# INTERNET COMMUNICATION THROUGH POINT-TO-POINT PROTOCOL ON LINUX OPERATING SYSTEM

ABDOULIE F. BADJIE

UNIVERSITI TEKNOLOGI MALAYSIA

# INTERNET COMMUNICATION THROUGH POINT-TO-POINT PROTOCOL ON LINUX OPERATING SYSTEM

ABDOULIE F. BADJIE

A thesis submitted in fulfilment of the
requirements for the award of the degree of
Master of Engineering (Electrical-Electronics and Telecommunications)

Faculty of Electrical Engineering
Universiti Teknologi Malaysia

MARCH 2005

This project is dedicated to my beloved parents, brothers, sisters and friends.

In addition, my thanks go out to the Islamic Development Bank for providing me the opportunity to study this program. Thanks are also extended to the Gambia Civil Aviation Authority for giving me the study leave to pursue this program. Last but not the least a lot of thanks is extended to all my friends at the University Technology Malaysia especially KRP G29 Hostel.

## ACKNOWLEDGEMENT

In preparing this thesis, I got a lot of feedback from different people. First I wish to express my sincere appreciation to my main thesis supervisor, Associate Professor Dr. Norsheila bt Fisal, for the encouragement, guidance, critics and advice she give to me to complete this project.

I am also indebted to Mr Adel Ali of the Telecoms Lab for lots of advice and guidance in the preparation of the project. Thanks also to Mr Muladi and all students of the same lab.

# ABSTRACT

The growth of the internet has meant that different devices are connected together and data has to be exchanged from one place to another. Since the internet consists of many networks connected together using different protocols, there is the need for a single independent protocol which can transport the data from one end to another irrespective of the type of protocol of the origin or destination of the connection. The Point-to-Point Protocol is such a protocol and is designed for simple links which transport packets between two peers. These links provide full-duplex simultaneous bi-directional operation, and are assumed to deliver packets in order. The use of sockets in the internet communication eliminates the reconfiguration of the link between the server and client. It also enables multiple access of a server by many clients. This project demonstrates the processes that's takes place when accessing a website or doing some client/server computing in the form of a link connecting two computers. The computers communicate through the use of sockets in Point-to-Point Protocol as the Media Access Control in the Data link Layer. Data communication takes place in the form of a Graphic User Interface using text and audio through the two computers.

# ABSTRAK

Perkembangan internet juga bermakna pelbagai peranti yang berbeza telah diintegrasikan bersama dan data boleh ditukar daripada suatu tempat ke tempat yang lain. Semenjak internat mengandungi banyak rangkaian yang tersambung bersama dengan menggunakan protocol berbeza, suatu protokol bebas tunggal diperlukan dimana ianya boleh membawa data daripada suatu titik akhir ke suatu yang lain berdasarkan jenis protokol asal atau destinasi sambungan. Protokol titik-ke-titik merupakan suatu protokol yang telah direkabentuk untuk hubungan mudah yang membawa paket diantara dua komputer. Hubungan ini menyediakan operasi dupleks-penuh serentak dua-arah, dan diandaikan untuk menghantar paket berdasarkan arahan. Penggunaan soket dalam komunikasi internat menghapuskan pengkonfigurasian semula hubungan diantara pelayan dan pelanggan. La juga membolehkan pelbagai akses pelayan oleh ramai pelanggan. Projek ini telah mendemonstrasikan proses yang digunakan apabila pengaksesan laman web atau sebahagian perkomputeran pelanggan/pelayan dalam format hubungan sambungan dua komputer. Komputer berkomunikasi melalui penggunaan soket dalam Protokol Titik-ke-Titik sabagai Kawalan Akses Media dalam Lapisan Hubungan Data. Komunikasi data mengambil tempat dalam format Pengantaramuka Grafik Pengguna dengan menggunakan teks dan audio melalui dua komputer

# TABLE OF CONTENTS

## LIST OF TABLES

## LIST OF FIGURES

## LIST OF ABBREVIATIONS

| | |
|---|---|
| AF_INET | Open Transport Reference |
| ATM | Asynchronous Transfer Mode |
| ARP | Address Resolution Protocol |
| BGP | Border Gateway Protocol |
| BSD | Berkeley Software Distribution |
| CDDI | Copper Distributed Data Interface |
| CLNP | Connectionless Network Protocol |
| CRC | Cyclic Redundancy Check |
| CSMA/CD | Carrier Sense Multiple Access/Collision Detection |
| CTS | Clear To Sense |
| DCD | DCD (Data Carrier Detect |
| DHCP | Dynamic Host Configuration Protocol |
| DLL | Data Link Layer |
| DNS | Domain Name Server |
| DSL | Digital Subscriber Line |
| DSR | Data Set Ready |
| DTR | Data Terminal Ready Data |
| FDDI | Fiber Distributed Data Interface |
| FTP | File Transfer Protocol |
| GCC | GNU C Compiler |
| GUI | Graphical User Interface |

| | |
|---|---|
| HDLC | High Level Data Link Control |
| HomePNA | Home Phoneline Networking Alliance |
| HTTP | HyperText Transfer Protocol |
| ICMP | Internet Control Message Protocol |
| IDP | Intrusion Detection and Prevention |
| IEEE | Institute of Electrical and Electronics Engineers |
| I/O | Input/ Output |
| IP | Internet Protocol |
| IPC | Interprocess Communication |
| IPX | Internetwork Packet eXchange |
| IRQ | Interrupt Request |
| ISDN | Integrated Services Digital Network |
| ISP | Internet Service Provider |
| LAN | Local Area Network |
| LCP | Link Control Protocol |
| LLC | Logical Link Control |
| MAC | Media Access Control |
| NCP | Network Control Protocol |
| NIC | Network Interface Card |
| OSI | Open System Interconnect |
| PEX | Xerox's Packet Exchange Protocol |
| PPP | Point-to-Point protocol |
| PPPoA | PPP over ATM |
| PPPoE | PPP over Ethernet |
| RFC | Requests for Comments |
| RIP | Routing Information Protocol |
| RTS | Ready To Sense |
| SLIP | Serial Line Internet Protocol |
| SMTP | Simple Mail Transfer Protocol |
| SNMP | Simple Network Management Protocol |
| SONET | Synchronous Optical NETwork |
| SPP | Standard Parallel Port |
| Tcl | Tool command language |
| TCP | Transmission Control Protocol |

| | |
|---|---|
| Tk | Toolkit |
| TTL | Time To Live |
| UART | Universal Asynchronous Receiver/ Transmitter |
| UDP | User Datagram Protocol |
| UUCP | Unix-to-Unix CoPy |
| XNS | Xerox Network Systems |
| Wi- Fi | Wireless Fidelity |
| WAN | Wide Area Network |
| WWW | World Wide Web |

# CHAPTER 1

# INTRODUCTION

## 1.1 Background

Before the advent of the Internet, the only way to connect to the outside world using a modem was to talk directly to the modem. This was a confusing and complex mess. For an idea of all the settings one had to have, just look at a serial control. Though there was a lot one could do with a modem, there were still a few things that were impossible, like being connected to two places at once.

The internet changed all that. Now, any person can have multiple connections open at the same time. And as an extra bonus, one does not have to fiddle with all those cryptic settings. Simply transmitting data is pointless though. One have to send it in some type of format in the form of 0s and 1s which are then transported through the physical layer such as an RS 232 by using serial cable, through wireless connection or through the telephone cable using an Ethernet card.

TCP/IP (Transmission Control Protocol/ Internet Protocol) is the basic communication language or protocol of the internet. It can also be used as a communications protocol in a private network (either as intranet or an extranet).

When you are set up with direct access to the internet, your computer is provided with a copy of the TCP/IP program just as every other computer that one may send messages or get information, also has a copy of TCP/IP.

TCP/IP is a two layer protocol. The higher layer, Transmission Control Protocol manages the assembling of a message or file into smaller packets that are transmitted over the internet and received by a TCP layer that reassembles the packets into the original message. The lower layer, Internet Protocol, handles the address part of each packet so that it gets to the right destination. Each gateway computer on the network checks this address to see where to forward the message. Even though some packets from the same message are routed differently than others, they will be reassembled at the destination.

TCP/IP uses the client/server model of communication in which a computer user (a client) requests and is provided a service (such as sending a web page) by another computer (a server) in the network. TCP/IP communication is primarily point- to-point, meaning each communication is from one point (or host computer) in the network to another point or host computer.

TCP/IP and the higher-level applications that use it are collectively said to be stateless because each client request is considered a new request unrelated to any previous one (unlike ordinary phone conversations that require a dedicated connection for the call duration). Being stateless frees network paths so that everyone can use them continuously. For TCP layer the connection remains in place until all packets in a message have been received [7].

Personal computer users usually get to the internet through the Serial Line Internet Protocol (SLIP) or the Point-to-Point Protocol (PPP).  These protocols encapsulate the IP packets so that they can be sent over a dial-up phone connection to an access provider’s modem. PPP is part of the data link layer of the OSI or TCP/IP model.

TCP is a connection-oriented transport protocol that sends data as an unstructured stream of bytes. By using sequence numbers and acknowledgment messages, TCP can provide a sending node with delivery information about packets transmitted to a destination node. Where data has been lost in transit from source to destination, it can retransmit the data until either a timeout condition is reached or until successful delivery has been achieved. It can also recognize duplicate messages and will discard them appropriately. If the sending computer is transmitting too fast for the receiving computer, it can employ flow control mechanisms to slow data transfer. TCP can also communicate delivery information to the upper-layer protocols and applications it supports.

The use of sockets in TCP/IP communication enables the creation of many clients accessing the server at the same time. The socket layer provides an application with a programming interface to the network that looks like a file. When an application writes to the socket, the socket layer sends data to an application on a remote host. When an application reads from a socket, the socket layer provides data received from a remote host.

The server application is the most important part of a protocol. It will do most of the work, and has to be in near-constant communication with multiple people. Sockets can either listen or connect. If a socket is listening, then it waits for other sockets to connect to it. If they're not listening, then they can connect to any other server that is listening. For a server naturally, all the sockets should be listening. That way, when a client connects, they can respond correctly.

## 1.2 Problem Statement

The Point-to-Point Protocol (PPP) originally emerged as an encapsulation protocol for transporting IP traffic over point-to-point links. PPP also established a standard for the assignment and management of IP addresses asynchronous (start/stop) and bit-oriented synchronous encapsulation, network protocol multiplexing, link configuration, link quality testing, error detection, and option negotiation for such capabilities as network layer address negotiation and data-compression negotiation. PPP supports these functions and is use more effectively in dialup networking, for example, say home computer to ISP connection for internet access by using Ethernet as the Network Interface Card. So to use in a two way direct cable RS232 connection one has to think of the speed disparity and the method of connecting the serial cable to the Ethernet card in the server for internet access. Also when data is traveling on serial communication lines, it can happen that data arrives faster than a computer can handle it (the computer may be busy doing something else, as, Linux is a multi-user, multi- tasking operating system). In order to ensure that data is not lost (data does not over run in the input buffer and hence get lost), some method of controlling the flow of data is necessary. There are two ways of doing this; either by using hardware signals (Clear To Send/Request to Send - CTS/RTS) or by using software signals (control S and control Q, also known as XON/XOFF). Whilst the later may be fine for a terminal (text) link, data on a PPP link uses all 8 bits - and it is quite probable that somewhere in the data there will be data bytes that translate as control S and control Q. So, if a modem is set up to use software flow control, things can rapidly go berserk. For high speed links using PPP (which uses 8 bits of data) hardware flow control is vital and it is for this reason that we must use hardware flow control.

## 1.3 Objectives

The objectives of the thesis are:

- To configure the Linux kernel for Point-to-Point Protocol communication
- To create a client/server TCP/IP communication through Point-to-Point Protocol link.
- To augment the TCP/IP link with a Graphical User Interface at both ends.

The Linux kernel will be configured to allow for PPP protocol communication with the various parameters specifying RS232 physical media. For data communication services a GUI is developed with TCL/TK programming language so that text messages can be exchanged from client to server. Audio files will also be sent through the link.

## 1.4 Scope

The scope of research will include establishing TCP/IP client/server communication through PPP for data communication. As a client/ server network, the communication will be bidirectional, i.e. any computer can be server or client. Using socket programming to write client and server side of the network. A GUI is created as the application layer for text and audio communication. The internet can be accessed through serial port from client with the server acting as the gateway by using Mozilla or any Linux compatible web browser.

## 1.5 Importance of Research

This project utilizes the Point-to-Point Protocol to communicate through the serial port to access the internet. This will be done in the form of client/server by using a GUI as the application layer. An internet access is done easily by everyone but very few know about sockets and how they operate, for example when a client queries a server to access server resources. Also the creation of the GUI and the different parameters to include in the configuration of the kernel makes it more interesting in knowing the principles of TCP/IP networking.

## 1.6 Research Methodology

The project is divided into two phases;

- **Configuring the Linux kernel**

In this part, the commands are written at the Linux prompt to get the desired parameters like speed at which data communication will take place, hardware flow control and how long the line should stay trying to connect if it fails in the initial connection phase. Also the IP addresses are assigned to both the server and client in the kernel configuration. Testing is done to see whether the connection is setup finally before any data communication takes place. Also the internet can be accessed through the client via the RS232 connector

- **Programming for Data Communication**

In this phase, the socket programs for client and server are written by using C language. The GUI is developed using TCL/TK programming language. Finally data communication services are provided through a client and server. Testing is also done with audio files across the connection.

## 1.7 Thesis Organization

This thesis consists of five chapters. Chapter 1 serves as an introduction to the thesis. It covers topics such as problem statement, objective of the research, scope of the project and the importance of the project.

Chapter 2 provides the relevant background for Point-to-Point Protocol, Data link Layer Services, How PPP negotiate for a line, Socket Programming, principles of Serial Communication.

Chapter 3 describes the implementation process which includes how to configure the network interface for PPP communication, server and client programming with sockets, and how to develop the GUI with TCL/TK language.

Chapter 4 describes the results and analysis of the project.

Finally, Chapter 5 concludes the thesis with a summary of the work that has been done, along with suggestions for future work.

# REFERENCES

1 S. Heath, "Essential *Linux," Boston Digital Press,* 1998

2. N. Wells and M. Lang," *Guide to Linux Installation and Administration",* Australia:Course Technology: 2000

3 M. Kofler, " *Linux: Configuration, Installation and Use",* Harlow, Eng: Addison-Wesley:1997

4 B. Brent, " *Practical Programming in Tcl & Tk ",* Prentice Hall:1997

5 A. S. Tanebaum, " *Computer Networks",* Pearson Education International, 2000

6 J Carlson, " *PPP Design and Debugging"*, Addison Wesley:1998

7 C. Metz, " *A pointed Look at the Point-to-Point Protocol",* IEEE Internet Computing:1999

8 J Romkey, "RFC 1055 - *Non-standard for transmission of IP datagrams over serial lines: SLIP*," IETF Network Working Group:1998

9 A. Chame, "*Integrated Processors in Internetworking Applications",* IEEE: 1*996*

10 C.L-Williams and J Drake "*Linux PPP HOWTO*," Commandpromt Inc:2000

11 C Hunt *," TCP/IP Network Administration",* Oreilly 1997

12 Cisco Documentation, " *Point to Point Protocol",* Cisco Systems:2002

13 G. McMillan, *" Socket Programming HOWTO",* http://www.amk.ca/python/howto/sockets/:1997

14 C M. Kozierok, “ *The TCP/IP Guide”,* http:www.tcpipguide.com:2004

15 http://www.sanpeople.com/ support/etherpad_ppp.asp

16 http://www.serial-networking.fsnet.co.uk/linuxroute.htm

17 V.S.Sastry, “ *GUI With Tcl/Tk”,* http://www.agocg.ac.uk/gv/issue40/gui.htm:1995

18 http://www.tcl.tk/doc/tclHistory.html

19 http://www.actiontecsupport.com/files/pm560lki.pdf