Wiesław Wolny
Mateusz Szołtysik

University of Economics in Katowice

OVERVIEW OF EXISTING COMPUTER NETWORK ENVIRONMENTS VIRTUALIZATION FOR COMPUTER NETWORK LEARNING

Introduction

Learning computer networks without performing practical experiments is really difficult, not to say it is almost impossible. Unfortunately, setting up a networking lab can be very expensive. Virtualization of computer networks allows setting up and performing networking experiments at low cost and with little effort. It allows to "create" several virtual network devices (full-fledged routers, switches, hosts, etc.) that can be easily interconnected in order to form a network on a single PC [GnVb14]. Networking equipments are virtual but feature many of the characteristics of the real ones, including the configuration interface.

Also testing configurations is a common need both for network administrators and for computer scientists interested in networking. The first can take advantage of a testing phase for checking that a particular configuration works as expected before deploying it, while the latter can exploit test results in order to validate theoretical models with practical experimentation. Ideally, testing should take place under the very same conditions in which the configuration is to be eventually deployed. However, this often means injecting artificially generated, potentially harmful traffic into a live network, which may cause damage to it. An effective alternative to live testing consists in implementing the network configuration of interest inside a safe, isolated software environment which closely reproduces the real target setting [GnWs14].

This paper introduces to the fundamental concepts of computer virtualization and presents concept of developed virtual environment software that enables teaching and experimenting networking on a personal computers.

An overview of Emulation Environments

An emulator is a software or hardware or both environment that that duplicates (or emulates) the functions of one computer system (the guest) in another computer system (the host), different from the first one, so that the emulated behavior closely resembles the behavior of the real system (the guest). A virtual machine is a running software that creates an abstraction layer between a hardware/software platform and other software (possibly an operating system). Applications running inside a virtual machine interface with this abstraction layer instead of the physical hardware. Therefore, a virtual machine may also implement virtual devices (disks, network interfaces, etc.) that are different from those available on the platform the emulator runs on.

There are a lot of emulation products available, which can be distinguished on the basis of the emulation technique adopted, of the type of device they emulate, and of the license with which they are distributed. This Section attempts to provide a taxonomy of emulators, with the purpose of outlining the landscape of available alternatives and of pointing out those products that are more network oriented. Table 1 shows a selected list of the existing emulation systems.

Table 1

Selected emulation related systems

Name	Emulator type	Emulated device	Capabilities	License
Emulab	Testbed	-	Large	-
Modelnet	Testbed	Linux box	Large	GPL/BSD
Netkit	User-mode kernel	Linux box	Medium	GPL
Planetlab	Overlay network	-	Large	Membership
UMLMON	User-mode kernel	Linux box	Medium	GPL
VDE	Overlay network	-	Large	GPL
VirtualBox	Full virtualization	x86 box	Small	GPL/Commercial
Virtual PC	Full virtualization	x86 box	Small	Free
VMware	Full virtualization	x86 box	Small	Commercial
VNUML	User-mode kernel	Linux box	Medium	GPL
Xen	Paravirtualization	x86 box	Medium	GPL/Commercial

The proposed classification coordinates have the following meaning:

- Emulation type specifies the technique used for virtualizing resources [Rimo07]:
  - *Full virtualization* [RouV14] indicates that the emulated entity is a full-fledged system consisting of system buses, CPU, memory, disk, and other devices, and that optimization techniques are used to improve the performance of the emulation. Among these techniques dynamic translation

is often used, which consists in translating blocks of binary code being executed in the emulated machine into instructions for the real host, and in caching the translated pieces of code for future execution. Full emulation adopts a complementary approach, in which every instruction of the emulated CPU is implemented as an entire function or procedure in the emulator. While this ensures compatibility and makes it easier to debug the code of the emulator, performance is severely impacted by this technique, and a high-end workstation is usually needed to achieve nearly native speed emulation. Native virtualization takes place whenever the emulator takes advantage of extensions available on recent families of processors (Intel VT [Inte14], AMD-V [AMDV14]), which allow a more effective distribution of resources between the emulated machine and the host it runs on, thus achieving much better performance.

  - In a *paravirtualization* [RouP14] environment each virtual entity is presented a special hardware abstraction layer. Virtual machines must run slightly modified versions of the standard operating systems, so that system calls are submitted to this abstraction layer (called hypervisor or virtual machine monitor) instead of the host operating system.
  - Some products exploit a user-mode kernel, often called User-Mode Linux [UMLK14, Dike06], which is a slightly modified version of a standard Linux kernel that is compiled to run as a userspace process. An instance of User-Mode Linux uses its own filesystem image and allocates a subset of the memory available on the hosting machine. A User Mode Linux kernel can start and schedule processes on its own and has its own virtual memory manager as well as every other kernel subsystem. Device drivers are suitably rewritten so that User Mode Linux can provide some support for virtualized hardware (disks, network interfaces, consoles). Differently from other emulation systems, User-Mode Linux does not directly interface with the hardware but achieves virtualization based on the system calls interface provided by the standard kernel.
  - Testbeds and overlay networks are large scale environments consisting of tens or hundreds of servers that can be geographically distributed (in which case they are often connected so as to form an overlay network). Such large scale architectures can be typically accessed by research institutions to perform controlled experiments on a realistic setting.

- The emulated device specifies the machine whose features are reproduced by the emulator. In most cases it is a standard PC which, by running suitable pieces of software, can be turned to a router, switch, or other network device.

- The capabilities describes the number of virtual entities (hosts, routers, switches, or whatever else) each emulator allows to start on standard workstation. A small scale emulator is usually conceived for running very few instances of virtual machines, as their resource requirements may be rather high. A large scale emulator is usually designed to run on a distributed architecture (possibly a cluster of geographically distributed workstations connected by an overlay network), which allows to perform arbitrarily wide-ranging experiments. Medium scale emulators typically allow to run around tens of virtual machines on a single workstation.
- License specifies the license agreement under which the emulator is being distributed.

Not all the products listed in Table 1 are tailored for performing networking experiments. Some of them are general purpose emulators that allow to run entire operating systems. We are more interested in the environments that provide configuration capabilities and tools to ease building and running emulated networks.

Virtual Distributed Ethernet (VDE) and Xen are components that are often used in emulation environments. Virtual Distributed Ethernet [VDE14] is a set of tools to create and manage a virtual network that can be spawned over a set of arbitrarily distributed physical computers. VDE can be used to handle tunnels that separate actual connectivity from the topology established by VDE virtual cables, thus providing with the ability to transparently distribute local network experiences on different nodes. Xen [Xen14] is a virtual machine monitor for x86 that supports execution of multiple guest operating systems with unprecedented levels of performance and resource isolation. It consists of a kernel patch and some userspace tools. Because it directly interfaces with hardware resources, using a Xen enabled kernel allows to run virtual machines (also called domains) with very high performance levels. Operating systems cannot run unmodified inside Xen domains, as the system calls they make must be mapped to software traps to the hypervisor. Yet, there are plans to extend Xen to support virtualization technologies found on recent processors (Intel VT [Inte14] and, in the future, AMD-V [AMDV14]), which would provide with the ability to run unmodified software.

Emulab, Modelnet, PlanetLab, and VINI are emulators based on the Xen hypervisor. They provides tools to setup and run switches, routers, and computers, possibly running some services. They all are large scale testbeds which allow to run experiments involving a set of geographically distributed nodes. The University of Utah makes available to researchers Emulab [Utah14], a cluster of networked high-end workstations that can be configured to run fully customizable tests. Emulab is widely used by computer science researchers in the fields of networking and distributed systems. It is also designed to support education, and has been used to teach classes in those fields. Modelnet [UCal14] is a software emulator developed at the University of California, San Diego. It is a large-scale network emulator that allows users to evaluate distributed networked systems in realistic Internet-like environments. ModelNet enables the testing of unmodified prototypes running over unmodified operating systems across various networking scenarios. In some sense, it combines the repeatability of simulation with the realism of live deployment. PlanetLab [Plan14] is a global research network that supports the development of new network services. Since the beginning of 2003, more than 1,000 researchers at top academic institutions and industrial research labs have used PlanetLab to develop new technologies for distributed storage, network mapping, peer-to-peer systems, distributed hash tables, and query processing. PlanetLab at the time this paper is being written consists of 1188 nodes at 586 sites and is managed by a consortium of academic, industrial, and government institutions.

Netkit, UMLMON, and VNUML are all medium scale software emulators that utilize a User-Mode Linux kernel to run the emulated network experiences. Netkit [Netk14] is extensively described in the following of this paper. UMLMON [Stol14] is a solution for managing a set of User-Mode Linux virtual machines. Gerd Stolpmann offers UMLMON, which is an add-on product to User Mode Linux that strongly simplifies its administration. Up to now a system operator needed several weeks to develop scripts in order to deploy User Mode Linux. UMLMON is a ready-to-use solution setting standards for User Mode Linux operating, especially concerning the number of features and the operational safety. VNUML [Gala04] consists of an XML (Extensible Markup Language) based language and an interpreter that can be used to describe and run an emulated network of User-Mode Linux virtual machines. VNUML developers also propose an interesting set of ready to use examples that implement some typical networking case studies.

Virtual Box, VMware and Virtual PC are virtualization products aimed to enterprise and home use. Oracle Virtual Box is a powerful x86 and AMD64/Intel64 virtualization product. Presently, VirtualBox runs on Windows, Linux, Macintosh, and Solaris hosts and supports a large number of guest operating systems. VirtualBox is being actively developed with frequent releases and has an ever growing list of features, supported guest operating systems and platforms it runs on. VirtualBox is a community effort backed by a dedicated company: everyone is encouraged to contribute while Oracle ensures the product always meets professional quality criteria. VMware [VMwa14] is the industry-leading virtualization software company and offers data center, desktop and personal virtualization products. VMware Player Plus is used in developed environment to run virtual machines on student's desktop terminals. Windows Virtual PC [Micr14] is the latest Microsoft virtualization technology. It allows to run more than one operating system at the same time on one computer, and to run many productivity applications on a virtual Windows environment.

The most important virtualization software selection criterion was license, capabilities and compatibility between used programs and emulators. As the basis for creating the system The Live Raizo was chosen.

The architecture of system

The Live Raizo is a Linux LiveCD distribution based on Debian Linux distribution with addition of Fluxbox window manager as default user desktop environment. This is a continuously developed Open Source project, used by us at the network laboratories.

This paper is based on the latest Live Raizo version available at time of writing and preparing classes for the new semester, which is version 5.14.02.02. It was released with 3.12-1 Linux kernel version, compiled for 686 CPUs architecture with added PAE (Physical Address Extension) support. That attribute allow 32-bit CPUs to access a physical address space more than 4GB. The reviewed Live Raizo version is based on Debian GNU/Linux from experimental and development branch named Jessie/Sid [SoLi14].

In the Live Raizo are also present 18 virtual machine instances, each of them with stable distribution Debian 7 (codename "wheezy") and 3.12-0 Linux kernel version. Similarly as the main Live Raizo's system also with the PAE support and the same CPU's architecture. Each of these virtual machines are the same, the only difference between them are their names:

- 6 personal computers,
- 6 routers,
- 6 servers.

All existing virtual machines are integrated with Oracle VM VirtualBox and GNS3 network simulator.

For our own purposes we modified the project's original ISO image file and we added Polish language translation to desktop environment. We changed boot menu to include options with that translation and link Cisco IOS (Internetwork Operating System) image file with configuration of GNS 3 simulator.

Simulators (GNS3)

GNS3 (Graphical Network Simulator) is an open source software released under GNU General Public License version 2.0 (GPLv2) [SoGn14]. It requires Dynamips, because it serves as a GUI front-end for that library and for Cisco IOS emulation Dynamips is necessary. GNS3 enables simulation of complex networks, close as possible to the performance of real, non-emulated networks. This is possible without having dedicated network hardware, e.g. routers and switches.

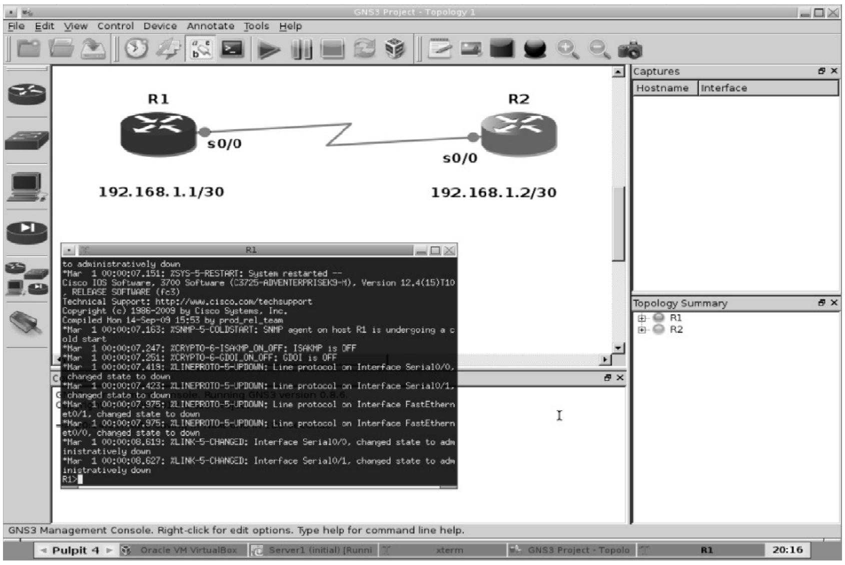
GNS3 is available for the computers with x86 architecture and most important operating systems, e.g. Microsoft Windows, GNU/Linux distributions and Apple's OS X.

For the providing a complete and accurate network simulations, GNS3 uses programs and emulators listed below:

- Dynamips – the Cisco IOS emulator, described in the next section,
- Dynagen – the text based user interface for Dynamips, also mentioned in the next section,
- Oracle VM VirtualBox – free virtualization software, responsible for running desktop and server operating systems as in real computer networks, as well as Juniper JunOS,
- Qemu – abbreviation for Quick EMUlator, it's a generic, free and open source emulator, virtualization environment and hypervisor,
- Wireshark – free and open source network packet analyzer. It is described in a separate section of this paper [DoJe2014, GnIn14].

GNS3 provides us an ability to design and configure network topologies (Figure 1). We can create a new topology project, add the required devices, which can be connected together by the use of suitable transmission medium to their relevant network interfaces. The new topology's creation process is very simple, based on drag and drop gesture. Later we are able to change and save our previously created network topologies.

In the reviewed Live Raizo image is included GNS3 in version 0.8.6.



```
to administratively down
*Mar  1 00:00:07.151: %SYS-5-RESTART: System restarted --
Cisco IOS Software, 3700 Software (C3725-ADVENTERPRISEK9-M), Version 12.4(15)T10
, RELEASE SOFTWARE (fc3)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2009 by Cisco Systems, Inc.
Compiled Mon 14-Sep-09 15:53 by prod_rel_team
*Mar  1 00:00:07.163: %SNMP-5-COLDSTART: SNMP agent on host R1 is undergoing a c
old start
*Mar  1 00:00:07.247: %CRYPTO-6-ISAKMP_ON_OFF: ISAKMP is OFF
*Mar  1 00:00:07.251: %CRYPTO-6-GDOI_ON_OFF: GDOI is OFF
*Mar  1 00:00:07.419: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0,
 changed state to down
*Mar  1 00:00:07.423: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/1,
 changed state to down
*Mar  1 00:00:07.975: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthern
et0/1, changed state to down
*Mar  1 00:00:07.975: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthern
et0/0, changed state to down
*Mar  1 00:00:08.619: %LINK-5-CHANGED: Interface Serial0/0, changed state to adm
inistratively down
*Mar  1 00:00:08.627: %LINK-5-CHANGED: Interface Serial0/1, changed state to adm
inistratively down
R1>
```

Fig. 1. GNS3 simulator running simple topology with two Cisco c3700 routers

Source: Own work based on existing program.

Dynamips and Dynagen

Dynamips is a free, open source software that is responsible for emulation of Cisco IOS systems on a traditional x86 PC with Windows or GNU/Linux operating systems and Apple Inc.'s computers with OS X operating system as well. The project has been started by Christophe Fillot in August 2005. From the 2007 project is developed by the GNS3 community.

It allows us to create virtual routers that can run a real Cisco IOS (Internetwork Operating System) by the use of the host computer resources. Only the IOS images compatible with supported hardware can be emulated by the Dynamips software.

Dynamips supports the following Cisco platforms:

- Cisco 1700 Series (1710, 1720, 1721, 1750, 1751 and 1760 routers),
- Cisco 2600 Series (2610, 2611, 2610XM, 2620, 2620XM, 2650XM, 2611XM, 2621, 2621XM and 2651XM routers),
- Cisco 3600 Series (3620, 3640, 3660 routers),
- Cisco 3700 Series (3725 and 3745 routers),
- Cisco 7200 Series (7206 routers) [GnHa14].

Dynagen it is a text based user interface, front-end for Dynamips emulator. It was oriented to the creation of previously prepared network scenarios of emulated Cisco routers, based on the Dynamips technology [DoJe2014].

Netkit

Netkit is an another feature available in Live Raizo image. It is free, open source and lightweight network emulator [NetW14]. Netkit includes four basic components:

- kernel,
- system file image,
- software for virtual hub,
- set of defined user commands [DoJe2014].

With this tool the process of creating, connecting and testing virtual network devices using UML (User-mode Linux) is simplified. Technology of User Mode Linux allows multiple instances of virtual Linux operating systems to be launched on the same host Linux operating system, with each instance of virtual systems receiving default Linux functionality. In the other words, User Mode Linux is a port of the Linux kernel, developed to run as process in the user process space. In Netkit the network devices are emulated as User Mode Linux virtual machines. Each of them run a Debian GNU/Linux operating system, which is full-fledged version of this Linux distribution. To manage the process of configuring and turning on a User Mode Linux virtual system, Netkit delivers a set of utilities [NetW14, NgRo11].

Netkit enables users an interface to set up emulated network experiences (labs), gives possibility to experiment with several networking technologies without need of editing the emulation parameters instead of configuring the laboratory's network itself. That is possible with no influence on real computer network. Support for sharing preconfigured virtual labs with other users is also present. If the emulated network has been defined in configuration file, the Netkit takes care of generating and launching the User Mode Linux virtual system. That virtual system can be converted into a specific network device, e.g. a router. It is possible by running appropriate software on this particular User Mode Linux virtual machine [NetW14, NgRo11].

It is also possible to describe the networks by using an XML-based language known as NetML (Network Markup Language) [NetM14]. By the use of NetML for description network, we can automatically receive configuration files which can be used with the real routers, or Netkit scripts which can be used to emulate network, that we described [NetW14].

Servers DHCP, DNS, FTP, tFTP, SSH

In environment of the Live Raizo project we are capable to test the most popular network services, which may often be difficult to perform in the academic setting. Each of available type of service is deactivated by default and will not start automatically. They must be started manually by providing the relevant commands, e.g. *sudo service ssh start*.

DHCP (Dynamic Host Configuration Protocol) and **DNS (Domain Name System)** services are implemented by the lightweight **dnsmasq** tool. Besides the two mentioned services it is responsible also for router advertisement and network boot and provides full IPv6 support. It is available on platforms: GNU/Linux, Android, *BSD, and Mac OS X. Dnsmasq is included in most GNU/Linux distributions and the ports systems of FreeBSD, OpenBSD and NetBSD [Dnsm14].

The DNS subsystem of dnsmasq provides a local DNS server for the emulated network. Local DNS names can be defined:

- by reading/etc/hosts file,
- by importing names from the DHCP subsystem,
- by configuration of a wide range of useful record types.

Internationalised domain names are also supported in the dnsmsq [Dnsm14].

The DHCP subsystem supports:

- DHCPv4,
- DHCPv6,
- BOOTP,
- PXE (Preboot eXecution Environment or Pre-Execution Environment).

Both static and dynamic DHCP leases are supported by the dnsmsq [Dnsm14].

FTP (File Transfer Protocol)

In Live Raizo the FTP service is delivered by two different projects: vsftpd [Vsft14] and ProFTPD [Prof14].

vsftpd (abbreviation for Very Secure FTP Daemon) is a GPL licensed FTP server for UNIX systems, including Linux. Several supported features by vsftpd:

- Virtual IP configurations,
- Virtual users,
- Bandwidth throttling,
- Powerful per-user configurability,

- Per-source-IP configurability,
- Per-source-IP limits,
- Encryption support through SSL integration [Vsft14].

ProFTPD (abbreviation for Professional FTP Daemon) is a highly configurable FTP server software released under GPL license, written for use on Unix and Unix-a-like operating systems. There is no support for native use under Microsoft Windows operating systems. It offers the features listed below:

- single main configuration file,
- easy to configure multiple virtual FTP servers and anonymous FTP services,
- hidden directories and files, based on Unix-style permissions or user/group ownership,
- anonymous FTP root directories do not require any specific directory structure, system binaries or other system files,
- shadow password suite support, including support for expired accounts [Prof14].

SSH (Secure Shell)

The Secure Shell service in Live Raizo is delivered by the **OpenSSH** (abbreviation for OpenBSD Secure Shell), a free and open source software. It is a free SSH/SecSH protocol suite, that provide encryption for network services, e.g. remote login or remote file transfer. OpenSSH is developed by the OpenBSD project and released under the BSD license [Open14].

Several OpenSSH features:

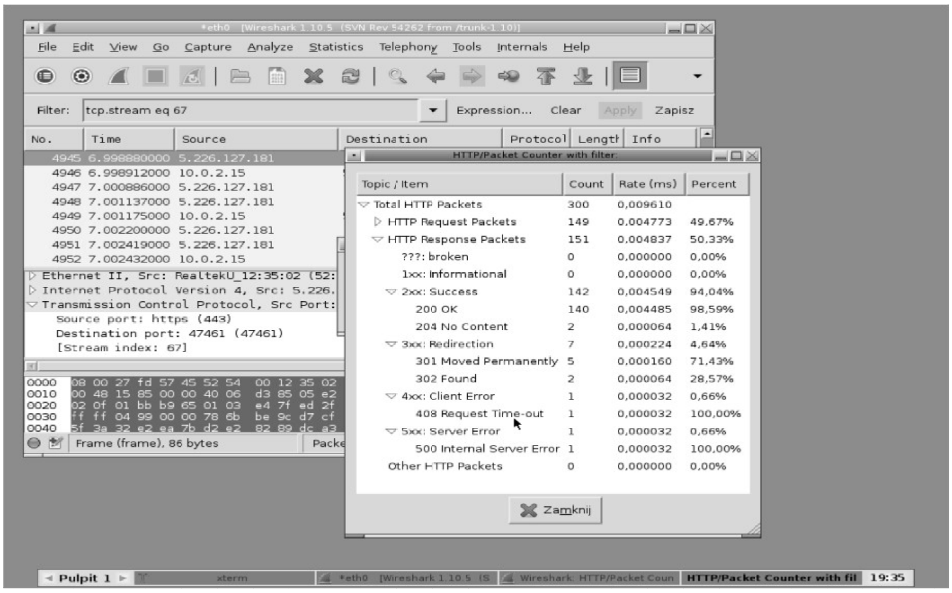
- Strong Encryption (3DES, Blowfish, AES, Arcfour),
- Strong Authentication (Public Key, One-Time Password and Kerberos Authentication),
- Port Forwarding (encrypted channels for legacy protocols),
- Interoperability (Compliance with SSH 1.3, 1.5, and 2.0 protocol Standards),
- SFTP client and server support in both SSH1 and SSH2 protocols,
- Data Compression [Open14].

Network protocol analyzer: Wireshark

Wireshark, originally named Ethereal is a free and open source software that allows to analyze network protocols. The origins of the project date back to 1998, with the original creator Gerald Combs. It is released under the GNU General Public License version 2 [Wiaw14, Wifq14].

This software enables capture and interactively browse the traffic running on a computer network. It offers advanced options for filtering the captured network traffic (Figure 2). In category of network protocol analysis utilities, Wireshark is world's most popular tool. It runs on most computing platforms, including:

- Microsoft Windows,
- OS X,
- GNU/Linux,
- UNIX [Wiaw14, Wifq14].


*eth0 [Wireshark 1.10.5 (SVN Rev 54262 from /trunk-1.10)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Expression... Clear Apply Zapisz
HTTP/Packet Counter with filter:
Zamknij
Pulpit 1
xterm
*eth0 [Wireshark 1.10.5 (S
Wireshark: HTTP/Packet Coun
HTTP/Packet Counter with fil
19:35

Filter: tcp.stream eq 67

```
No.   Time         Source          Destination   Protocol Length Info
4945 6.998880000 5.226.127.181
4946 6.998912000 10.0.2.15
4947 7.000886000 5.226.127.181
4948 7.001137000 5.226.127.181
4949 7.001175000 10.0.2.15
4950 7.002200000 5.226.127.181
4951 7.002419000 5.226.127.181
4952 7.002432000 10.0.2.15
▷ Ethernet II, Src: RealtekU_12:35:02 (52:
▷ Internet Protocol Version 4, Src: 5.226.
▽ Transmission Control Protocol, Src Port:
    Source port: https (443)
    Destination port: 47461 (47461)
    [Stream index: 67]
0000  08 00 27 fd 57 45 52 54  00 12 35 02
0010  00 48 15 85 00 00 40 06  d3 85 05 e2
0020  02 0f 01 bb b9 65 01 03  e4 7f ed 2f
0030  ff ff 04 99 00 00 78 6b  be 9c d7 cf
0040  5f 3a 32 e2 ea 7b d2 e2  82 89 dc a3
Frame (frame), 86 bytes    Packe
```

Topic / Item	Count	Rate (ms)	Percent
▽ Total HTTP Packets	300	0,009610	
▷ HTTP Request Packets	149	0,004773	49,67%
▽ HTTP Response Packets	151	0,004837	50,33%
???: broken	0	0,000000	0,00%
1xx: Informational	0	0,000000	0,00%
▽ 2xx: Success	142	0,004549	94,04%
200 OK	140	0,004485	98,59%
204 No Content	2	0,000064	1,41%
▽ 3xx: Redirection	7	0,000224	4,64%
301 Moved Permanently	5	0,000160	71,43%
302 Found	2	0,000064	28,57%
▽ 4xx: Client Error	1	0,000032	0,66%
408 Request Time-out	1	0,000032	100,00%
▽ 5xx: Server Error	1	0,000032	0,66%
500 Internal Server Error	1	0,000032	100,00%
Other HTTP Packets	0	0,000000	0,00%

Fig. 2. Wireshark capturing traffic from interface eth0. Additional packet counter window with statistics about occurred HTTP Response codes

Source: Own work based on existing program.

Several features available and offered by Wireshark:

- live reading data from Ethernet, IEEE 802.11, PPP/HDLC, ATM, Bluetooth, USB, Token Ring, Frame Relay, FDDI, and others (depending on available platform),
- captured network data can be browsed by using the GUI, or via the TTY-mode in TShark utility,
- coloring rules can be applied to the packet list for faster, intuitive analysis,
- capture files compressed with gzip can be decompressed "on the fly",
- output files can be exported as XML, PostScript®, CSV, or plain text,
- decryption support for many protocols, including IPsec, ISAKMP, Kerberos, SNMPv3, SSL/TLS, WEP, and WPA/WPA2 [Wiaw14].

In addition to capturing traffic from active network interfaces in existing host environment, we can also import files with captured network traffic from other capture tools to perform offline analyze of the data. It supports dozens of input file formats [Wugi14], but the most popular file extension are:

- libpcap − captures from Wireshark/TShark/dumpcap, tcpdump, and various other utilities using the libpcap's/tcpdump's capture format,
- pcap-ng – "next-generation" successor to the libpcap format [Wugi14].

In this edition of Live Raizo, Wireshark is available in version 1.10.5.

Summary

The purpose of this paper was to describe the existing tool for providing the more interactive networking laboratories.

Virtual computer networks can be easier to implement and manage than physical networks. We can provide that type of virtual computer infrastructures to the students, without any concerns. All failures in virtual networks can be quickly removed by restoring them to its original state. That can't be possible in most cases in the real computer networks. Therefore these tools allow us to extend the range of available topics and also provide ability to introduce more risky subject, which are difficult to implement in real academic computer network setting. The Live Raizo project isn't something, that we must invent and develop. This is a existing solution, that can be adapted for your own purposes and ready to implementation. In the near future we are planning introduce Live Raizo solution in virtualized environment of new building at our university. We want every student at the computer networking classes should be able to run prepared image from virtualization server right on their "thin client" computer terminal.

References

[AMDV14] AMD: AMD Virtualization, http://www.amd.com/en-us/solutions/servers/virtualization, retrieved May 20, 2014.

[Dike06] Dike J.: User Mode Linux. Prentice Hall, 2006.

[Dnsm14] Dnsmasq − Network Services for Small Networks, http://www.thekelleys.org.uk/dnsmasq/doc.html, retrieved May 10, 2014.

[DoJe2014] Dobrilovic D., Jevtic V., Odadzic B.: Virtualization Technology in Higher Education IT Courses. „Journal for Information Technology Education Development and Teaching Methods of Technical and Natural Sciences" 2012, Vol. 2(1), University of Novi Sad, Zrenjanin, Republic of Serbia, pp. 66-72.

[Gala04] Galan F., Fernandez D., Ruiz J., Walid O., de Miguel T.: Use of Virtualization Tools in Computer Network Laboratories. Proc. 5th International Conference on Information Technology Based Higher Education and Training (ITHET 2004), Jun 2004, pp. 209-214.

[GnHa14] GNS3 − Hardware emulated by GNS3, http://www.gns3.net/hardware-emulated/, retrieved May 15, 2014.

[GnIn14] GNS3 − Introduction to GNS3, http://www.gns3.net/documentation/gns3/introduction-to-gns3/, retrieved May 15, 2014.

[GnVb14] GNS3 − VirtualBox, http://www.gns3.net/documentation/gns3/virtualbox-emulation/, retrieved July 10, 2014.

[GnWs14] GNS3 – Packet Capture, http://www.gns3.net/documentation/gns3/packet-capture/, retrieved July 11, 2014.

[Inte14] Intel: Intel® Virtualization Transforms IT, http://www.intel.com/content/www/us/en/virtualization/intel-virtualization-transforms-it.html, retrieved May 20, 2014.

[Micr14] Microsoft: Windows Virtual PC, http://www.microsoft.com/en-US/download/details.aspx?id=3702, retrieved May 20, 2014.

[Netk14] Netkit, http://wiki.netkit.org, retrieved May 20, 2014.

[NetM14] <NetML/> − How Many Ways Are There to Describe a Computer Network? Maybe One…, http://www.dia.uniroma3.it/~compunet/netml/, retrieved May 15, 2014.

[NetW14] Netkit Wiki, http://wiki.netkit.org/index.php/Main_Page, retrieved February 20, 2014.

[NgRo11] Nguyen H., Roughan M., Knight S., Falkner N., Maennel O., Bush R.: How to Build Complex Large-Scale Emulated Networks, Testbeds and Research Infrastructures. Development of Networks and Communities, Springer, Berlin-Heidelberg 2011, pp. 3-18.

[Open14] OpenSSH Features, http://www.openssh.com/features.html, retrieved April 28, 2014.

[Orac14] Oracle: VM VirtualBox, https://www.virtualbox.org, retrieved May 20, 2014.

[Plan14] PlanetLab, http://www.planet-lab.org, retrieved May 20, 2014.

[Prof14] The ProFTPD Project: Home, http://www.proftpd.org/, retrieved April 28, 2014.

[Rimo07] Rimondini M.: Emulation of Computer Networks with Netkit. Universita degli Studi di Roma Tre, Roma 2007.

[RouV14] Rouse M.: What Is Virtualization? Definition from WhatIs.com, http://searchservervirtualization.techtarget.com/definition/virtualization, retrieved July 10, 2014.

[RouP14] Rouse M.: What Is Paravirtualization? Definition from WhatIs.com, http://searchservervirtualization.techtarget.com/definition/paravirtualization, retrieved July 10, 2014.

[SoGn14] Sourceforge.net – GNS3, http://sourceforge.net/projects/gns-3/, retrieved May 16, 2014.

[SoLi14] Sourceforge.net – Live Raizo, http://sourceforge.net/p/live-raizo/, retrieved February 10, 2014.

[Stol14] Stolpmann G.: UMLMON. http://www.gerd-stolpmann.de/buero/umlmon.html.en, retrieved May 20, 2014.

[UMLK14] The User-mode Linux Kernel Home Page, http://user-mode-linux.sourceforge.net/, retrieved May 20, 2014.

[Utah14] University of Utah: Emulab Network Emulation Testbed, http://www.emulab.net/, retrieved May 20, 2014.

[UCal14] University of California, San Diego Department of Computer Science: ModelNet, http://modelnet.ucsd.edu/, retrieved May 20, 2014.

[VDE14] VDE: Virtual Distributed Ethernet, http://sourceforge.net/projects/vde/, retrieved May 20, 2014.

[VMwa14] VMware, http://www.vmware.com, retrieved May 20, 2014.

[Vsft14] vsftpd – Secure, fast FTP server for UNIX-like systems, https://security.appspot.com/vsftpd.html, retrieved April 28, 2014.

[Wiaw14] Wireshark – About Wireshark, https://www.wireshark.org/about.html, retrieved May 24, 2014.

[Wifq14] Wireshark. Frequently Asked Question, https://www.wireshark.org/faq.html, retrieved May 24, 2014.

[Wiug14] Wireshark User's Guide – 5.2.2. Input File Formats, https://www.wireshark.org/docs/wsug_html_chunked/ChIOOpenSection.html#ChIOInputFormatsSection, retrieved May 24, 2014.

[Xen14] Xen: The Xen™ Virtual Machine Monitor, http://www.cl.cam.ac.uk/research/srg/netos/xen/, retrieved May 20, 2014.

PRZEGLĄD ISTNIEJĄCYCH ŚRODOWISK WIRTUALIZACJI SIECI KOMPUTEROWYCH I ICH ZASTOSOWANIE W NAUCZANIU

Streszczenie

W artykule przedstawiono koncepcję wykorzystania osiągnięć wirtualizacji do nauczania sieci komputerowych. Pierwsza część skupia się na omówieniu technologii wirtualizacji i emulacji oraz narzędzi, które umożliwiają wykorzystywanie tej technologii. Druga część zawiera opis zmodyfikowanego na własne potrzeby projektu Live Raizo oraz pozostałych narzędzi wchodzących w jego skład.