Chapter 1 Introduction

With the rapid growth of the Internet, millions of users are accessing the Internet at the same time. The flow of the internetwork traffic becomes heavier and heavier. This condition will cause intermediate routers to be congested. Under heavy congestion condition, senders may encounter two common problems:

- Packets delay and
- Packets drop

Packet delay will cause Round Trip Time (RTT) to expire. Besides intermediate routers congestion full receiver buffers will also cause incoming packets drop. This drop will cause transmission timeout, which require senders to retransmit the unacknowledged packet, this will affect the performance of the network. To reduce traffic congestion, a congestion control protocol is needed. Transmission Control Protocol is one of the Transport layer congestion control protocol.

Most current Internet communication uses the Transmission Control Protocol and Internetworking Protocol (TCP/IP) protocol suite. More than 36 million hosts are distributed over more than 134,000 networks. Parts of the backbone operate at 155 Mbps [BEHROUZ].

For wired networks, there are several versions of congestion control mechanisms. Four of the most common congestion control mechanisms are Tahoe, Reno, NewReno and Selective Acknowledgement (SACK) [RFC2581]. For Tahoe, Reno and NewReno, only the sender performs congestion control mechanism, whereas, for SACK, both sender and receiver perform congestion control mechanism. All of these mechanisms have different performance depending on how they make use of Fast Retransmit and Fast Recovery algorithm.

In order to understand their performance, there are two methods:

- Actual network system

Establish a real network system by connecting physical devices into the network system. Real data can be collected.

- Simulation

  Create a network simulation environment to simulate a real network situation.

To establish an actual network system, the study is costly. In addition, different network topologies and congestion scenarios are very difficult to be adapted. Simulation will solve the larger amount of cost investment. On the other hand, simulation only needs to have a set of digital computer to simulate an actual network environment. The simulation result can be collected in a short period of time compared to an actual system. Furthermore, with simulation a network topology can be adapted as needed easily.

In this thesis, the simulation method is used to compare the performance of Tahoe, Reno and NewReno. UM simulator, UMJaNetSim is selected for this purpose.

UMJaNetSim is already inbuilt with Tahoe and Reno but does not include any NewReno component. Part of this thesis is to develop NewReno implementation into the UMJaNetSim.

1.1 Introduction to Networking Technologies

Transmission Control Protocol (TCP) is a vital protocol suite, which is one of the layers in networking technology. The following section will describe an overview of OSI model; mapping TCP/IP model with OSI model; and TCP protocol suite.

1.1.1 Introduction of OSI Reference Model

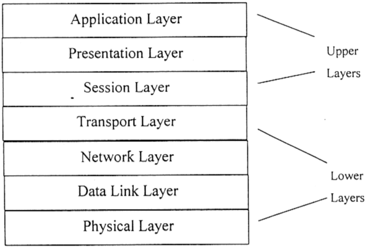

Figure 1.1 OSI Model

The OSI model is shown in Figure 1.1. The application, presentation, and session layers are all application-oriented in that they are responsible for presenting the application interface to the user. All three are independent of the layers below them and are totally oblivious to the means by which data gets to the application. These three layers are called the upper layers.

The lower four layers deal with the transmission of data, covering the packaging, routing, verification, and transmission of each data group. The lower layers do not worry about the type of data they receive or send to the application, but deal simply with the task of sending it. They do not differentiate between the different applications in any way.

The following sections explain each layer of the architecture of the OSI model.

1.1.1.1 The Application Layer

The application layer is the end-user interface to the OSI system. It is where the applications, such as electronic mail, USENET news readers, or database display modules, reside. The application layer's task is to display received information and send the user's new data to the lower layers.

In distributed applications, such as client/server systems, the application layer is where the client application resides. It communicates through the lower layers to the server.

1.1.1.2 The Presentation Layer

The presentation layer's task is to isolate the lower layers from the application's data format. It converts the data from the application into a common format, often called the *canonical representation.* The presentation layer processes machine-dependent data from the application layer into a machine-independent format for the lower layers.

The presentation layer is where file formats and even character formats (ASCII and EBCDIC, for example) are lost. The conversion from the application data format takes place through a "common network programming language" (as it is called in the OSI Reference Model documents) that has a structured format.

The presentation layer does the reverse for incoming data. It is converted from the common format into application-specific formats, based on the type of application the machine has instructions for. If the data comes in without reformatting instructions, the information might not be assembled in the correct manner for the user's application.

1.1.1.3 The Session Layer

The session layer organizes and synchronizes the exchange of data between application processes. It works with the application layer to provide simple data sets called *synchronization points* that let an application know how the transmission and reception of data are progressing. In simplified terms, the session layer can be thought of as a timing and flow control layer.

The session layer is involved in coordinating communications between different applications, letting each know the status of the other. The session layer handles an error in one application (whether on the same machine or across the country) to let the receiving application know that the error has occurred. The session layer can resynchronize applications that are currently connected to each other. This may be necessary when communications are temporarily interrupted, or when an error has occurred that resulted in loss of data.

1.1.1.4 The Transport Layer

The transport layer, as its name suggests, is designed to provide the "transparent transfer of data from a source end open system to a destination end open system," according to the OSI Reference Model. The transport layer establishes, maintains, and terminates communications between two machines.

The transport layer is responsible for ensuring that data sent matches the data received. This verification role is important in ensuring that data is correctly sent, with a resend if an error is detected. The transport layer manages the sending of data, determining its order and its priority.

1.1.1.5 The Network Layer

The network layer provides the physical routing of the data, determining the path between the machines. The network layer handles all these routing issues, relieving the higher layers from this issue.

The network layer examines the network topology to determine the best route to send a message, as well as figuring out relay systems. It is the only network layer that sends a message from source to target machine, managing other chunks of data that pass through the system on their way to another machine.

1.1.1.6 The Data Link Layer

The data link layer, according to the OSI reference paper, "provides for the control of the physical layer, and detects and possibly corrects errors that can occur." In practicality, the data link layer is responsible for correcting transmission errors induced during transmission (as opposed to errors in the application data itself, which are handled in the transport layer).

The data link layer is usually concerned with signal interference on the physical transmission media, whether through copper wire, fiber optic cable, or microwave. Interference is common, resulting from many sources, including cosmic rays and stray magnetic interference from other sources.

1.1.1.7 The Physical Layer

The physical layer is the lowest layer of the OSI model and deals with the "mechanical, electrical, functional, and procedural means" required for transmission of data, according to the OSI definition. This is really the wiring or other transmission form.

When the OSI model was being developed, a lot of concern dealt with the lower two layers, because they are, in most cases, inseparable. The real world treats the data link layer and the physical layer as one combined layer, but the formal OSI definition stipulates different purposes for each.

1.1.2 Mapping TCP/IP model with OSI Model

This section will map the TCP/IP model with OSI model. The OSI and TCP/IP layered structures are shown in Figure 1.2.

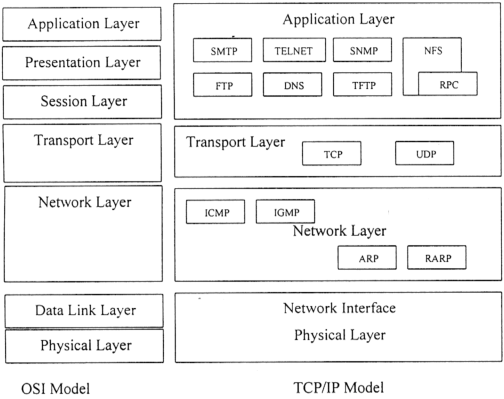

Figure 1.2 The OSI and TCP/IP layered structures

1.1.2.1 Physical and Data Link Layers

At the physical and data link layers, TCP/IP does not define any specific protocol. It supports all of the standard and proprietary protocols. X network in a TCP/IP internetwork can be a local area network (LAN), a metropolitan area network (MAN), or a wide area network (WAN).

1.1.2.2 Network Layer

At the network layer (or, more accurately, the internetwork layer), TCP/IP supports the Internetworking Protocol (IP). IP contains four supporting protocols: Address Resolution Protocol (ARP), Reverse Address Resolution Protocol (RARP), Internet Control Message Protocol (ICMP), and Internet Group Message Protocol (IGMP).

1.1.2.3 Transport Layer

The transport layer is represented in TCP/IP by two protocols: Transmission Control Protocol (TCP) and User Datagram Protocol (UDP). The IP is a host-to-host protocol, meaning it can deliver a packet from one physical device to another. UDP and TCP are transport level protocols responsible for delivery of a message from a process (running program) to another process.

1.1.2.4 Application Layer

The application layer in TCP/IP is equivalent to a combined session, presentation, and application layer in the OSI Model.

The Telnet program provides a remote login capability. This lets a user on one machine log onto another machine and act as though he or she were directly in front of the second machine. The connection can be anywhere on the local network or on another network anywhere in the world, as long as the user has permission to log onto the remote system.

You can use Telnet when you need to perform actions on a machine across the country. This isn't often done except in a LAN or WAN context, but a few systems accessible through the Internet allow Telnet sessions while users play around with a new application or operating system.

File Transfer Protocol (FTP) enables a file on one system to be copied to another system. The user doesn't actually log in as a full user to the machine he or she wants to access, as with Telnet, but instead uses the FTP program to enable access. Again, the correct permissions are necessary to provide access to the files.

Once the connection to a remote machine has been established, FTP enables you to copy one or more files to your machine. (The term *transfer* implies that the file is moved from one system to another but the original is not affected. Files are copied.) FTP is a widely used service on the Internet, as well as on many large LANs and WANs.

Simple Mail Transfer Protocol (SMTP) is used for transferring electronic mail. SMTP is completely transparent to the user. Behind the scenes, SMTP connects to remote machines and transfers mail messages much like FTP transfers files. Users are almost never aware of SMTP working, and few system administrators have to bother with it. SMTP is a mostly trouble-free protocol and is in very wide use.

Domain Name System (DNS) enables a computer with a common name to be converted to a special network address. For example, a PC called Darkstar cannot be accessed by another machine on the same network (or any other connected network) unless some method of checking the local machine name and replacing the name with the machine's hardware address is available. DNS provides a conversion from the common local name to the unique physical address of the device's network connection.

Simple Network Management Protocol (SNMP) provides status messages and problem reports across a network to an administrator. SNMP uses User Datagram Protocol (UDP) as a transport mechanism. SNMP employs slightly different terms from TCP/IP, working with managers and agents instead of clients and servers (although they mean essentially the same thing). An agent provides information about a device, whereas a manager communicates across a network with agents.

Network File System (NFS) is a set of protocols developed by Sun Microsystems to enable multiple machines to access each other's directories transparently. They accomplish this by using a distributed file system scheme. NFS systems are common in large corporate environments, especially those that use UNIX workstations.

The Remote Procedure Call (RPC) protocol is a set of functions that enable an application to communicate with another machine (the server). It provides for programming functions, return codes, and predefined variables to support distributed computing.

Trivial File Transfer Protocol (TFTP) is a very simple, unsophisticated file transfer protocol that lacks security. It uses UDP as a transport. TFTP performs the same task as FTP, but uses a different transport protocol.

1.1.3 Introduction to Transmission Control Protocol

Transmission Control Protocol (TCP) is a protocol suite. TCP supports various types of application programs, which are located at Application Layer - the layer above TCP. Example of application programs are File Transfer Protocol (FTP), Trivial File Transfer Protocol (TFTP), Domain Name System (DNS), TELNET, Simple Mail Transfer Protocol (SMTP), Hypertext Transfer Protocol (HTTP), Simple Network Management Protocol (SNMP), BOOTP and DHCP.

For local client programs and remote server programs, TCP differentiates each program by assigning each program with a number, which is a port number. With this port number, an application program is able to know which process the arriving packets belong to.

In the TCP/IP model, TCP is located at the Transport Layer, i.e. the layer above Network Layer. TCP also services the Network Layer, which is the Internet Protocol (IP) Layer. Basically, TCP provides services that are not provided by the layers below. Among the services provided are segmentation and reassembly packet, flow control, error control and congestion control.

TCP is a connection-oriented network. Connection-oriented network means TCP needs to establish a network path between its sender and its receiver before data transfer and acknowledge reply. This means that TCP is reliable.

TCP congestion control consists of various types of TCP versions like Tahoe, Reno, NewReno and Selective Acknowledgement (SACK). The inter-network structure is an end-to-end wired network. Each TCP version uses one or more types of algorithm, such as Slow Start, Congestion Avoidance, Fast Transmit and Fast Recovery.

A more detailed explanation will be given in chapter two.

1.2 Introduction to Simulation

Computer simulation is a discipline of designing a model of an actual or theoretical physical system, executing the model on a digital computer, and analyzing the execution output [FISHWICK96]. Within a simulated system, simulation models can be used for testing as well as for researching discrete and dynamic change entities. This model can imitate an actual system.

A simulated system or simulation is cheaper than an actual system in terms of cost and adaptation on a network model. In simulations, it is also easy to change the system's properties and functionality if researchers intend to test and verify the system. By using simulations, events can be run within minutes. Besides that researchers can plan and design a network simulation based on a real time topology and perform analysis of varying traffic types and loads, network capacity planning, traffic aggregation and other experiments.

A more detailed look at network simulator will be given in chapter three.

1.3 Objective of The Thesis

This thesis undertakes a detailed study of TCPIP architecture, with an emphasis on creating a simulated environment for the testing and evaluation of TCP/IP topologies. The following lists several primary objectives for this project:

- To study and explore TCP versions of Tahoe, Reno and NewReno.
- To study, understand and differentiate the capabilities of Tahoe, Reno and NewReno.
- To study simulation techniques and approaches.
- To implement and develop a NewReno component on TCP with IP architecture.

1.4 Scope of The Thesis

The fundamental scope of this thesis is to integrate the NewReno congestion control component inside the existing network simulator, UMJaNetSim. The following statements summarize the scope of this thesis in accordance with the stated objectives:

- The study and survey of TCP congestion control version including Tahoe, Reno and NewReno.
- To integrate NewReno into the UM developed simulator, UMJaNetSim v 0.5.
- To create a simulated environment that allows testing and experimentation of Tahoe, Reno and NewReno simulations
- To create a user friendly graphical user interface (GUI) TCP with IP network simulator.

1.5 Report Organizations

This report has a total of 6 chapters. It is organized as following:

Chapter 1 provides the overview of current network technologies, introduction of the OSI Reference model and mapping the TCP/IP model with the OSI model. This chapter also defines the introduction of the thesis, including objectives and scope of the thesis.

Chapter 2 provides an in-depth discussion on current Transmission Control Protocol congestion control version such as Tahoe, Reno, NewReno and Selective Acknowledgement. This chapter will first introduce Transmission Control Protocol. Then discuss congestion control algorithms such as Slow Start, Congestion Avoidance, Fast Retransmit and Fast Recovery algorithm. This chapter will discuss the few important Transmission Control Protocol terminology. The remaining section of this chapter will discuss how Tahoe, Reno, NewReno and Selective Acknowledgement works with congestion control algorithm.

Chapter 3 provides a study on various network simulators available. The advantages and disadvantages of each simulator are discussed and analyzed carefully. This chapter also discusses the various programming approaches that can be used and concludes with a discussion on the choice of programming language selected.

Chapter 4 covers the architecture, Application Programming Interface (API) and features of the UM developed network simulator, UMJaNetSim v0.5.

Chapter 5 covers the design aspects of the TCPIP simulator component. This includes the system architecture design, the class design and the system topology design. The functionality and the attributes of each object classes are included in this chapter. It also covers the implementation details of the NewReno component for the TCP/IP simulator and different types of testing done on the simulator. There are performance analysis on Tahoe, Reno and NewReno. Performance evaluation for the same TCP version in the TCP/IP topology is conducted.

Chapter 6 summarizes the research on the development of the TCP/IP simulator. It summarizes the findings of this project and the outcome of the development process.