REMOTE ACCESS SERVER
IN THE LABORATORY OF COMPUTER NETWORKS

The paper presents a new education platform that allows to provide remote access to the computer network laboratory of the Chair of Communications and Computer Networks. The remote access server’s hardware and the software architecture and its application is described.

1. INTRODUCTION

Nowadays it is difficult to overestimate the importance of telecommunication networks and the knowledgeability of its configuration governs in an engineer’s daily life. Students of electronics and telecommunications familiarize with the rules of the structure and configuration of modern telecommunication networks during the 30-hour course in the laboratory of computer networks in the third year of the full-time studies. Over that time students learn about routers and switches and gain basics skills in configuring of the devices. However, limited time of the course and the number of students in the laboratory groups has a hampering effect on the efficiency of the teaching process in configuring devices and, consequently, launching a network structure that consists of a few or a dozen of nodes. Thus, fulfilling the students’ expectations, a remote access server has been designed and created in the Chair of Communications and Computer Networks. The remote access server allows students to have remote access to laboratory beyond the hours of their scheduled lab classes. The device is designated for the students and the participants of the courses organized by Cisco Network Academy attached to the Chair of Communication and Computer Networks.

The paper has been divided into 4 chapters. Chapter 2, presents the device architecture and the software. Chapter 3 describes the application of remote server in the laboratory. The paper is concluded with an enumeration of possibilities of other applications and extensions of the presented device.

2. DEVICE ARCHITECTURE

The remote access server (Access Server) has been entirely designed and created by the authors. There were following assumptions in the project viewed as prerequisites: the use of free CAD tools, a selection of cheap electronics parts available on the Polish market, the use of cheap production technology. A
possibility to add more serial cards\(^1\) is ensured in the project. The device allows to control up to 16 devices.

1.1. Hardware

The hardware has been designed using Eagle tool \([1]\). This program in the freeware version allows designing PCB boards limited to dimensions 8 / 10 cm with two layers. The second restriction is not important because of the project’s budget. We decided to use two PCB layers in order to reduce PCB documentation costs. Because of limited size of the laboratory’s rack we decided to divide the hardware into three basics modules:

- **Main module**: main controller with Ethernet interface,
- **RS232C module**: card with 4 RS232C ports,
- **Module 230V**: module that switches on or switches off power to remote devices.

**Fig. 1. Remote router hardware architecture**

Figure 1 shows the block diagram of the device. RS232C cards have been connected to the main module by SPI bus. SPI bus consists of 4 signal wires. Three of them are shared by all devices that work in the bus. Master module chose slave device using forth line that is individual for each slave device. The RS232C card controls its 230V module.

The main module consists of: microcontroller (AtMega128), Ethernet interface and DC/DC pulse converter. The module has SD cart reader. In the project, two versions of the main module have been designed:

- Module with Lantronix Ethernet – serial converter,
- Module with enc28j60 Ethernet driver.

\(^1\) Serial ports RS232C allow to connect configure (console) ports of network devices. Each card has 4 serial ports and allows configuring up to 4 network devices.
Remote access server in the laboratory of computer networks

The first version has Ethernet – serial converter that simplifies the module because network stacks are implemented in the converter. Implementation of network stacks on the microcontroller is not required and its 4kB of internal microcontroller memory is sufficient. External memory is redundant. Figure 2 shows the first version of the main module. The microcontroller has two serial ports[1]. One of them is connected to the Ethernet – serial converter. The other is connected to the voltage converter TTL/RS232C that conditions the µC signals to RS232C voltage levels. The SPI bus allows communication with serial cards and SD flash memory card. It is possible to connect extra memory. Extra memory is marked as dashed box in Fig. 2. Adding extra memory may be necessary in order to add FAT32 file system support.

The second variant offers more possibilities because the microcontroller supports network stacks. Such an approach requires own network stack implementation, as well as more microcontroller resources. Figure 3 shows the diagram of the second version of the main module. Both serial ports of µC are connected to the voltage converter and can work as a console or modem ports. The Ethernet interface is connected to µC by SPI bus. Both versions of the main modules work with the same serial cards.

Figure 4 shows the serial card module. The module has microcontrollers AtMega162 connected to the SPI bus. This particular type has been chosen because it has two serial ports and it is inexpensive. Using 4 cheaper µC’s is not a good idea because each µC requires its own address line. However, serial card module has 2 µC’s and 4 serial ports that are connected to TTL/RS232C voltage converters. Please note that each card has two connectors A and B (Fig. 3). The first card is connected to the main module by connector A. Successive cards are
connected to the connector B of the previous card. Such a solution allows us to connect cards in a serial chain and a separate PCB board with bus is unnecessary. The bus is at the bottom (reverse) layer of each serial card and links connector A with connector B.

![Diagram of Serial Card Module](image)

**Figure 4. Serial card module**

The 230V modules have optoisolators that provide galvanic isolation between triacs and serial cards. The optoisolators used in the project have zero crossing circuits that allow to reduce the electrical disruption when the power line is switched on. Each card has 4 power lines and each line can control power up to 8 amperes but the total current controlled by the card can be higher than 12 amperes because of the power connector parameters. Such power is sufficient because typical router needs current between 0.5 and 2 amperes [1].

1.2. Software

The main module software has been written in C language. As the compiler we used a free compiler avr-gcc [4]. The codes were written in two different freeware IDEs (Integrated Development Environments): Avr Studio[5] and WinAvr[6]. Both of this IDEs have some disadvantages. Avr Studio can cooperate with avr-gcc compiler but the offered editor can not switch between functions and hide the code. Win-Avr has better IDE but worst cooperation with programmers and debuggers. In the project we use Jtag programmer that cooperates with Avr Studio. In the software of the main module the following functions have been implemented:

- Communications with Lantronix converter,
- Cooperation with RS232C cards (sending and receiving data),
- Command Line Interpreter that works on console port and allows reading the device status.

Communication between the main module and the serial modules is frameless. The SPI bus works in 8 bit mode. Each bit has a special meaning. Bit 7 must be zero in
order to detect if the card is connected, bit 6 determines if there is a data or command, bit 5 specifies the serial port of µC (serial port no. 0 or serial port no. 1) and the 4-th bit determines if the bits 3-0 carry more significant/less significant part of data. Such an approach requires sending 2 bytes thought SPI bus in order to send one byte of data between the main module and the serial card but is much more error-resistant and simplifies the serial card software.

The firmware of serial cards has to be very fast-responding because the serial card works as a slave device in SPI bus. In order to provide high efficiency, the software has been written in an assembler. The software provides four circular buffers for serial data (2 buffers for two receivers and 2 buffers for two transmitters). Receiver buffers store data until the master device can read the received data. Transmitter buffers store data until the µC’s transmitters initiate sending. Buffering data allows reducing number of polling of slave devices. The frameless communication simplifies the software of serial card. Code has 300 bytes of code (150 instructions)\(^2\).

At present, the software co-operates only with the first version of the main module. The application of serial ports limits the number of connection with remote access router to one. The result is necessary to specify the number of the serial port in one TCP connection and the number data we want to send to this serial port in data fields. The data are sent in Type Length Value container. The type of container specifies the serial port and the length specifies the number of data we want to send. Special types, such as switching power on or off line, need to have constant length in order to reduce the probability of unwanted operation because of a corrupted TCP packet.

3. REMOTE SERVER IN THE LABORATORY

Laboratory of Computer Networks includes 9 routers and a switch manufactured by Cisco Systems. The diagram of the Computer Networks Laboratory is shown in Fig. 5. The diagram presents the router connected directly or through DSL (Digital Subscriber Line) modems and IPBX (ISDN Private Automatic Branch Exchange) by serial cables (V.35, RS232, RS449 and X.21).

The Ethernet ports of the routers and NIC’s of the host which are enabled to configure network devices are located within appropriate virtual LAN. Such a solution allows us to treat a laboratory network as an intranet which connect 9 divisions of the firm, with the assumption that each division has one host (computer) connected to the intranet. The configuration ports of the network devices are connected to Access Server. The server task is to enable remote access to network devices connected to them through TCP connection. Therefore, the \textit{xRouter} program allows us to configure all the devices.

\(^2\) Avr microcontrollers have 8 bit architecture but commands have 16 codes.
**xRouter** program enables access to configuration ports of all network devices (Fig. 6) and to switch on/off all connected devices. Figure 7 shows GUI of the program.

The Power control of the network devices performed by Access Server is enabled by a remote restart of the network devices. This function is indispensable for refreshing IOS in the FLASH and the configuration file in the NVRAM memories. This procedure is applied by each terminated session. Due to above, terminated new session begins with the same, unchanged configuration. Of course, during each session students can save configuration files but uploading is possible only after gaining access to a given network devices through server. The remote Access Server makes it possible to carry on with scheduled exercises without being physically present in the laboratory. It should be remembered though that in each case students have no availability to access NIC’s of the computer in the laboratory.

---

3 Session i.e. TCP/port connection with particular network devices through access server.
Remote access server in the laboratory of computer networks

Within the framework of the exercises students can practice:
- elements of the configuration of Cisco Systems router such as: works modes of the router and interface and addressing configuration,
- routing problems: RIP and IGRP protocols and static routes,
- DHCP Server configuration on the Cisco router.

Additionally students can use Access Server for practicing advanced routing (OSPF, EIGRP and BGP protocols) and multilayer scalable networks configurations based on switches (this requires connection of Cisco Switches to Access Server depending on the needs, i.e. 3560, 2960, 2950, 2900 or 1900). Access Server will also be suitable for the newly prepared laboratory of advanced computer networks dedicated to new teaching profile „Computer Networks”. In this laboratory the Access Server is used for enable remote access to multi-service Cisco Routers and layer 2 and 3 switches. Beside the above issues, there are many

---

4 Due to limited length of the paper we do not included detailed description of the laboratory exercises. The list of exercises is available at http://compnet.et.put.poznan.pl.
additional ones related to safety and reliable functionality of networks which can be practiced with the presented Access Server.

4. CONCLUSIONS

In the paper we discuss the architecture and the software used for Access Server designing. The Access Server worked out at the Chair of Communication and Computer Networks Poznan University of Technology. The server is used for remote access to the network devices in the computer networks and advanced computer networks laboratories for students of electronics and telecommunication at PUT. The structure of the existing computer networks laboratory is described. Additionally, the use of Access Server in profiled exercises is indicated.

It is worth emphasizing that the presented Access Server can be applied for many other purposes outside laboratory environment. For example, it can used to control other devices. Thanks to its modular structure, it will possible to construct RS485 card dedicated for this server. Eight analog inputs are also designed in the server architecture. These inputs can be connected to sensor and other measuring devices such as: thermometers, door looks control or flooding sensors.

The other variant of the server enables also in/out lines with interruptions which are suitable for co-operation with of all kinds of serial buses such as for example one-wire. Additionally, RS232C port in the main module can be used to co-operate with PSTN or GSM modem, for example in case of a failure in the LAN.

Further development of the server can be versatile but from the view point of the advanced computer network laboratory it can include an implementation of TCP/IP network stack directly by students.

REFERENCES