on the client and also more than one client's information cannot be viewed at a time.

6. FUTURE ENHANCEMENT

SNMP accommodates the management of devices that do not implement the SNMP software by means of proxies. A proxy is an SNMP agent that maintains information on behalf of one or more non-SNMP devices. The values can be directly got from system rather than getting them through a class.

The above are the main features being involved in the future development.

This topic is an elementary basis of Network monitoring. Modules such as installed software components, permitted use of software, quota allocation of hard disk space, illegal installation and uninstallation can be added which helps the system to monitor the entire system through network. Adding new features to the makes the system sophisticated and useful.In Network Resource Analyzer new features can be added in future and can be improved as needed. Thus limiting the illegal intruders and making the network stable and secure.

8. REFERENCES

[1]. William Stallings, "SNMP, SNMPv2, SNMP v3, AND RMON 1 AND 2", Third edition, Addison-Wesley, 2000.

[2]. Andrew S.Tanenbaum, "Computer Networks", Fourth Edition, Pearson Education, 2005.

[3]. Herbert Schildt, "The Complete Reference Java", Seventh Edition , Tata Mcgraw-Hill , 2009.

[4]. Mani Subramanian, "Network Management Principles and Practice", Second Edition, Addison-Wesley, 1999.

[5]. Ted Grevers, Joel Christner, "Application Acceleration and WAN Optimization Fundamentals", Pearson Education, First Edition, 2007.

[6]. http://www.cisco.com/en/US/products/ps6870/index.html

[7]. http://www.SNMPLink.org

[8]. http://www.javabeginner.com/java-swing/java-swing-tutorial