A study on the possibility to use Raspberry Pi as a console server for remote access to devices in virtual learning environments

Diyana Kyuchukova, Georgi Hristov, Plamen Zahariev and Svilen Borisov
Department of Telecommunications
University of Ruse "Angel Kanchev"
Ruse, Bulgaria
dkyuchukova@uni-ruse.bg, ghristov@uni-ruse.bg, pzahariev@uni-ruse.bg, ODPleven@mail.bg

***Abstract* — During the last years many virtual laboratories have been developed, but not all of them were successfully adopted. Designing, planning and creating a virtual laboratory can be a very difficult process, which can get even more complicated if the laboratory has to provide access to real equipment. One of the most problematic steps when developing such virtual laboratories is the selection of the right access method. At the present moment of time there are many technologies that allow users to access devices remotely. One of these possible solutions is to use the Telnet/SSH protocols for remote access. Nevertheless, the use of these protocols require at least one active network interface per device, which in some cases can present a problem. There are some other drawbacks of using Telnet/SSH for remote access as well – virtual laboratories are often created with the purpose to be used in the education process, which means that the users of these systems in most cases are students. These less experienced users might accidently delete the configuration of the interface that is dedicated for the remote access, which will result in the immediate loss of connection to the device and will require administrator intervention to fix this access issue. An alternative solution to provide remote access is to use a console server. A detailed analysis on the possibilities for remote access using console servers, present them as reliable, but also as a very expensive solutions. In this paper we will investigate the possibilities to replace the expensive console servers with much cheaper devices. One such device is the Raspberry Pi – a low cost device that was developed to enhance the education process of children and students in the field of the computer sciences and their related subjects. In this paper we will investigate and compare the characteristics of the Raspberry Pi to the most widely used console servers and we will present a way to use the Raspberry Pi as a console server for remote access to telecommunication routers and switches.**

Keywords — virtual and remote laboratories, virtual learning environment, rapsberry pi, remote console access

I. INTRODUCTION

In the last years dozens of new sites and platforms for e-learning have emerged. Many of them provide free and unrestricted access to the learning materials, while other require subscription for a given period of time or per course [1]. There are even such that are limiting the access to the learning materials, based on the user affiliation to the hosting organization or based on the geographical location of the user.

The quality of the learning materials and the learning possibilities, which are provided by the e-learning sites and platforms also varies [2]. Some of them are offering only limited, uncompleted, wrong or even sometimes fabricated information, while other are providing detailed, structured and correct information. In terms of capability some e-learning sites and platforms provide text-based information only, while other provide additional build-in tools for assessments or simulation of certain processes [3]. Some laboratories even offer tools for remote access to real devices. These are considered the most complex to design and create, but they also provide the highest level of user satisfaction. Additionally, the experience that is gained when learning from such a platform or website is comparable to the in person traditional education.

II. CURRENT STATE OF THE TECHNOLOGIES FOR ONLINE ACCESS TO REMOTE DEVICES

Over the years many technologies and protocols for remote access have been developed, including Apple Remote Desktop Protocol, the Appliance Link Protocol, the Remote Desktop Protocol, etc. The majority of them are proprietary and provide remote access to specific hardware platforms or computer systems [4]. This makes them unsuitable as a universal solution for remote access. Among the first widely accepted universal protocols for remote access are Telnet and SSH (Fig.1).

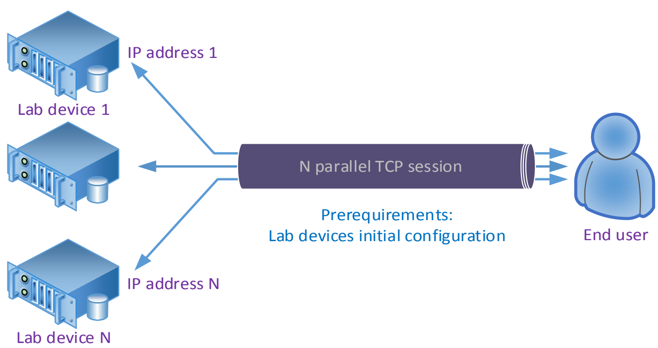

Fig. 1. Telnet or SSH remote access to laboratory devices

978-1-4799-1756-3/15/$31.00 © 2015 European Union

Both protocols have many similarities – they are client-server network protocols, they provide bidirectional interactive console communication, they are independent of the end devices and the operation systems, etc. The main difference between them is the fact that SSH is a cryptographically secured protocol that provides reliable console and terminal communication, while Telnet provides no encoding and security for the transmitted data.

Configuring the specialized network devices to support remote access using Telnet or SSH is a straightforward process, but requires at least one active network interface with a real IP address and Internet connectivity (Fig.1). In many cases this requirement can present an issue, since some of the specialized network devices are having limited number of interfaces. Additionally these devices are expected to be used by students or trainees, which are less experienced and might accidentally delete or modify the settings of the devices, which will make the devices useless and will require a system administrator to manually restore their connectivity settings.

A better solution for providing remote access to specialized laboratory devices is to use a middleware console server (Fig. 2). With this approach any remote access software can be used to establish a connection to the middleware server, which is connected to the console or auxiliary interfaces of the specialized laboratory devices. In this way the middleware device provides the possibility to use all network interfaces of the laboratory equipment and to modify the configuration of the devices without risking the connectivity to them. While this solution is solving some of the issues, which were discussed above, it is presenting several new ones. The first of these new challenges is the fact, that the middleware server needs to be initially configured and then regularly maintained by a network administrator. Additionally, the settings for the different connections to the devices have to be provided to the users. This is putting the user into a completely different non-transparent situation, where it is known that there is a middleware device. Additionally, many users might find it difficult to use remote access software for the establishing of the connection to the devices.

In order to make the user access to the laboratory devices much easier, a third approach can be implemented (Fig. 3), where the middleware device is still used, but is indirectly accessed using a graphical user interface (GUI).

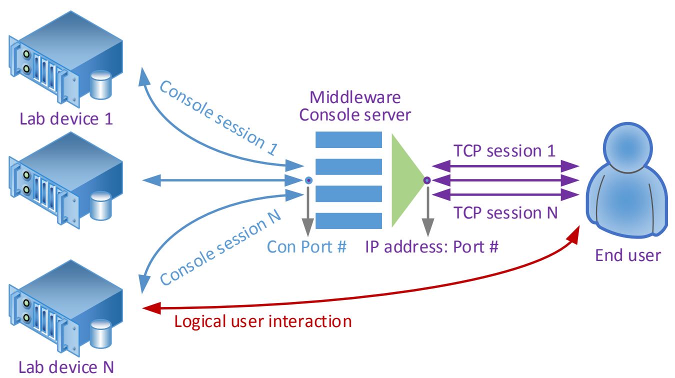

Fig. 2. Middleware system as a mediator in the remote access process

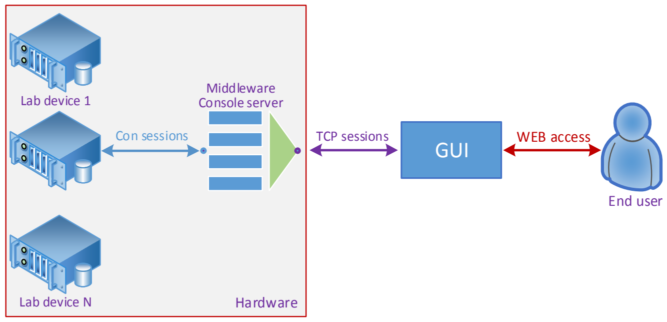

Fig. 3. Remote access to the laboratory equipment using GUI

With this approach the users receive much higher level of satisfaction when connecting to the devices and are completely unaware of the middleware console server.

III. Architecture and State of the Virtual Laboratory for Communication Networks and Systems

In the beginning of 2012 we have designed and created a *Virtual laboratory for Communication Networks and Systems* [5]. The laboratory is based on the third approach, which was discussed in the previous section of the paper, and is now one of the learning environments that are used by students from the Department of Telecommunications at the University of Ruse. The laboratory has undergone several upgrades and modifications since its first days and now represents a complete virtual learning environment that provides the most recent learning materials, video tutorials, interactive tools and resources in the area of the local and global communication networks. Among the other things, the laboratory is providing to the students the possibility to simultaneously perform practical exercises on real devices remotely. This is achieved by organizing the specialized devices in small groups (pods) and by providing access to them (Fig. 4) from a custom developed graphical user interface (Fig. 5). Every pod consists of 6 devices – 3 network routers (Cisco 1941, Cisco 2610, Cisco 2620MX or Cisco 2911) and 3 LAN switches (mainly Cisco Catalyst 2950 and Catalyst 2960 series). The devices from every pod are connected to a separate local network and the entire setup represents a complete practical environment for the students to perform all tasks and experiments in the area of the local and global IP networks.

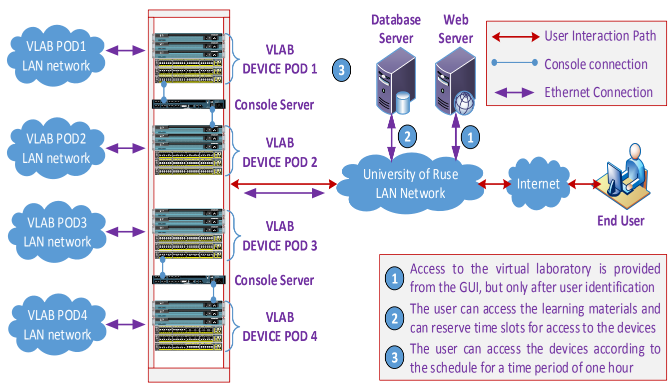

Fig. 4. Architecture of the virtual laboratory and the user interaction path

Fig. 5. The home screen of the web based GUI of the virtual laboratory

In order to provide connectivity to the specialized devices we use two different console servers – a Cisco 2511 Access server and a Perle STS 16 IOLAN console server (Fig. 6).

a) b)

Fig. 6. Both middleware console servers used in the laboratory – a Cisco 2511 Access server (a) and a Perle STS 16 Iolan console server (b)

While full and detailed description of both servers can be obtained from the resources provided by the manufacturers [6, 7], their most important characteristics are given in the table below.

TABLE I. CHARACTERISTICS OF THE CONSOLE SERVERS

Parameters	***Device***	
	Cisco 2511 Access server	***Perle STS 16 IOLAN console server***
Processor	Motorola 68030 - 20MHz	MPC8349E - 400 MHz
RAM (MB)	16	64
Flash (MB)	16	16
Number of Serial Ports	16 ports via octal fan-out cables	16 RS232 DTE on RJ45
Operating System	Cisco IOS	Perle Custom Linux/Unix based

From the table it is clearly visible that the devices are very different in terms of both hardware and software components, and the only common thing they share is the number of Serial ports. This makes the maintenance and support of the devices difficult and requires wider area of knowledge. Additionally, the Cisco 2511 Access server is no longer supported and in case of any software or hardware issue the device will become unusable. Replacing the device with another console server is always an option, but the latest products on the market either provide additional functionality that cannot be fully used in the laboratory or are too expensive.

IV. INVESTIGATION ON THE POSSIBILITY TO REPLACE THE CONSOLE SERVERS WITH RASPBERRY PI DEVICES

The Raspberry Pi family of pocket-sized microcomputers has been developed by The Raspberry Pi Foundation [8] and represents a cheap, but functional environment for practical teaching of programing languages and other related to the computer sciences subjects. Currently on the market there are several different versions of the microcomputer, with the main characteristics of the devices presented in Table 2.

TABLE II. CHARACTERISTICS OF THE RASPBERRY PI FAMILY OF MICROCOMPUTERS

Parameters	***Raspberry Pi***			
	Model A	***Model B***	***Model B+***	***2***
Processor	700 MHz single-core ARM1176JZF-S			900 MHz quad-core ARM Cortex-A7
GPU	250 MHz Broadcom VideoCore IV OpenGL ES 2.0			
RAM (MB)	256	512	512	1024
USB 2.0	1 port	2 ports	4 ports	
Storage	SD/MMC/SDIO card slot		MicroSD card slot	
Network interfaces	-	10/100 Mbit/s Ethernet interface		

Further in this paper we will present the Raspberry Pi Model B+ microcomputer (Fig. 7) as a potential solution for replacement of the console servers. These microcomputers provide much more possibilities, than these required by us for the purpose of the virtual lab, but due to their very low prices, small dimensions and general capabilities we consider them suitable for this task. The Raspberry Pi microcomputer has no direct serial interfaces, which complicates things a little bit, but this issue can be solved with the help of a USB to RS-232 cable [9] or a GPIO serial port board. By default the microcomputer is not preinstalled with an operating system, so that the end user can select the one that best suits his needs. Several operating systems are available for free download at the web site of the Raspberry Pi Foundation. In our setup we have used the latest stable version of the Raspbian OS.

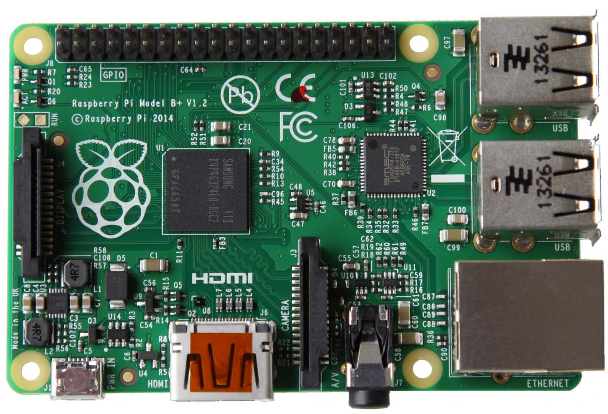

Fig.7. Raspberry Pi Model B+ microcomputer

The process of using the Raspberry Pi as a console server can be described with the following steps:

- The first of these steps is the selection of the serial port, and there are two options to choose from – the built-in serial port and the virtual serial ports. The first case will require the use of a GPIO serial board with a DB9 interface or similar hardware. This option is always a solution, but due to the limited number of GPIO ports, the connection to every of the devices in the pod must be provided by a different Raspberry Pi, which is very ineffective. An alternative option is to use a USB to Serial cable for every of the four USB interfaces of the microcomputer. In this way a single Raspberry Pi can provide the connection to four devices. The difference between both options is the way the Raspberry Pi refers to the devices. In the first case it is using the built-in set of devices */dev/ttyS0*, while in the second case the devices might be accessible as */dev/ttyUSB0* to */dev/ttyUSB3*. Despite the fact that the second approach is much more efficient, some USB to Serial cables are not supported by the Raspberry Pi operating systems, which might present issues and might require the installation of additional drivers or the replacement of the cables.

- The second step is the configuration of the serial ports, so that a console session can be established. The parameters of the serial ports, which require configuration are Speed/Baud rate, Number of Bits, Parity, Number of Stop Bits and Flow control.

Once these steps are completed a direct access to the devices can be established using the shell of the Raspbian OS. In order to provide transparent access to the laboratory equipment an additional script is executed to replace the access to the shell of the microcomputer with direct access to the command terminal of the routers and the switches. This functionality can also be achieved with software tools like ser2net [10]. The Java applet, which is used by the GUI for SSH access to the console server (Fig. 8), also needs to be corrected to use the IP address of the Raspberry Pi. The complete architecture of the laboratory is presented on Fig. 9.

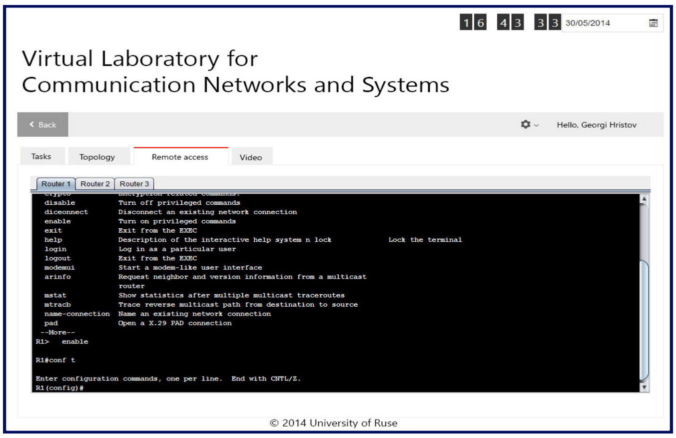

Fig. 8. Direct access to the command terminal of the remote network devices provided using a Java applet for SSH connection embedded in the GUI of the virtual laboratory and a Raspberry Pi microcomputer as a console server

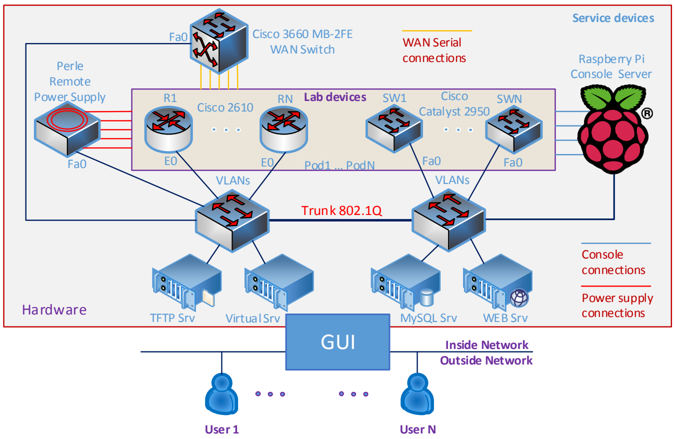

Fig. 9. The architecture of the virtual laboratory with the Raspberry Pi microcomputer as a console server

Conclusions

Although not specifically designed for the role of a console server, the Raspberry Pi microcomputer is capable of providing such functionality. The limitations of the microcomputer, in terms of GPIO or USB ports, require the use of several devices to provide the same number of console ports as those provided by the dedicated console servers. The much lower price of the Raspberry Pi compensates for this disadvantage and makes the microcomputer a feasible replacement for the console servers.

Acknowledgments

The work presented in this paper is completed as a partial fulfillment of Project №BG051PO001-3.3.06-0008 “Supporting Academic Development of Scientific Personnel in Engineering and Information Science and Technologies” of the European Social Fund under Operational Programme “Human Resources Development”. The contents of this document are the sole responsibility of the authors and the “Angel Kanchev” University of Ruse and shouldn’t be considered as an official position of the European Union or the Bulgarian Ministry of Education and Science.

References

[1] Saeed, F.A., Comparing and Evaluating Open Source E-learning Platforms, International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume 3, Issue 3, July 2013

[2] Ehlers, U. "Quality in e-learning from a learner’s perspective." European Journal for Distance and Open Learning 1 (2004): 73-90.

[3] Gilbert, J., Morton S., Rowley, J. "e-Learning: The student experience." British Journal of Educational Technology 38.4 (2007): 560-573.

[4] Li, L., Zheng, Y., Zhong, S., Research on Web-based Virtual Laboratories, International Conference on Computer, Mechatronics, Control and Electronic Engineering, Changchun, 2010, pp. 478-481.

[5] G. Hristov, P. Zahariev, D. Kyuchukova, “Challenges and solutions for real-time remote console access to telecommunication equipment”, International conference on ITHET, York, UK, 2014

[6] http://www.cisco.com/c/en/us/support/routers/2511-router/model.html

[7] http://www.perle.com/products/iolan-sts-terminal-server.shtml

[8] http://www.raspberrypi.org

[9] http://www.ftdichip.com/Products/Cables/USBRS232.htm

[10] http://sourceforge.net/projects/ser2net