Cent. Eur. J. Comp. Sci. • 2(3) • 2012 • 344-353
DOI: 10.2478/s13537-012-0026-4

VERSITA

Central European Journal of **Computer Science**

New approach to remote laboratory in regard to topology change and self–repair feature

Research Article

Katarína Kleinová*, Peter Feciľak†

Technical University of Košice,
Letná 9, 04001 Košice, Slovakia

Received 10 February 2012; accepted 17 August 2012

Abstract: Remote laboratories deal with performing real lab experiments remotely via the Internet. Recent advances in the Internet/web technologies and computer-controlled instrumentation allow net–based techniques to be used for setting up remote real laboratory access. This paper deals with the solutions for remote laboratory providing services for the operation of remote laboratory. The paper addresses problems related to the requirement of flexible, secure and easy remote access to laboratory equipment as well as the need for hardware and/or software re–configuration. This paper presents a unique concept for logical topology change with the usage of Q-in-Q tunneling and automated self–repair feature after failure.

Keywords: virtual laboratory • Q-in-Q tunneling • AToM • knowledge evaluation • automated password-recovery • logical topology • remote access

1. Introduction

A virtual laboratory in terms of this paper represents an environment which is used for blended distance learning in Networking Academy program. The main purpose of the virtual laboratory is to provide remote access to physical devices in the network laboratory with the goal of reaching exercises on real devices placed in virtual laboratory environment, combining them with virtual devices, and providing services for a wide range of applications used in complex labs (like Authentication server, Virtual Private Network server, Active directory (domain) server, etc.).
Main areas which must be reflected by a modern virtual laboratory are as follows:

- Remote access to real or virtualized devices (defined interface).
- Devices maintenance (password-recovery procedure, power on/off).
- Reservation system.

* *E-mail: katarina.kleinova@cnl.sk (Corresponding author)*
† *E-mail: peter.fecilak@cnl.sk*

- Content for exercises (labs).
- Physical/Logical topology re-configuration (dependent on the exercise).
- Knowledge evaluation system.

This paper presents several identified drawbacks of existing solutions. A couple of virtual laboratory solutions were already published in [2, 5]. The paper follows with a detailed solution description for logical topology change/re–configuration and automated topology repair after failure. This paper also describes physical lab environment components which we have used in the virtual laboratory at the Academy Support Center at Košice.

2. Drawbacks of existing solutions

In this section we will pass through each aspect of modern virtual laboratory (Section 1) and we will describe some drawbacks of existing solutions that are used in other virtual laboratories.

2.1. Remote access to real or virtualized devices

Depending on devices used in a virtual laboratory, it is necessary to define an interface for remote access to the equipment. In case of a remote laboratory for computer networks, we usually use network devices like routers and switches that can be managed over the Telnet/SSH (Secure shell) protocol or through serial or auxiliary interfaces. In general, it is TCP/IP or serial communication interface (RS232) that is using 9600 bits per second speed by default.

The cheapest way to access devices remotely is by using their own TCP/IP management interface, like using the Telnet or SSH protocol. This solution has its weaknesses in the need of correct configuration of the TCP/IP stack on used devices. In the case when an IP address will be re-configured by a user of the virtual laboratory, the device will not be accessible remotely anymore at the defined IP address. Even if we place some kind of a warning message, such as ”Do not re-configure interface, etc.”, we will be unable to guarantee accessibility of virtual laboratory devices as it might be malfunctioned by the user. Therefore it is more stable if remote access is based on a terminal server that has defined access interface used for accessing remote devices connected to the terminal server. Usually Telnet or SSH protocols are used for remote access.

There are a lot of laboratories that use terminal server with Telnet access assigning to different ports for each device connected to the terminal server. A terminal server based on a Cisco integrated services router (ISR) with 8 to 32 serial asynchronous interfaces is mostly used. There is also the requirement of direct access to the terminal server settings for managing situations like clearing frozen sessions or changing port speeds. If there is no other user interface to communicate settings directly to the terminal server, then there is no way to access devices with different speeds of console port than default. This might be the main disadvantage of this solution. The device could become unusable for next reserved sessions as soon as a user changes the speed of console port during a lab exercise configuration.

However, direct Telnet access (even relayed through the terminal server) might be problematic in networks with too restrictive security policies, where, due to the security weakness of the Telnet protocol, it is usually blocked. Therefore it is highly recommended to provide a secure way of communication with the terminal server.

A user interaction in the remote network topology is also an important part of the virtual laboratory solution. There is lack of virtual laboratories which allow to combine remote laboratory equipment and users' own equipment with possibility of connecting them into a remote network topology. A remote network topology usually contains intermediate devices like routers and switches, and there is lack of end devices like computers, IP phones and printers that can be controlled remotely. There is a strong need for terminal services not only for serial communication, but also for virtualization of operating systems and emulation of other network devices.

2.2. Devices maintenance

There are several users' actions that can be done by the user and can completely malfunction a virtual laboratory. These actions include:

- Re-configuration of console access or privileged exec mode passwords *will cause inaccessibility of the virtual laboratory for next users trying to access some devices.*

- Changing the speed of console or auxiliary port on a device *will cause a virtual lab device inaccessibility due to need for speed change at the terminal server.*

- Enabling security features that are blocking the password-recovery procedure *will cause the lab equipment to be unreachable due to impossibility of automatic password-recovery procedure*.

- Erasing the flash memory *will cause inability to boot the device and due to this problem it will not be accessible for lab training.*

Therefore there is need for virtual laboratory equipment maintenance.
In modern virtual laboratories there is need for command authorization that cannot be done on the Cisco ISR terminal server. Therefore many virtual laboratories that are using a Cisco terminal server are facing problems listed above and are solving them by employing a person who is manually checking devices after each lab reservation. There is also a possibility to authorize commands at the IOS (Integrated Operating System) application level with an AAA (Authentication, Authorization, Accounting) server using the Tacacs or Radius protocol. The solution using an authorization server has its weakness in that it relies on correct device configuration and its connectivity to the AAA server. It is also limiting in case that authentication, authorization and accounting is part of an exercise.

2.3. Reservation system

Each virtual laboratory uses its own reservation system. Usually the reservation process is not easy. Some reservation systems are based on manual account creation (on devices or on the terminal server) allowing a user to access devices remotely. This process can be partially or fully automated, which means that during reservation of lab session there are processes including:

- Receiving a lab reservation request from community of virtual laboratory users. There are different forms of request processing – e-mail, web form, phone call to the maintainer, etc.

- Approval and/or direct reservation.

- Creating an account and defining its access rules.

- Notification of a person who is wishing to reserve lab equipment.

Processing of electronic requests (done via a web form) can be almost fully automated, however there is also a possibility of other non e-form requests to the lab equipment maintainer. Therefore there is request for easy and fast process of equipment reservation integrated into traditional work user interfaces without spending too much time logging into the reservation system, filling form items like e-mail of requester, date and time of lab reservation and notifying the requester back.

2.4. Physical/Logical topology re-configuration

Sometimes it is necessary to re-configure the network topology depending on the exercise that a user wants to perform. There are a lot of virtual laboratories that do not allow topology re-configuration and all labs are based on the same topology or they allow topology re-configuration only by technical staff physically changing the network topology. There are some approaches that allow automated change of physical topology based on connection matrices that physically interconnect wires by relay circuits. Other virtual laboratories are changing logical topology of the Ethernet network instead of physical topology re-configuration. There is the issue with using solutions based on VLANs for creating interconnection on L2 device for exercises related to L2 protocols like CDP (Cisco Discovery Protocol), STP (Spanning Tree Protocol), VTP (Vlan Trunking Protocol).

2.5. Exercise contents and knowledge evaluation

Every virtual laboratory has its technical limitations. Based on technical limitations, there is a limited set of exercises that can be done on the equipment of a virtual laboratory. Laboratories that did not solve topological re-configuration, are usually dedicated to specific areas and therefore there is lack of scalability and possibility of doing a wide range of exercises is typically missing. If the laboratory is more static than dynamic in terms of topology creation, then there is usually no option for content creation (like interconnection of devices together by web-oriented applications such as Packet Tracer [1]).

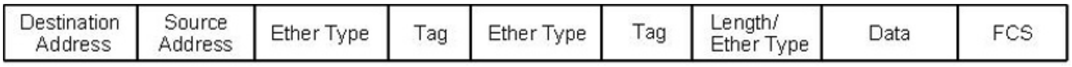
An important part of a modern virtual laboratory is a system of knowledge evaluation. Based on exercises performed by a user in the virtual laboratory, there should be a system responsible for collecting and evaluating users' files. There are a number of virtual laboratories that are evaluating exercises only by comparing representative solutions against users' solutions. The percentage of the difference between both solutions (template and user) is inverse percentage to 100%. There are some problems of that solution as it is not so exact and also there is almost no variability in exercises (e.g. the IP address needs to be the same) and therefore an exercise has to be stated so precisely that there is no other solution for the task.

3. Approach to topology change and self–repair feature

There are a couple of solutions for drawbacks mentioned in Section 2 presented in [2] and [5]. In this section we present our unique approach to logical topology management with the usage of Q–in–Q [4] and automated password–recovery feature for routers and switches which is part of automated self–repair feature executable after a failure.

3.1. Q–in–Q tunneling and its specific usage in remote laboratory with the goal of topology change

Q–in–Q (802.1q Tunneling) is mainly used by internet service providers offering the service for interconnection of customer sites on Layer2 Ethernet technology. The goal of this technique is to double tag an Ethernet frame and this allows customer traffic separation. Figure 1 shows the structure of a double tagged Ethernet frame.



Destination Address	Source Address	Ether Type	Tag	Ether Type	Tag	Length/ Ether Type	Data	FCS

Figure 1. Double tagged Ethernet frame structure.

802.1q enables service providers to use a single VLAN IDs to support customers who have multiple VLANs. On the customer site, a customer switch is connected to a service provider switch and uses trunk mode on that port. On the service provider switch 802.1q Tunnel mode is used. When Q-in-Q tunnelling is enabled, trunk interfaces are assumed to be a part of the service provider network and access interfaces part of the customer network. A trunk interface can be a member of multiple VLANs from service providers. An access interface can receive tagged or untagged frames. A customer has usually allocated only one VLAN from service provider and every frame coming from the customer is tagged with the allocated unique VLAN tag.

Figure 2 shows the service provider network when Q–in–Q double tagging is used for two customers. Every frame from a customer on transmitted site is tagged at the service provider switch. The original VLAN tag of the frame is not changed and is transmitted transparently, so every frame has two VLAN tags (inner and outer tag) in the service provider network. The inner tag is the customer VLAN tag, the outer tag is the service provider allocated VLAN tag. On the receiving site, the outer tag is removed in the service provider switch and the frame with original customer tag is forwarded to customer. On the base of this, different customers can use the same VLAN tags.

Because of trunk mode on customer port, there are a few requirements:

- Port on the service provider switch must set BPDU (Bridge Protocol Data Unit) Filter and Root Guard to prevent the customer switch to act as a STP root switch.
- It is necessary to configure protocol tunnelling to enable the VTP between customer switches (not working by default) in service provider switches.
- UDLD (UniDirectional Link Detection), PAgP (Port Aggregation Protocol) and CDP (disabled by default) can work.

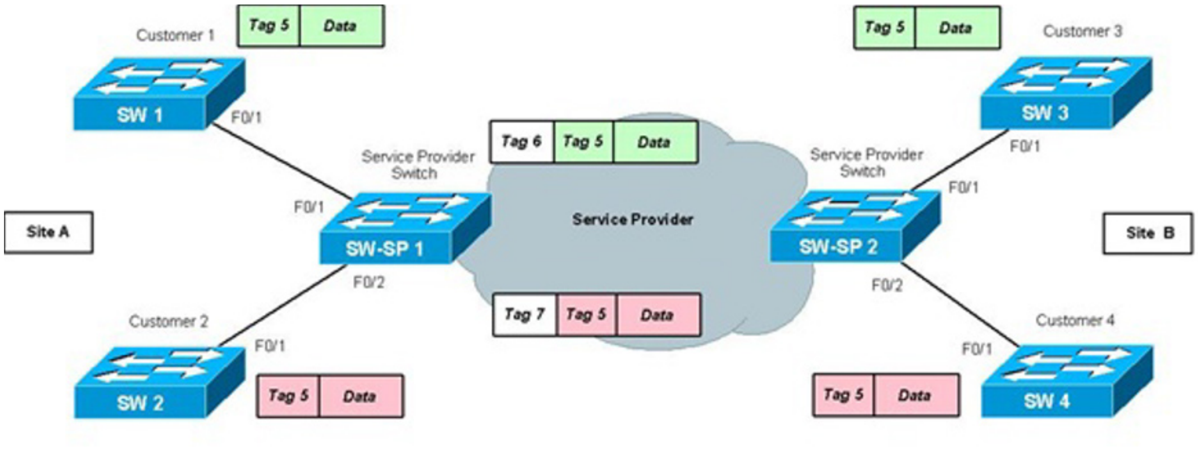


Figure 2. Service provider network with Q–in–Q.

- VLAN ID field in the 802.1q frame has 12 bits therefore the maximum VLAN ID allocated to the customer can be 4096.

There are few limitations of Q-in-Q tunnelling:

- Q-in-Q tunnelling does not support IGMP (Internet Group Membership Protocol) snooping or most access port security features.
- It is not available to disable MAC learning.
- There is no per-VLAN (customer) policing or per-VLAN (outgoing) shaping and limiting with Q-in-Q.

However, Q–in–Q tunneling provides a great feature – the transparent interconnection of customers (physical ports of device). Our approach to logical topology building of virtual laboratory uses this feature.
Our physical topology of a virtual laboratory is shown in Figure 3. This network topology is similar to the one presented in [3]. The topology is physically static, we do not use any connection matrices as there is no need for doing this. We have decided to manage the topology more logically than physically by using Q–in–Q tunneling and Any Transport Over MPLS (AToM) [6]. These technologies allow us to logically create interconnections between all devices by using separated VLANs and to tunnel layer 2 protocols like CDP/DTP/STP by using Q–in–Q tunneling. Also interconnections between devices using serial interfaces (WIC-2T) can be done by frame relay circuits or by using encapsulation (tunneling) to MPLS (AToM). For each interconnection of devices we are using internally different VLAN to separate traffic and L2 tunneling techniques to provide transparent bridging of interfaces. Table 1 shows an example configuration of virtual laboratory components when building a logical topology from a physical topology (Figure 3) shown in Figure 4.

Table 1. Example of MLS configuration for interconnection SW1-SW3.

MLS(config)# interface range Fa0/9, Fa0/15
MLS(config-if-range)# switchport mode dot1-tunnel
MLS(config-if-range)# l2protocol-tunnel cdp
MLS(config-if-range)# l2protocol-tunnel stp
MLS(config-if-range)# l2protocol-tunnel vtp
MLS(config-if-range)# switchport access vlan 10
MLS(config-if-range)# description SW1-SW3

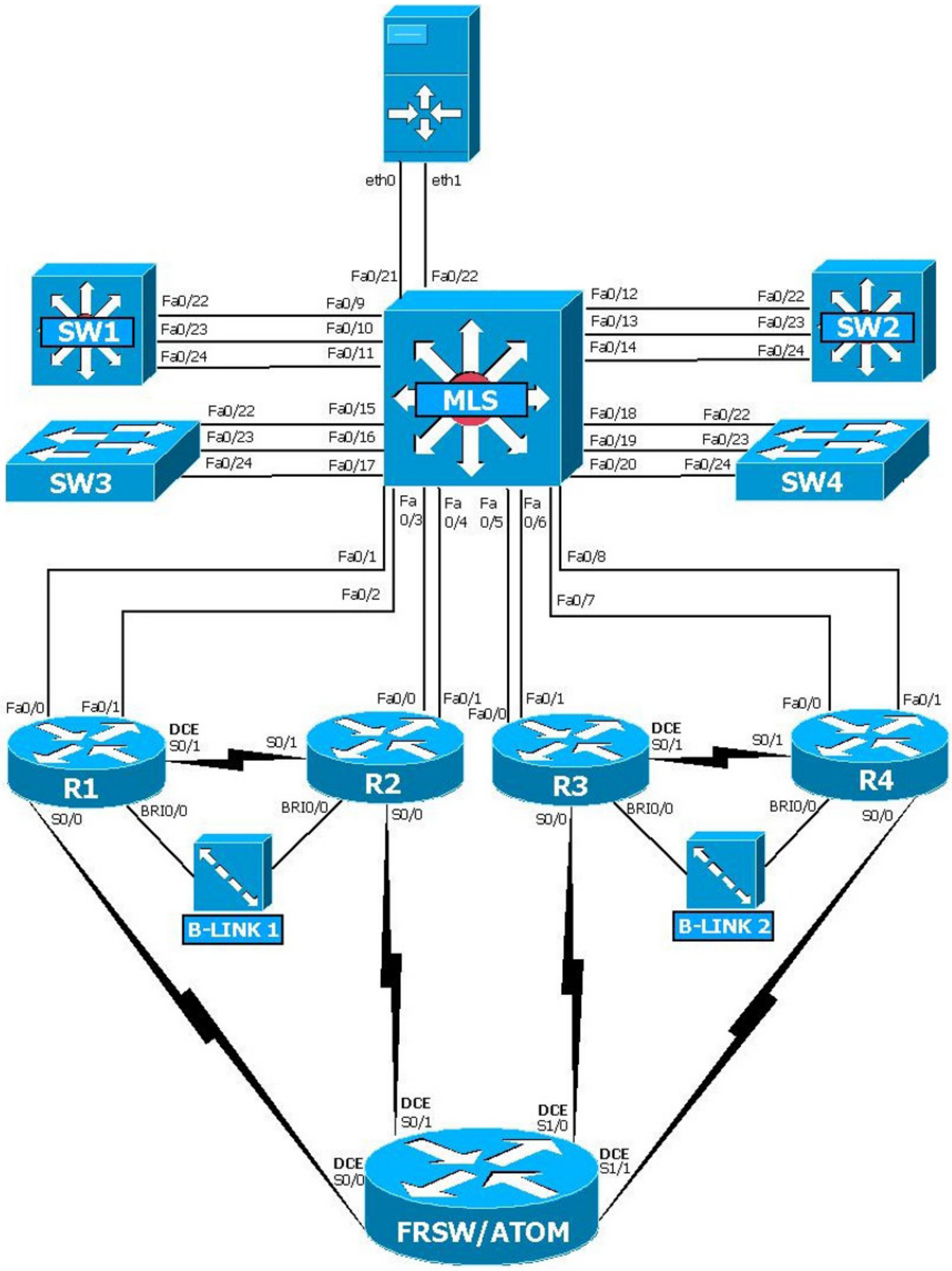


Figure 3. Virtual laboratory topology.

There is no need for hardware re-configuration in switching between exercises when Q–in–Q tunneling is used with the goal of establishing logical topology over the physical interconnections of the laboratory. It is less expensive and easier

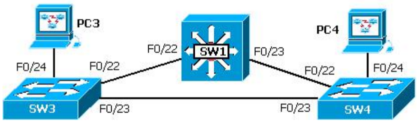


Figure 4. Example of virtual laboratory topology.

to implement such a feature of re-configuration as a part of software solution of the virtual laboratory. Depending on an exercise, this allows the user to load different configurations of logical topologies and to make full use of virtual laboratory resources.

3.2. Automated password-recovery feature for routers and switches

Automated password-recovery is a part of the self–repair feature of the laboratory which is used for recovery of virtual/real equipments after failure due to user changes done during exercises. There are two different password-recovery actions that need to be supported by a virtual laboratory. Password-recovery for routers is a bit different than for switches. There is a strong need for speed change ability on each console interface for successful password-recovery on routers and mechanical push of the MODE button for password-recovery on Catalyst switches. Key to password–recovery on routers is in control+break sequence generation. Practically this break sequence is generated by slowing speed down of console port lower than speed currently used and by sending 10 spaces. Therefore password-recovery on routers is done in the following steps:

1. Power cycle the router (off/on).
2. Change speed of console port to 1200 bits per second.
3. Send 10 spaces (0x20).
4. Change speed of console port back to default (9600 bits per second).
5. Configure config-register to 0x2142.
6. Reload the router.
7. Change config register back to default (0x2102).

The main user interface is a web application so from the point of view of the end user, the password–recovery is just one–click from the web interface. However, in the background of the application there is a set of scripts running, which prepare the setup needed for the steps described above. The combination of Bash–script and utilities like expect and setserial are used with the goal of changing the console speed and for automated user–interactivity.

Password-recovery procedure on Catalyst switches requires to manually push the MODE button. The easiest way to do this is by shorting the MODE button circuit by the contact relay managed from the virtual laboratory server. As we want to keep warranty on our virtual lab equipment, we have developed a unique prototype for manual pressing of MODE button. "Buttoner" device is managed by SNMP (Simple Network Management Protocol) and is mounted in a rack on the top of a catalyst switch. Figure 5 shows the prototype of the BS104 device which is rack–mounted.

Basic elements of the prototype are:

- Lantronix XPort - the TCP/IP to RS232(TTL) transmitter
- Atmel microprocessor - processing unit
- 4 stepper motors - provides manual push of the button on the switch

Figure 6 shows internal components of the buttoner device.

For the purpose of power control we have used an SNMP managed power distribution unit (Figure 7).

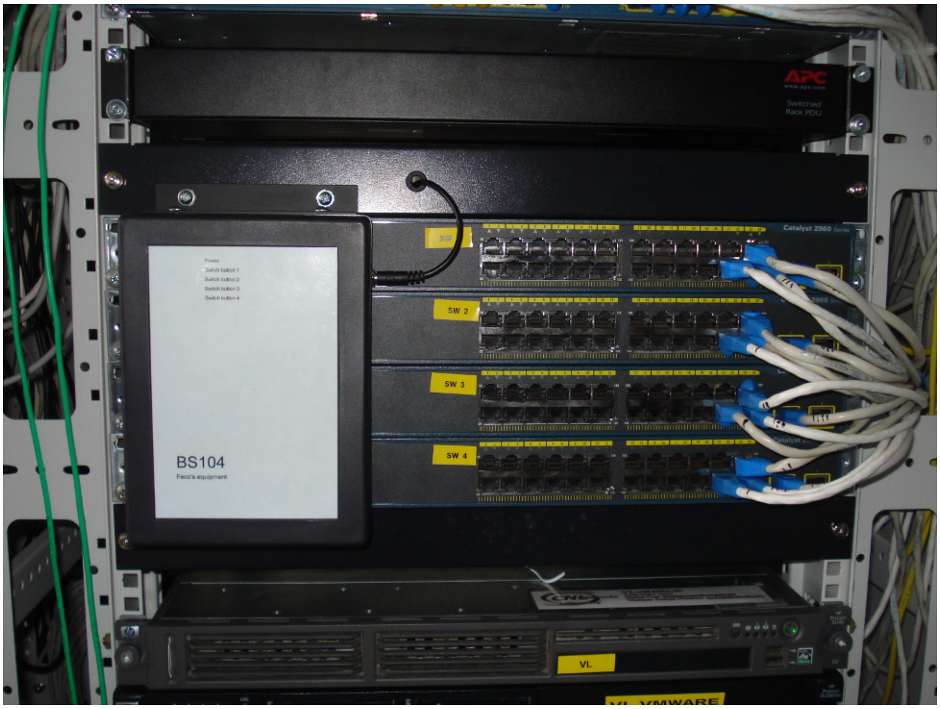


Figure 5. BS104 prototype device.

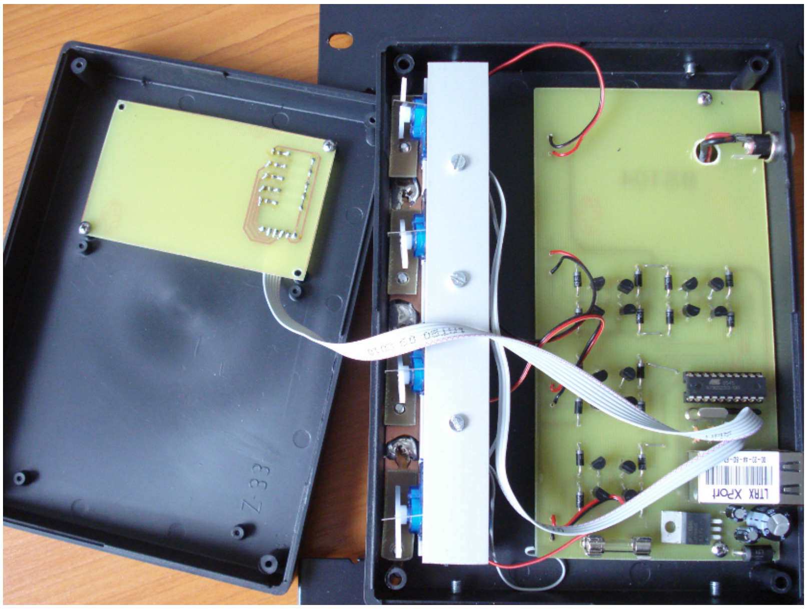


Figure 6. Internal components of BS104 prototype device.

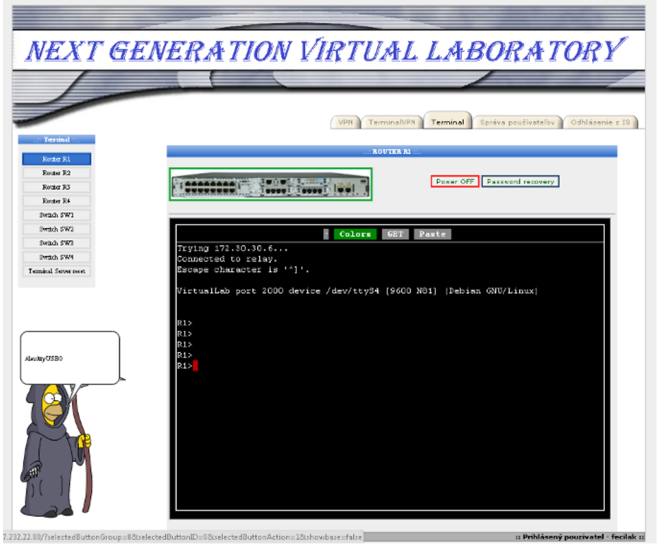
Within the steps for password–recovery, the first operation is to power cycle devices. Automated Bash–scripts are starting with an SNMP message toward an APC power distribution unit with the request of power–cycling individual ports depending on the device chosen. Once done, a second SNMP message is sent to the BS104 device with the

Figure 7. APC switched rack power distribution unit.

request for a manual push of the button. After 10 seconds, BS104 can release the button and the device will remain in the service mode. At this stage, a combination of Bash–scripts and the expect tool finalizes the password–recovery procedure by automating user interaction and clearing configurations.

3.3. Web user interface

A web user interface (Figure 8) acts as the interface for communication with the user. It is the central element for putting all the virtual laboratory pieces together. We have used some technologies like AJAX (Asynchronous JavaScript and XML) terminal that allows us to tunnel communication in case of limited access from user environment, PHP (Programmable Hypertext Preprocessor) and Java technologies for running terminals from an end station in a non-firewalled environment.



```
Trying 172.30.30.6...
Connected to relay.
Escape character is '^]'.

VirtualLab port 2000 device /dev/ttyS4 [9600 N81] (Debian GNU/Linux)


R1>
R1>
R1>
R1>
R1>
```

```
7.232.22.80/?selectedButtonGroup=8&selectedButtonID=0&selectedButtonAction=1&showbase=false
```

Figure 8. Web user interface of virtual laboratory.

4. Conclusion

Remote laboratories have come a long way since their introduction in the late 1980s/early 1990s. The last decade witnessed a move from breaking technological barriers to the enhancement of pedagogical features, and the next generations will undoubtedly include embedded tutoring and personal assessment features that will improve their pedagogical effectiveness still further. The integration of remote experiments with the remaining e-learning contents offers an enormous potential for improving the pedagogical success of science and engineering students and therefore we will more focus on this topic in our future work.

In this paper we have presented several solutions for a remote laboratory which we are operating at Academy Support

Center at Košice. The main goal of this paper was to present our unique approach to topology change with the usage of service provider technology of Q–in–Q and to demonstrate our own approach to topology self–repair. This was clearly presented in Section 3. Our future work will also be devoted to videoconferencing as a part of training in a virtual laboratory.

Acknowledgment

This work is the result of the project implementation: Development of the center of information and communication technologies for knowledge systems (project number: 26220120030) supported by the research & development operational program funded by the ERDF.

References

[1] Cisco Packet Tracer, [on-line 11.3.2011] URL: http://www.cisco.com/web/learning/netacad/course_catalog/PacketTracer.html

[2] Feciľak P., Kleinová K., Jakab F., Solutions for virtual laboratory, In: Proceedings of ICNS 2011, The Seventh International Conference on Networking and Services, May 22 to May 27 2011, Venice/Mestre, Italy

[3] Grellneth I. et al., VLAB at the Institute of Computer Systems and Networks FIIT STU, Acta Electrotech. Inf., 10, 65-71, 2010

[4] IEEE 802.1Q-in-Q VLAN Tag Termination, [on-line 11.3.2011] URL: http://www.cisco.com/en/US/docs/ios/lanswitch/configuration/guide/lsw_ieee_802.1q.html

[5] Jakab F., Janitor J., Nagy M., Virtual Lab in a Distributed International Environment, SVC EDINET, The Fifth International Conference ICNS 2009 on Networking and Services, LMPCNA 2009, Valencia, 20-25 April 2009, Valencia, Spain, IEEE Computer Society

[6] Lobo L., Lakshman U., MPLS Configuration on Cisco IOS Software, Cisco Press 2006