Emulating Cisco Network Laboratory Topologies in the Cloud

Igor Grellneth*, Martin Hruby* and Peter Vilhan**
* Institute of Computer Systems and Networks, Faculty of Informatics and Information Technologies, SUT, Bratislava, Slovakia
** Institute of Applied Informatics, Faculty of Informatics and Information Technologies, SUT, Bratislava, Slovakia
grellneth@fiit.stuba.sk, hruby@fiit.stuba.sk, vilhan@fiit.stuba.sk

***Abstract*—Presently a multitude of virtualization solutions have been implemented in both classroom environments and by students at home. Their success and widespread deployment made them an ideal surrogate to real hardware based laboratories and an excellent aid in the learning process. However, the intense hardware requirements often limit the number of emulated devices used, and the number of concurrent sessions. To address these issues, in this paper we propose a cloud-based virtual laboratory.**

I. Introduction

Among the most popular Cisco IOS emulation tools in recent years are Dynamips and Dynagen upon which the successful GNS3 environment is based. The usefulness of these tools in learning was an incentive for us, and prompted us to create a Virtual Laboratory (only VLAB further on) [1]. The VLAB provides a framework for virtualization of the network topology and enables registration, authentication, reservation and resource authorization for the student. The VLAB is implemented on a single Unix server and this represents a few issues which we would like to address. In this paper we take our existing VLAB environment a step further and move the existing implementation into the cloud to utilize the available computing resources.

Our motivation behind this effort is to be able to deliver new Cisco CCNP Troubleshooting laboratory lessons to students regardless of their location on a shared, cloud-based emulation environment. These lessons feature preconfigured networks which contain a bug, unwanted functionality or an undesirable byproduct which the student has to identify and reconfigure the network in order to achieve the desired result. Currently in class and on real equipment, the task of preparing the lab (cabling, base configuration, etc.) itself brings up new issues, which are not part of the lesson and therefore it makes the whole troubleshooting part more complicated and spread out in many different directions. This way it is quite difficult to focus on the topic the lab was meant for, because the student does not see the expected outcome described in the lab until he solves the issues he added with the installation of the lab itself. Having dedicated hardware and preinstalled configurations for the Cisco CCNP Troubleshooting course is, in our view, inefficient, time-consuming and expensive. Therefore we propose to conduct laboratory lessons in this course in a virtualized environment capable of running many concurrent sessions with automatic configuration deployment to the emulated network devices. A natural candidate for addressing this issue was our existing VLAB.

Utilizing VLAB during lessons helped eliminate problems with cabling and the need to prepare configurations before the actual lesson begins. It can be used by up to four students at once with present hardware configuration (4GB RAM). High availability of virtual laboratory allows students to use it up to 96 simulation hours a day and 672 simulation hours a week. If single student needs to use it 3 hours a day, VLAB can be used by 224 students [2]. However, VLAB being limited to a single server, resources were still scarce and usage had to be scheduled in advance. The VLAB server also represents a single point of failure because an unexpected outage renders all pending emulations useless and configurations are lost. We will cover the original architecture of the VLAB in the following chapter and address its issues in the chapter III.

II. Original Concept

Originally, the VLAB was conceived as a server-centric model where at the heart of the system was a single Unix-based server optimized for virtualization and emulation tasks. As per the requirements, the server had to support multiple concurrent dynamips hypervisor processes. User-mode Linux was deployed to provide PC emulation and together with the VLAB web-application, these three components formed the student interface. Students would log into the VLAB web-application and schedule a laboratory lesson, which would become available to them via a combination of the Server IP address and TCP port mapped to the instance of an emulated device running on the Server. The whole concept is depicted in Fig.1 below.

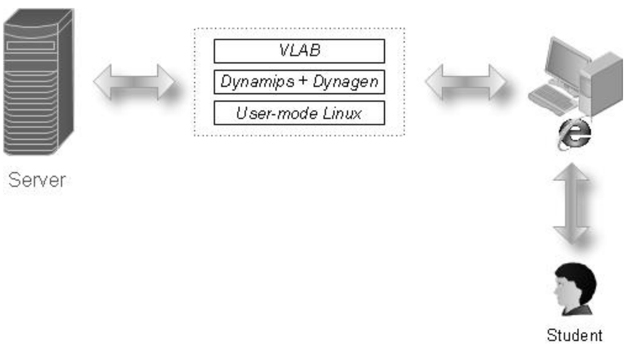

Figure 1. A high-level overview of the VLAB

After a laboratory lesson is selected and scheduled, the topology together with an initial configuration is displayed to the student (see Fig.2 below).

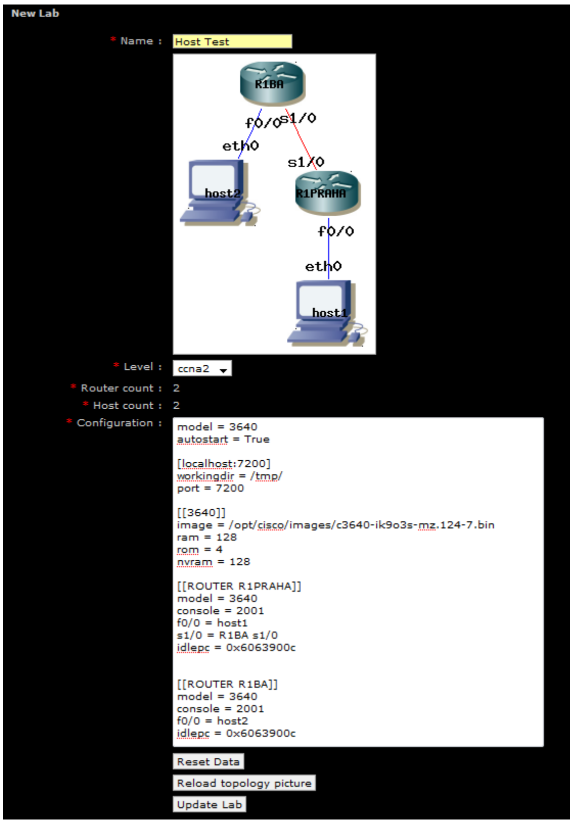

Figure 2. A loaded laboratory lesson in the VLAB web-application

The task of configuring the emulated devices itself can then be done my means of an applet which has established connections to all of the emulated devices (see Fig. 3) or by directly establishing secure terminal session (SSH) to the VLAB Server.

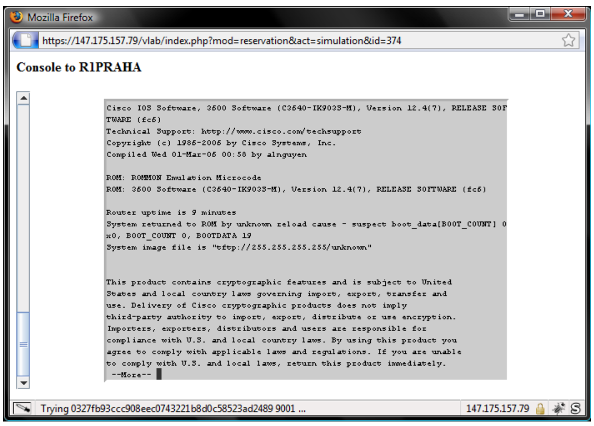

Figure 3. The configuration applet in the VLAB web-application

A major advantage of the VLAB, above those already listed, is the ability to automatically score a completed laboratory lesson after the student has finished. This scoring is done in a twofold manner. First the configurations are compared to correct reference configurations and secondly, the entire topology is functionally tested to determine whether connectivity was established as desired.

III. PROPOSED CONCEPT

To address the shortcomings of the VLAB – mainly susceptibility to being overloaded with increasing number of students, the upper limit on the number of emulated devices and the fact that it represents a single point of failure, we propose a cloud-based virtual network laboratory.

A cloud computing environment might provide an ideal solution to the issues we face when delivering the Cisco CCNP Troubleshooting course. Dynamically allocated resources will serve well in the academic environment as, in our view, there are a lot computing resources available at universities, yet not many of those are utilized to their maximum potential at all times. For our purposes we have chosen the Ubuntu Enterprise Cloud (only UEC further on), because it provides an easy to use open-source virtualization capability, applications and flexibility to help deploy a cloud within an organization [3]. It is important to note that UEC is compatible with Amazon EC2 cloud. The UEC is composed of the following key components:

- **eucalyptus-cloud** – cloud controller, front-end services
- **eucalyptus-cc** - includes the Cluster Controller that provides support for the virtual network overlay
- **eucalyptus-sc** - includes the Storage Controller
- **eucalyptus-walrus** - includes the Walrus storage system
- **eucalyptus-nc** - includes the Node Controller that interacts with KVM through Libvirt to manage individual VMs

We have modified the original VLAB architecture (see Fig. 1) and broke it up into two layers: bottom and top (see Fig. 2). The bottom layer contains network nodes which are part of the cloud. There should be a minimum of 2 network nodes in the cloud active at any given time. One of these nodes will be dedicated as a Cloud Controller which will manage all the other nodes, virtual machine instances and available images and resources inside of the cloud. Other nodes in the cloud will function as resources of computing power and storage space, hosting the virtual machine. These nodes are running Ubuntu Linux Server, which provides a scalable and manageable cloud computing solution. The number of nodes in the cloud is not limited which is important for future growth.

The top layer in our architecture is composed of the actual virtual machine. Users use cloud resources by an interface composed of our existing VLAB, a combination of Dyn@ng and Mindterm which is a Java SSH client applet [4, 5]. It is important to note at this point, that all of the components we have chosen are free and in the public domain. We have integrated these components with little effort and only minor changes were made to the existing VLAB environment. The Fig.4 below represents a high level overview of our proposed architecture.

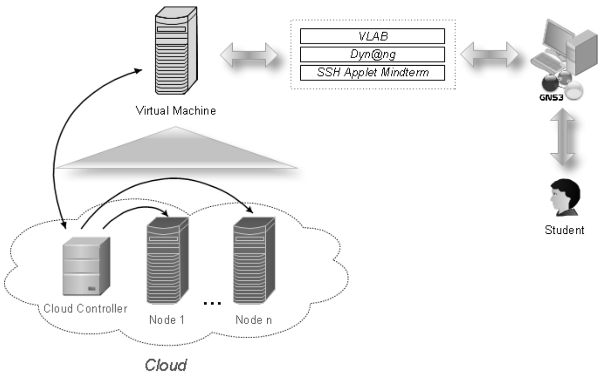

Figure 4. A high-level overview of the proposed architecture

It is important to merge all the components of the user interface into an easy to use, reliable and secure component of our architecture.

Dyn@NG was chosen instead of Dynagen because of the immense advantage it offers by offloading running emulations. This in turn enables us to run a large number of concurrent emulations. Another advantage is the fact that students now don't have to have user accounts on the virtual machine (Server) to gain access to their running virtual laboratory. During implementation we have identified a security imperfection of this approach. With Dyn@NG, the resources are becoming available to everyone, because to enable load imposed by the emulation from client to the server to be moved, we need to make the ports on which the hypervisor and device consoles are running available. This situation can be partly resolved by dynamic firewall measures limiting the availability of those ports to only a selected IP address.

Mindterm was chosen to tunnel ports from particular cloud node which is running the emulation to the student's workstation. From the student's point of view, the whole process of connection establishment is transparent. Mindterm, now integrated into our VLAB web-application will, after laboratory lesson start and resource allocation, establish a secure (SSH) session with the virtual machine and configure SSH tunnels as mentioned above. To integrate the Mindterm applet into our VLAN web-application, the following code is crucial and present in every student's running laboratory lesson:

```
<APPLET
CODE="com.mindbright.application.MindTerm.class"
  CODEBASE="./" WIDTH=0 HEIGHT=0>
      <PARAM NAME="archive"
VALUE="mindterm.jar"
      <PARAM NAME="sepframe" VALUE="true">
      <PARAM NAME="debug" VALUE="false">
      <PARAM NAME="protocol" VALUE="ssh2">
      <PARAM NAME="server"
VALUE="vlab.fiit.stuba.sk">
      <PARAM NAME="port" VALUE="22">
      <PARAM NAME="username"
VALUE="logged_in_username">
      <PARAM NAME="password"
VALUE="logged_in_password">
      <PARAM NAME="quiet" VALUE="true">
      <PARAM NAME="alive" VALUE="20">
      <PARAM NAME="term-type"
VALUE="xterm-color">
      <PARAM NAME="geometry"  VALUE="80x24">
      <PARAM NAME="local0"
VALUE="/general/127.0.0.1:7202:localhost:7202">
      <PARAM NAME="local1"
VALUE="/general/127.0.0.1:1051:localhost:1051">
      <PARAM NAME="local1"
VALUE="/general/127.0.0.1:2006:localhost:2006">
  </APPLET>
```

The above underlined sections of code are dynamically generated.

The hereby proposed and described cloud-based virtual laboratory was successfully implemented and the preliminary results from our testing are listed in the following chapter.

IV. Preliminary Results

We have conducted preliminary testing on our proposed architecture, focusing on the ability of Dyn@NG to spread requested emulations on available resources in the cloud. On the below figure (Fig. 5), the load imposed on the virtual machine is plotted over the course of the day. The load decreases as more resources are allocated inside of the cloud.

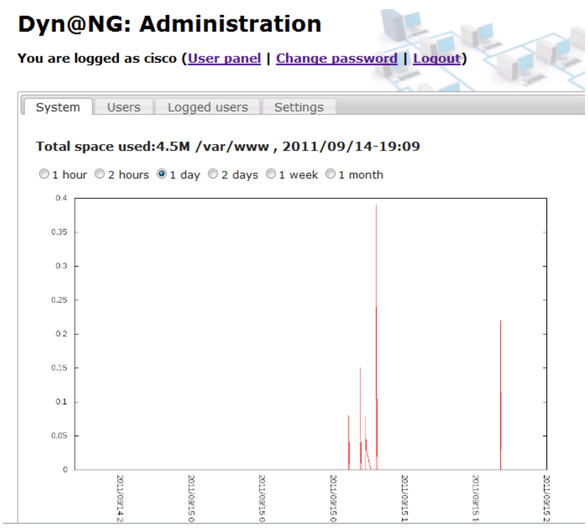

Figure 5. Plotted one-day load imposed on the virtual machine

On the below figure (Fig. 6), please observe the port forwarding, as established during a student session. Note, that the student does not have to have an active account on the virtual machine.

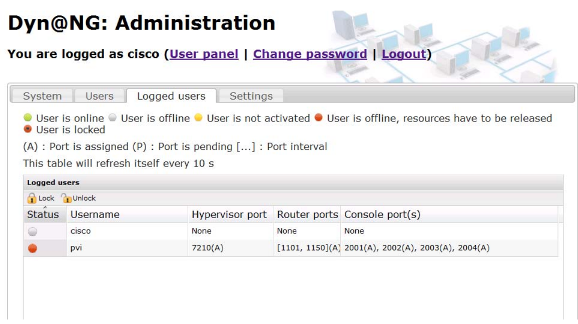

Figure 6. Port forwarding as seen in the administration of Dyn@NG

Since the running laboratory lessons are being emulated on resources inside of the cloud, the student doesn't have access to packet dumps. Therefore we utilize the Dyn@NG capture functionality to provide the packet capture function to the student via the VLAB web-application (as seen on Fig. 7 below).

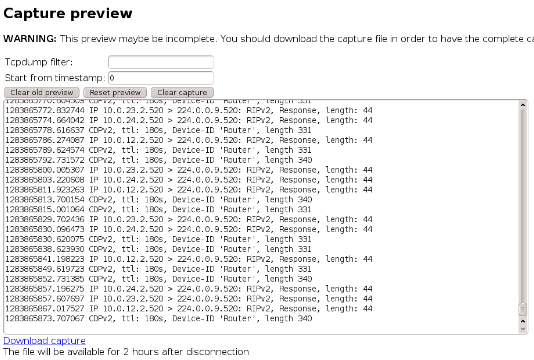

Figure 7. Dyn@NG packet capture feature

We have experimented with creating multiple simultaneous student sessions in our implemented cloud-based virtual laboratory. We have also integrated the Dyn@NG packet capture feature with our existing VLAB web-application to allow the students to implement packet sniffing on their running emulated topologies. This might greatly improve the learning experience, because in a real wired network laboratory it is not possible to capture packets on serial interfaces. Therefore, solutions to some networking issues might become more apparent to the student than without the ability to observe network traffic directly. At present time we have not concluded our testing of the proposed cloud-based virtual laboratory. Current testing however shows signs of great improvement over our previously implemented VLAB environment.

V. Comparison

In this chapter we compare our newly proposed solution with the original VLAB. The cloud-based solution is, in our view, a modern approach to solving issues where large amounts of computer resources need to be accessed quickly, securely and reliably. We believe that our proposed architecture which integrates the UEC, Mindterm, Dyn@NG and our existing VLAB is a good approach when it comes to emulating the network topologies we are faced with in the Cisco CCNP Troubleshooting course. We have not yet finished our testing of our proposed concept, yet we believe that the great improvement will, at least performance-wise, help us in delivering high-quality virtualized courses.

Conclusion

We have proposed a cloud-based virtual laboratory, building on our previous single-server virtual laboratory, in order to deliver the new Cisco CCNP Troubleshooting course. To accomplish this task we utilized existing approaches to cloud-based computing (UEC) and available secure terminals (Mindterm) to make the best use of load-balancing Cisco IOS emulation (Dyn@NG). The biggest advantages of our proposed solution are high availability of computing resources, scalability to allow satisfaction of possible future high demands and most importantly, a better learning experience for the students.

Acknowledgement: The support by Slovak Science Grant Agency (VEGA 1/0649/09 "Security and reliability in distributed computer systems and mobile computer networks") is gratefully acknowledged.

References

[1] I. Grellneth, M. Bielik, S. Gula, P. Marko, V. Michalec, P. Peti, and Z. Schusterova, "VLAB at the Institute of Computer Systems and Networks FIIT STU" *Acta Electrotechnica et Informatica*, vol. 10, pp. 65-71, April-June 2010

[2] I. Grellneth, Z. Schusterova, "An Aproach to Implementation of Virtual Network Laboratory" *In Proceedings of the 10th International Conference Virtual University VU'09*, December 10-11, 2009, Bratislava, Slovakia. ISBN - 978-80-89316-12

[3] UEC/Architecture [online], cited on September 14, 2011 https://help.ubuntu.com/community/UEC/Architecture

[4] Dyn@NG [online], cited on September 14, 2011 http://netgroup.polito.it/tools/dyn-ng

[5] Mindterm – World-class Java SSH2 Client [online], cited on September 14, 2011 http://www.cryptzone.com/products/agmindterm/