

Design and Implementation of an Intelligent Network Monitoring and Management Tool in Internet and Intranet

**Prof. Sasikumar Gurumurthy^{1*}, A. Valarmozhi²
and M. Priya³**

*¹School of Computing Science & Engg.,
VIT University, Vellore, Tamil Nadu, India*

**Corresponding Author E-mail: g.sasikumar@vit.ac.in*

*^{2&3}Department Information Technology,
Bannari Amman Institute of Technology, Sathyamangalam, India
E-mail: mvalar113@gmail.com, mpriyashri@yahoo.in*

Abstract

This paper is aimed at developing a version controller which can be used to control any type of documents. This is typically useful when a group of people are concurrently changing the documents in a repository. This project is aimed at internet/intranet based web application, where the people can access documents from different locations. We developed this application using technology asp.net with c#.net. This project is totally a company based project that works with the roles specified in the company. Web based Document Version Controller, which will cover the following modules: Login: User login and access privilege to the repository. Home: Here where the user can access documents based on his role. He can perform almost all types of file operations. Edit: User can edit the existing editable file here. Registration: Any one can create an account, but to access any thing administrator should provide roles. Roles involved in this project. Administrator: He is the root node of the entire hierarchy in tree model repository. Administrator's privilege is to access all the files in the repository. Project Manager: He is the second node after the administrator. He can't access the administrator files; He can access his and his inherited one. Team Leader: He is last node in repository tree model hierarchy. He can access only his files.

Keywords: Network, Management, Control, Internet, Intranet

Introduction

The goal of this paper is to simulate an overflowing ATM network and establish a router with congestion control based on the GCRA algorithm. The TCP/IP protocol suite is the standard requirement for all applications that need to communicate over the Internet. As TCP/IP applications are unable to specify the QoS parameters needed for most Asynchronous Transfer Mode (ATM) services, we tend to use the GCRA Algorithm. The purpose of Cell-Rate Guarantees for Traffic across ATM Network is to provide QoS. ATM is a connection-oriented switching technology, utilizing statistical multiplexing of fixed-length packets, known as cells. The purpose of traffic control is to minimize congestion[3][6]. In an ATM network when the Source Machine continuously sends cells to the Destination Machine through the Router Machine, there will be a possibility of occurring congestion. When congestion occurs the Routing Machine cannot accept more cells and hence these cells will be discarded. This causes regeneration and retransmission of the discarded ATM cells.

Problem Description

Existing System

In the existing system when transferring data from sender to receiver there may be chances of data loss. No user can be known whether the router is free after he sends data to the router. Also no intimation when the buffer at the router is full or busy or free [7]. It causes retransmission of data to the router and hence redundant bandwidth usage and consumption of time. The Limitations of Existing System: It causes retransmission of data to the router and it causes redundant bandwidth usage and consumption of time.

Proposed Systems

In this System, using several algorithms viz. Leaky bucket algorithm and Virtual scheduling algorithm congestion can be avoided. Leaky Bucket Algorithm continuously receives and sends the data at the particular interval of time to avoid congestion or flooding of data. Virtual scheduling algorithm monitors the congestion occurrence, when congestion occurs it intimates or warns user and simultaneously intimates when the router is free and data transferred successfully to destination. Some Advantages over Existing System: Avoids congestion or flooding of data, Saves time, Intimation about whether the router is busy or not.

System Analysis

After the problem is clearly understood and the solution is proposed the next step is to conduct feasibility study. The main objective of the feasibility study is to determine whether the proposed system, is feasible or not. Mainly there are three types of feasibility study to which proposed system is subjected as described below.

Technical Feasibility

Technical feasibility deals with the software and hardware requirements. The

hardware required is Pentium IV 1.8 GHz processor, 128 MB or higher RAM and sufficient disk storage capacity 40 GB. The software required is Java swings. If the requirements are made available with the system then the proposed system is said to be technically feasible.

Operation Feasibility

The system offers greater levels of operational feasibility. So the users need not have any experience to use it. The processing of system is very high and flexible.

Economic Feasibility

It is the most important aspect of feasibility study. The package is being developed is definitely feasible from economic point of view because the software and hardware requirements and the number of operating personnel required for the operation of these project is minimum to the greater extent.

Function Feasibility

Following is a list of functionalities to be supported by the Network management tool. More functionality can be added to this list. And, in places where the description of functionality is not adequate, you can make appropriate assumptions and proceed.

System Description

Network Management and Monitoring Tool is used for getting the system information of any system present in the network. The basic idea is that the Administrator does not need to go to the client machine to know the configuration details. This has been implemented using Client –Server architecture. Here the client program contains the code for getting connection to the server. Each system should have the client program to get connection to the server in the network. Now a days it is very essential to install a monitoring tool on the server in any sort of network, whether a corporate, or an institutional network, to maintain the network. This tool/application will keep track of all the employees/students presently working on the network [2] [5].It helps the server to take the snapshot of active clients screen at any point/instant of time. Its main objective is to build software that monitors all the system currently working in the network and ease the task of administrator as well as close the doors for the hacker.

System Modules

This project is mainly based on java Sockets that establish the connection between master and the slave system. The administrator/user will enter the IP Address of the system that he wants to work on. The next requirement is slave program must be running on the remote system in order to establish the connection [9][4]. This project is divided into five modules: 1.Finding open ports of the server, 2.finding all network devices, 3.Shared resource of the system and user using it, 4. Screenshot of the remote system, 5.Close all open ports. Network monitoring management, which will cover the following modules:

Finding Open ports of the server

In this module user will we enter the IP Address of the remote system as well as the range of port number to find all open ports. It work in the following ways: A. Firstly the Client sent the request to the server, and then socket class is designed so that it can be connected to the server socket and thus can initiate the protocol exchanges. B. Then the creation of the “SOCKET” object implicitly establishes a connection between the client and the server. C. Lastly through the use of Action Manager we can perform the required action as described in the “PORT” class.

Finding all network drives

This module will develop on the server side. JDBC connectivity will use to find all the system in the network. The database will contain some sort of possible IP Addresses of networking device. That is the system is connected or not in other words it is “UP” or “DOWN” respectively. This is what we have to check it first. So this module is going to work in the following ways: a. First of all we have to know the IP address range of all the computers which are going to be connected as the client to the server computer. b. After this we must check that the client part of the program should be in the running state at that point of time. c. After the connection is established through SOCKET(S,400 i.e. port number 4000 , and after setting the IP address range i.e. all the four bits of it then we go for the option “whole Network. d. Finding it will find that whether port number 4000 is open on a particular client or not. If it is open it will gave UP either DOWN.

Shared Resource of the system and user using it

This module will find shared resource of any system which is using it. It also gives information about open files of shared files and which system is using these resources. Basically in this module it is checking the system information i.e. its shared resources if any and its configuration. It works in the following ways :(1) First it checks the connectivity if found , we then have to give the IP address of any particular system.(2) It will access that client through its open port i.e port number 4000 and then displays its information using its inbuilt function and java like gel system info etc.

Scress shot of the remote system

This module will return the Screen Shot of the remote system. It works in the following ways (1) first of all it also demands IP address like other option available. (2) It will check the IP address is valid or not, if valid then it connect MASTER (server) to the SLAVE (client) through port number 4000. (3) It uses some inbuilt function and some user defined class and function to take the SNAPSHOT of that particular client. (4) Basically there are two classes SHOT {} and SHOT1 {} is going to be used by using the “Getoutputstream ()” and “getinputstream ()” and some functions like “repaint”. It redraws the desktop screen that can be either html page or word page or it may be anything at that point of time on that screen.

Close all open Ports

This module will try to close all the open ports of the remote system to prevent from the hackers. Basically after every user defined function is used and connection is

verified, and when that user defined function executes “dispose ()” function is called so that it closes all the open codes. Thus it prevents it from hackers.

System Implementation and overview of technologies used

Sockets

Sockets are a low-level programming interface for networked communications. They send streams of data between applications that may or may not be on the same host. Sockets originated in BSD UNIX. The socket APIs can be used with almost any kind of underlying network protocol[1]. Since the protocols that transport data across the network can have radically different features, the socket interface can be quite complex.

Java supports a simplified object-oriented interface to sockets that makes network communications considerably easier. Socket uses stream to send and receive data over the network. Streams provide a unified I/O interface; reading or writing across the Internet is similar to reading or writing a file on the local system. Java provides different kinds of sockets to support two distinct classes of underlying protocols. Java's Socket class uses a connection-oriented protocol.

A connection-oriented protocol gives you the equivalent of a telephone conversation; after establishing a connection, two applications can send data back and forth; the connection stays in place even when no one is sending any data. The protocol ensures that no data is lost and that it always arrives in order. While the Datagram Socket class, uses a connectionless protocol.

A connectionless protocol is more like the postal service. Applications can send short messages to each other, but no attempt is made to keep the connection open between messages, to keep the messages in order, or even to guarantee that they arrive. The Socket class speaks TCP, and the Datagram Socket class speaks UDP, both are standard Internet protocols. These protocols are available on any system that is connected to the Internet[8].

Client and Server

When talking about network applications, it's common to about clients and servers. The Client is the one which sends a request to the server. While the server is always in listen state for any incoming requests. The most important difference between a client and a server is that a client can create a socket to initiate a conversation with a server application at any time, while a server must prepare to listen for incoming conversations in advance. The java.net Socket class represents a single side of a socket connection on either the client or server. In addition, the server uses the java.net Server Socket class to wait for connections from clients. An application acting as a server creates a Server Socket object and waits, blocked in a call to its accept() method, until a connection arrives. When it does, the accept() method creates a Socket object the server uses to communicate with the client.

Sockets and Server sockets

Socket object for each client, as shown server carries on multiple conversations at once; there is only a single Server Socket, but one active in following figure 1.

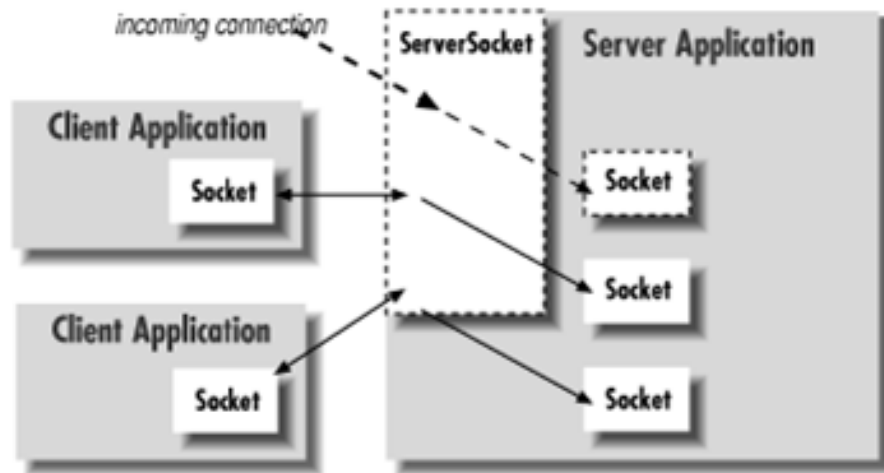


Figure 1: Socket and Server Socket

A client needs two pieces of information to locate and connect to another server on the Internet: a hostname (used to find the host's network address) and a port number. The port number is an identifier that differentiates between multiple clients or servers on the same host. A server application listens on a prearranged port while waiting for connections. Clients select the port number assigned to the service they want to access. Once the client is connected to the server on the specified port they then use separate I/O channels for sending and receiving information.

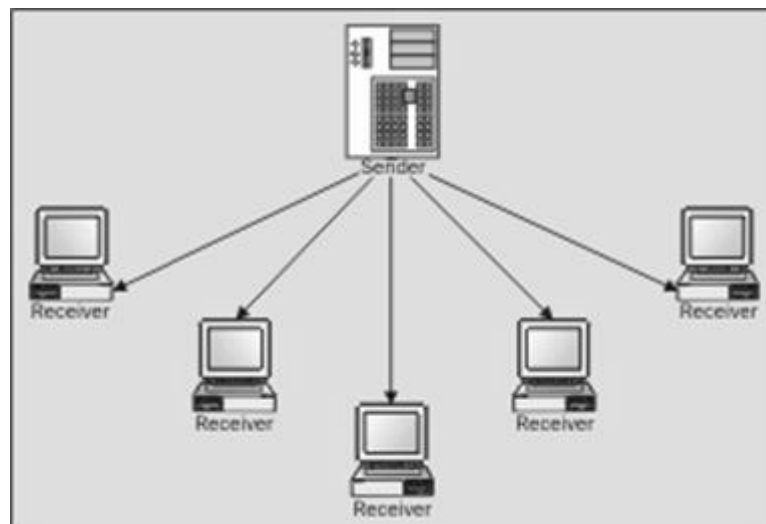


Figure 2: Network Connection

Conclusion and future enhancement

The package was designed in such a way that future modifications can be done easily.

The following conclusions can be deduced from the development of the project. Automation of the entire system improves the efficiency. It provides a friendly graphical user interface which proves to be better when compared to the existing system. It gives appropriate access to the authorized users depending on their permissions. It effectively overcomes the delay in communications. Updating of information becomes so easier. System security, data security and reliability are the striking features. The System has adequate scope for modification in future if it is necessary.

Here we provided a way for getting documents and sending documents, we will provide the way of chatting in this site so that they can chat here itself and view the document and make discussions here itself in convenient manner. The next enhancement that we can add is the chat server built in this site so that they can discuss about document here itself.

References

- [1] A Simple Network Management Protocol (SNMP)", RFC1157.
- [2] Author: O Reilly, C# COOK BOOK, O Reilly Publications, Page No 25-225Deshpande.
- [3] "On the Future of Internet Management Technologies", Jurgen Schonwalder, Aiko Pras, and Jean-Philippe Martin-Flatin, IEEE Communications Magazine, Oct 2003.
- [4] "A survey of the Criteria on selecting the suitable Network Management System", P. Chatzimisios, A. Vafiadis, P.Kamargiannis and D. Stoimenos, SCI 2003.
- [5] Performance Network Management Architecture", Andrea Silva Ramos, Anilton Salles Garcia, Rodolfo da Silva Villaca and Rodrigo Bonfa Drago.
- [6] "Network Management Basics", Cisco systems, http://www.cisco.com/univercd/cc/td/doc/cisintwk/ito_doc/nmbasics.htm .
- [7] "Network Management Tools", Mike Jude, Business Communications Review, May 2002.
- [8] "Heterogeneous Tools for Heterogeneous Network Management with WBEM", Kenneth Carey and Fergus O'Reilly, Adaptive Wireless System Group, Cork Institute of Technology.
- [9] "On Management Technologies and the Potential of Web Services", George Pavlou, Paris Flegkas, Stelios Gouveris, and Antonio Liotta, University of Surrey, IEEE Communications Magazine, July 2004.
- [10] "Aircraft Data Network (ADN) , Part I – *Systems concepts and Overview*",http://www.arinc.com/aec/draft_documents/664p1_d2s1.pdf.

