

An Internet Based Communication Server for Remote Control

*Zun-Po Huang, Wen-Shyang Hwang, and Jimmin Wang**

Computer Network Protocol and Applications Laboratory

Department of Electrical Engineering, National Kaohsiung University of Applied Sciences

**Kinghold Technology Co., Ltd.*

Email: adrian@wshlab2.ee.kuas.edu.tw

Abstract

Today, communication has been applied to everywhere; most devices are equipped with the communication capacity. However, some available older version devices are not provided with the long distance communication functions. This paper introduces a multi-port communication server to solve the problem. This server is able to provide an effective multipoint-to-multipoint communication service. The implementation of such a communication server, based on a low cost with multi-port, is presented. It will describe the driver design of communication server and the interface with the upper-layer access protocols such as PPP, SNMP in Linux. In addition, the server also provided the Web interface and NAT mechanism; hence the administrator can manage his/her remote equipments easily via Internet. This paper will describe the detail about the implementation.

Keywords: *Communication Server, Web Interface, PPP, Remote Control*

1 Introduction

Communication technologies are everywhere now; include wireless (e.g. IEEE 802.11, Bluetooth,

infrared rays etc.) and wire (e.g. Ethernet, USB, RS-232, IEEE 488 etc.). The equipments without the long distance communication function are also required to communicate via Internet for remote control today. The most communication technology of these equipments is the traditional RS-232. It is worth to investigate how to upgrade the communication capacity of these equipments to communicate each other, and how to perform the remote control via Internet.

Equipments communicate in the Internet have to obtain IP addresses first, however the version 4 IP address is insufficient now. There are two solutions to solve this problem now. One is IPv6. Another is NAT (Network Address Translation). Before the version 6 IP address is popular, the NAT technique should be the candidate. This paper will describe the driver implementation of the multi-port communication server implemented by Kinghold Technology Co. in 2001. The server is an industry computer with multiple RS-232 ports, and prepares to support the remote control via Web interface, and an intelligent handling function. The intelligent handling function is used to monitor data and deal with events immediately through the default Knowledge base on database, and then notify administrator to handle it further.

2 Communication server hardware

In this section, we will introduce the hardware of the communication server, it can be divided to three main parts: Multi-port card, communication server drive card, and communication server host. Figure 1 shows a Multi-Port card, which main function is used to receive and transmit data from/to remote equipments. The connector of Multi-Port card is Registered Jack-45 (RJ-45) and it will be converted to RS-232 serial port standard. There is an interface that is shown as a red oval drawing in Figure 1, it is used to connect with the communication server drive card as shown in Figure 2 We have developed the driver for this card (Figure 2, ISA slot), which main function is used to drive the communication server in Linux. Figure 3 shows the communication server host that can provide sixteen equipments to connect with.

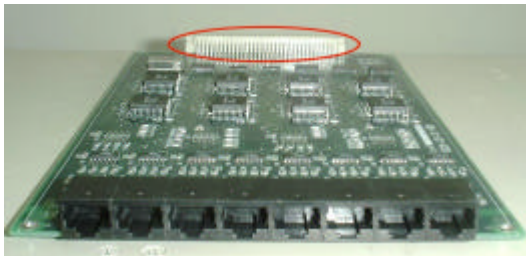


Figure 1. Multi-Port Card



Figure 2. Communication Server Drive Card



Figure 3. Communication server host

3 System Architecture

The system architecture of the communication server is shown as in Figure 4. A tty driver is developed for the communication server first. At that time, we use software named “mgetty” to listen to signals from ports of the communication server in order to aware whether any equipment is dialing to the communication server. While the dialing tone is arrival, Mgetty will response the phone call and then execute the handshake process between the communication server and the equipment (DTE). After handshaking, the control authority of mgetty would hand over pppd (PPP Daemon). The job of pppd is used to transmit or receive PPP packets and configures the options such as MTU/MRU (Max. Transmission/Receive Unit) at both sides. As long as DTE connect to DSE (Data Switching Equipment), administrator can obtain the information about utilization of communication server via Internet immediately. It is worth to be motioned. In our experience, the communications by using modem and null modem are all tested. There are many situations for DSE directly connecting with DTE by using RS-232 without modem, the technique is named as “Null modem.”

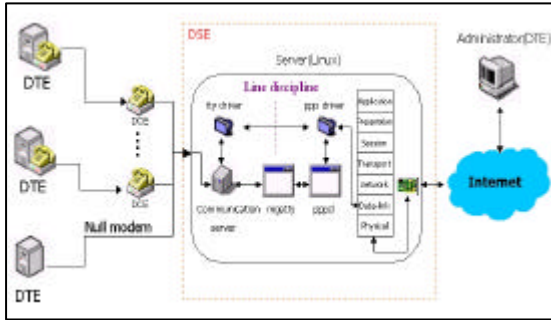


Figure 4. System architecture

The procedure of DSE operations is shown as in Figure 5. The frame from the remote equipment will be received by the multi-port card driver, and inserted into the corresponding received queue. The tty driver would remove frames from the received queues, and then passes these frames to the PPP module. The PPP module inspects these frames to subdivide into two kinds of frame formats. One is the PPP control frame format that is used to configure the options such as MTU, MRU at both sides. While the PPP module receives the PPP control frame, it will pass these frames to pppd (at user mode) in order to negotiate with the parameters that both sides would like to configure. Another frame format is the data packet in the network layer such as IPv4, IPv6, and IPX. When the PPP module receives these network layer packets, it would pass them to the corresponding process on Linux kernel. Finally, the NAT mechanism would update their packet headers, and transmit them to Internet.

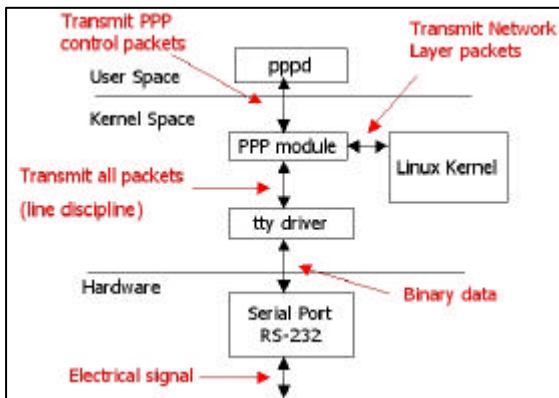


Figure 5. DSE Operations

4 Implementation results

While DTE connects to DSE, we use the software named BIEW (Binary view) [10] to view the binary file (/proc/kcore) on communication server for finding out what data buffer temporarily at that time. BIEW is an advanced file viewer with built-in editor for the binary, hexadecimal and disassembler modes. The transmitted or received PPP frames at both sides are monitored by BIEW as shown in Figure 6. In the figure, a red shape (0x7e, 0xff, .0xc6 0x7e) is marked presents that the DSE is transmitting a PPP frame that is an IPCP (Internet Protocol Control Protocol) to the remote DTE. By using BIEW, we can accurately verify the communication between DSE and DTE and the tty driver for communication server whether successful to work.

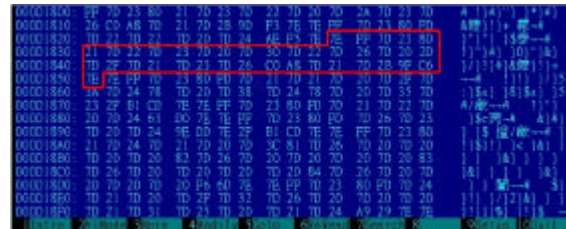


Figure 6. Viewing data buffer on communication server

In this implementation, the PPP frame is the major data format. The PPP frame format [7] is shown as in Figure 7. The figure does not include the start and stop bits of asynchronous links, nor any bits or octets inserted for transparency. The fields are transmitted from left to right. The Flag field is a single octet and indicates the beginning or end of a frame (named frame separator). The Flag field consists of the hexadecimal code 0x7e. There is only one Flag is required between two frames. The

Address field is a single octet and contains the hexadecimal code 0xff, it is an *All-Stations address*. PPP does not assign any individual station address. The All-Stations address means this frame will always be received. The Control field is a single octet and contains the hexadecimal code 0x03.

The Protocol field is two octets long, and its value identifies the protocol that encapsulated in the Information field of the frame. For example, the protocol field that contains the hexadecimal code 0x8021 is to indicate an IPCP encapsulated in the Information field of the frame. It could find them in Figure 6 about the sixth and seventh character from the place we marked with the red curve. There is a little difference of the PPP frames between in Figure 6 and in Figure 7, because of the procedure Octet-stuffed [8].

Flag	Address	Control	Protocol	Information	FCS	Flag
0x7e	0xff	0x03	16 bits		16 bits	0x7e

Figure 7. PPP frame format

Since the hardware is a novel design, the original serial port device will work abnormally. A new device for the communication server is written; the figure 6 shows all communication is successful. In addition, we built a Web Server Administration System shown as in Figure 8 to get the information which administrator would like to care about. In Figure 8, we could click “Administrator Login” to login “Administration Menu” as shown in Figure 9. There are five options can be chosen. These options are follows:

A. The “SerialPort options” presents that the administrator would like to configure options such as MTU, MRU etc. for serial ports as shown in Figure 10. It shows the administrator

could configure the existence serial port (rs1 or ttyS1 in Figure 10) or create a new serial port if needed.

- B. The “Traffic Analysis” presents that the administrator could get how much traffic on the communication server now. After clicking the “Traffic Analysis” option, the sub-menu page is as shown in Figure 11. While the “Total Traffic Analysis” option is chosen, the statistic traffic page is as shown in Figure 12. It presents a traffic analysis on a specify day that the administrator could choose over his own will. As this example, we choose the traffic analysis chart on June 13, 2002.
- C. The “On-Line equipments” function is use to know how many equipments are connected to the communication server now.
- D. The “I.D. for specific equipments” function presents that equipment must get a user name and a password if it would make a phone call to communication server by using Dial-up.
- E. The “Add a I.D. for specific equipments” function presents the administrator would add an I.D. for specific equipment.

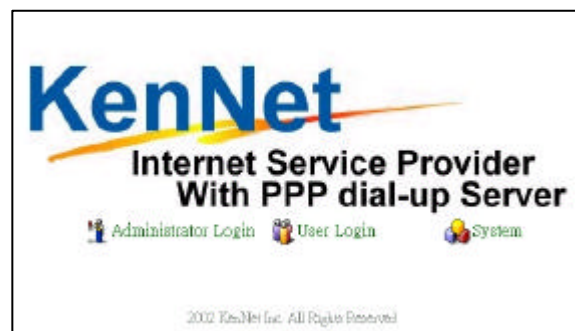


Figure 8. Web Server Administration System



Figure 9. Administration Menu



Figure 10. Configure Serial Port Menu



Figure 11. Traffic Analysis Menu

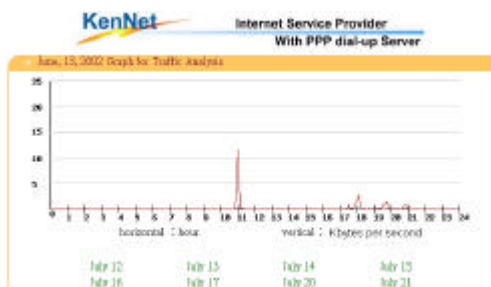


Figure 12. Graph for Traffic Analysis

5 Conclusion and future work

In this paper, we propose a communication server with multi-ports to connect multiple remote equipments simultaneously. A novel tty driver is written on Linux. This driver not only supports each port communication between DSE and DTE but also redirects data from PPP to Ethernet. Therefore, the administrator could configure the relative options for equipments and obtain the information about the utilization of communication sever immediately

through the web server that have built by using PHP and MySQL.

In the future, more SNMP (Simple Network Management Protocol) functions, is an Internet-standard protocol for managing each node on an IP network, will be implemented in the communication server to realize an “intelligent” communication server

References

- [1] Marino, P.; Poza, F.; Nogueira, J. “Industrial LAN’ s with real-time communication servers”, IEEE International Workshop, pp.119-127, 1997
- [2] Jing Bai, Bingyi Hu, Yonghong Zhang, and Datian Ye, “A Communication Server for Telemedicine Applications”, Information Technology in Biomedicine, IEEE Transactions, Volume: 1 Issue: 3, pp. 205-209, Sept. 1997
- [3] Geoffrey C. Sauerborn, “The distributed interactive simulation (DIS) lethality communication server”, Distributed Interactive Simulation and Real-Time Applications, pp.82-85, 1998
- [4] Alessandro Rubini & Jonathan Corbet, “Linux Device Drivers, 2nd Edition”, O’ REILLY, June 2001.
- [5] Rémy Card and Frank Mével, “the LINUX KERNEL book”, WILEY, 1998.
- [6] Richard Stones and Neil Matthew, “Beginning Linux Programming”, WROX, October 1999.
- [7] W. Simpson, “The Point-to-Point Protocol (PPP) for the Transmission of Multi-Protocol Datagrams over the Point-to-Point Link”, RFC 1331, May 1992.
- [8] W. Simpson, “PPP in HDLC-like Framing”, RFC 1662, July 1994.
- [9] William Stallings, “Operating Systems, Fourth Edition”, Prentice Hall, 2001
- [10] <http://biew.sourceforge.net/>, BIEW software.