Maxim Fomin

# GNS3
# FOR NETWORK EMULATION

Bachelor's thesis
IT / Networking

2017

South-Eastern Finland
University of Applied Sciences

South-Eastern Finland University of Applied Sciences

| Author (authors) | Degree | Time |
|---|---|---|
| Maxim Fomin | Networking engineer | May 2017 |

**Title**

GNS3 for network emulation

60 pages
4 pages of appendices

**Commissioned by**

XAMK

**Supervisor**

Tomi Pahula


**Abstract**

This study was performed to determine possible usage cases of virtual networks. Modern network professionals are in need of evaluating and testing of different networking conditions. While it is possible to implement all of the situations with physical networks, it is in specific conditions faster, cheaper, and wiser to do this virtually. GNS3 is a virtual environment, which emulates and interconnects specific networking devices, including routers, firewalls, switches and other devices which are interconnectable within OSI model.

The objective of the thesis was to evaluate GNS3, test its ability to emulate networks in a beneficial way, provide solutions for possible problems which may occur during exploitation and writing instructions for possible future use.

GNS3 is able to launch on any x86 based computer with multiple OS families, for this thesis Windows OS family was used, as most convenient system for launching different software, needed to perform tests. The best way of analyzing GNS3 is to use it to create different network configurations and document the way it works.

During this research, all laboratory scenarios with routers were successfully performed and GNS3 showed itself capable of being a powerful tool for network emulation.



**Keywords**

GNS3, network emulation, virtual networks, cisco

| Tekijä/Tekijät | Tutkinto | Aika |
|---|---|---|
| Maxim Fomin | Insinööri (AMK) | Toukokuu 2017 |
| **Opinnäytetyön nimi**<br><br>GNS3 for network emulation | | 60 sivua<br>4 liitesivua |
| **Toimeksiantaja**<br><br>XAMK | | |
| **Ohjaaja**<br><br>Tomi Pahula | | |


**Tiivistelmä**

Tässä työssä tutkittiin virtuaaliverkkojen soveltavuutta ICT laboratorion opetusympäristössä. Nykyaikaiset verkko-ammattilaiset tarvitsevat eri verkko-olosuhteiden arviointia ja testaamista. Vaikka kaikki tilanteet voidaan toteuttaa fyysisten verkkojen kanssa, se on tietyissä olosuhteissa nopeampaa, halvempaa ja viisaampaa tehdä virtuaaliverkoissa. GNS3 on virtuaalinen ympäristö, joka emuloi ja liittää fyysisiä verkkolaitteita, mukaan lukien reitittimet, palomuurit, kytkimet ja muut laitteet, jotka ovat yhteydessä toisiinsa.

Opinnäytetyön tavoitteena oli arvioida GNS3:a ja kirjoittaa ohjeet mahdollisen tulevan käytön varalta. Arvioinnissa testattiin GNS3:n kykyä jäljitellä verkkoja hyödyllisellä tavalla ja tarjota ratkaisuja mahdollisiin ongelmiin, joita saattaa ilmetä käytön yhteydessä.

GNS3 pystyy käynnistymään millä tahansa x86-pohjaisella tietokoneella ja on yhteensopiva monien käyttöjärjestelmien kanssa. Tässä työssä käytettiin Windows-ympäristöä, koska se oli kaikista helppokäyttöisin järjestelmä erilaisten tarvittavien ohjelmistojen käynnistämiseksi. Paras tapa analysoida GNS3:a on käyttää sitä erilaisten verkkokokoonpanojen luomiseen ja dokumentoida tapa miten se toimii.

Tämän tutkimuksen aikana kaikki reitittimien kanssa suoritetut testit onnistuivat menestyksekkäästi ja GNS3 osoitti kykenevänsä olemaan tehokas verkon emuloinnin työkalu.



**Asiasanat**

GNS3, network emulation, virtual networks, cisco

**CONTENTS**

**ABBREVIATIONS**

| | |
|---|---|
| AS | Autonomous System |
| ASBR | Autonomous System Border Router |
| BGP | Border Gateway Protocol |
| CE | Customer Edge |
| CCIE | Cisco Certified Internetwork Expert |
| CCNA | Cisco Certified Network Associate |
| CCNP | Cisco Certified Network Professional |
| CDP | Cisco Discovery Protocol |
| CPU | Central Processing Unit |
| Dot1Q | IEEE 802.1Q encapsulation |
| eBGP | Exterior Border Gateway Protocol |
| IEEE | Institute of Electrical and Electronics Engineers |
| EIGRP | Enhanced Interior Gateway Routing Protocol |
| ESXi | Elastic Sky X integrated |
| GUI | Graphical user interface |
| iBGP | Internal Border Gateway Protocol |
| ICMP | Internet Control Message Protocol |
| IOS | Internetwork Operating System |
| IOSv | Internetwork Operating System (virtual) |
| IP | Internet Protocol |
| IPsec | Internet Protocol Security |
| IPv4 | Internet Protocol version 4 |
| IPv6 | Internet Protocol version 6 |
| LDP | Label Distribution Protocol |
| MPLS | Multiprotocol Label Switching |
| NIC | Network Interface Cards |
| OS | Operating System |
| OSI | Open Systems Interconnection |
| OSPF | Open Shortest Path First |

| | |
|---|---|
| PE | Provider Edge |
| QinQ | IEEE 802.1ad standard |
| RAM | Random Access Memory |
| VM | Virtual Machine |
| VPN | Virtual Private Network |
| WAN | Wide Area Network |

# 1 INTRODUCTION

The GNS3 is a network emulation software with built-in Dynamips hardware emulator, which is able to run Cisco IOS for a limited number of Cisco router models.

Because the actual IOS is emulated, the possible configurations may represent large area of user's needs. It is possible to emulate routers and WAN Interface Cards. Network inside the GNS3 can be connected to any network outside via network adapters on the GNS3 host machine. This also means that the GNS3 can be connected to other virtual networks, and other emulated hardware may be used in the GNS3 topology.

This software piece gives the user an ability to experiment with the network without needing real hardware, which may be cheaper, faster and more convenient. In this research most common usage cases for students will be evaluated in virtual laboratories and based on this, the conclusion about how GNS3, as well as other emulators with the same functionality, can be used to learn new topics, prepare to exams and to understand learning material better by practicing.

In this study, capabilities of GNS3 will be tested by running virtual network topologies and scenarios, similar to the ones that XAMK university students face during their education process. An analysis of GNS3 performance and capabilities to successfully assist the potential user during those specific tasks, will provide enough data for a complex conclusion about the usability of that application, either alone or in combination with other software and technics.

# 2 INSTALLATION

GNS3 executable installer is available from official website, after registration. Supported operating systems are MS Windows, Linux and Mac OS. The Installer is a bundle of GNS3, Dynamips and other programs for monitoring and testing,

which may vary for different OS families. Installation of these programs is automatic (on Windows OS) and only require the user to perform basic input. For this thesis Windows version was used, as it was already installed with the other software running on the computers which were used for testing.

## 2.1 VM installation for version 2.X

To install GNS3 2.X VM, a suitable environment for running virtual machines is needed. GNS3 developer provides VM for VirtualBox, VMware ESXi and Workstation. For this thesis, VMware ESXi version 6.5.0 build 4887370 was used. GNS3 VM package for ESXi is supplied in .ova-format. In ESXi, the latest version 6.5.0 has a problem with opening *.ova files to an ESXi server, so an update of server's GUI is needed. To update server, esxui-signed-5214684.vib update package was downloaded from VMware website to the /tmp directory on the server, then the update was performed via following console command:

```
esxcli software vib install -v /tmp/esxui-signed-5214684.vib
```

After reboot, update proved to solve the problem. Another solution was to convert *.ova file to *.OVF file.
After uploading, virtual machine's settings that are adequately represent needed hardware usage needs to be modified. GNS3 VM emulates network hardware, and to avoid double emulation, CPU assisted virtualization should be used whenever possible. This function is supported by most of the modern processors and may be turned on from the virtual machine's settings. Figure 1 shows virtual machine configuration windows in ESXi with hardware virtualization checkbox turned on.

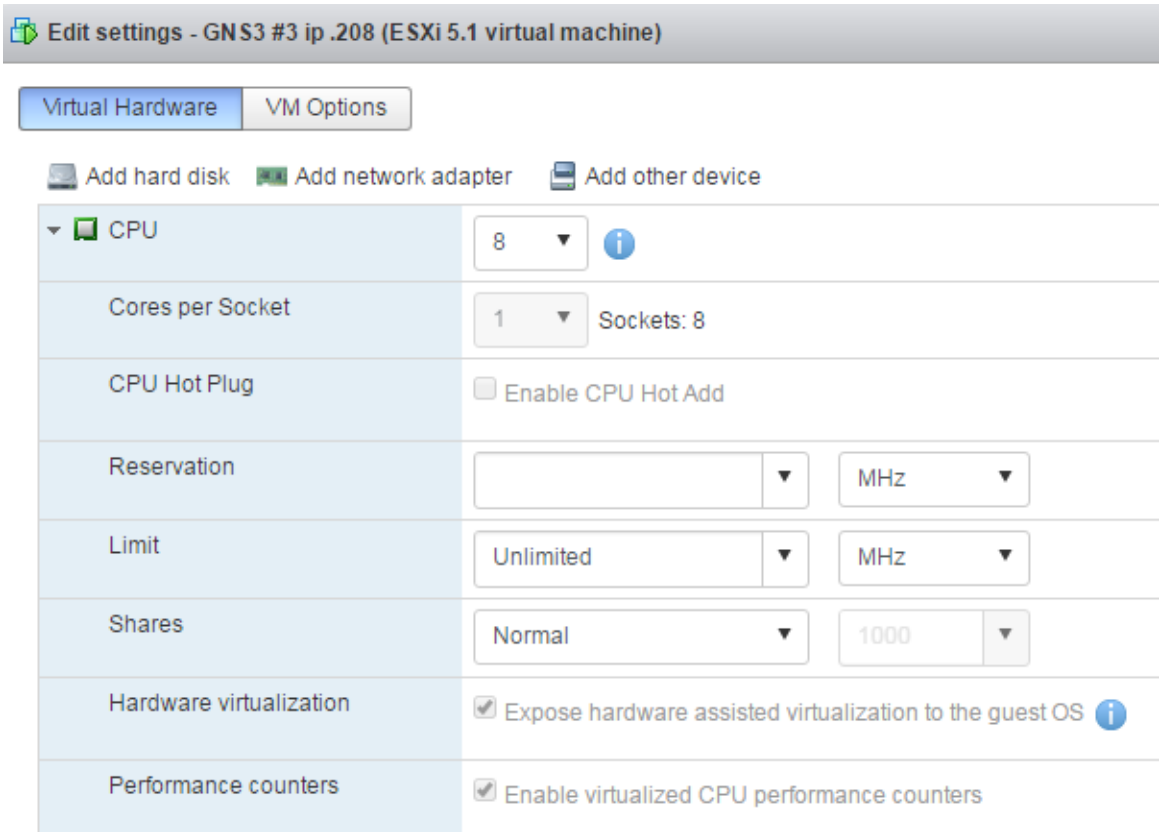



Figure 1. Demonstration of virtual machine's settings.

## 2.2 Hardware used

Tests were performed on different hardware configurations; only most important components are listed.

PC1 Intel core2quad Q6600 (Kentsfield) @ 3200 MHz 4 cores, 6 GB of RAM, 120GB SSD sata1 drive for OS and all GNS3 files. Ram was not sufficient for some extreme cases, and launching 8 routers simultaneously caused noticeable lagging during the router's boot up process for few minutes. During this normal operation of GNS3 was not possible. This configuration is able to sustain enough routers for most configurations, but for more comfortable user experience a better configuration is recommended.

As every router is a different thread, having a CPU with more cores will increase the performance of multi-router configuration.

PC2 Inter Core i5 (Haswell) @ 4500 MHz 4 cores, 16 GB of RAM, 250GB sata2 drive for OS and all GNS3 files. This configuration was able to perform emulation without lagging and memory shortages in any situations, even when other software was used simultaneously. Simultaneous launching of 8 routers causes 100% CPU load for few seconds, without any noticeable lag. Router's boot up process is significantly faster than on the PC1.The PC2 provided more comfort than PC1. Still, GNS3 VM used a lot of RAM, making the system less responsive, if another application with the need for the significant amount of free memory was launched simultaneously.

With the availability of new hardware and new software, running GNS3 on entirely new configuration was possible. GNS3 version 2.0 may split its functionality between two different software pieces: server and client. Server is a virtual machine running Ubuntu Linux and necessary GNS3 overhead. Client connects to the server via network and control server's operation. All performance intensive tasks are run within the server, while the GNS3 client is only used to control server and this task is not hardware intensive.

PC3 is FUJITSU PRIMERGY RX300 S5 is a server machine. It is the last machine used for this research. It was used to run GNS3 server virtual machine, under VMware ESXi Version: 6.5.0, while controls and operation of the GNS3 environment were performed by the GNS3 client on another machine. In this case, performance was sufficient for all tasks in this thesis.

### 2.3 Software licenses

Cisco IOS licenses are available upon buying a Cisco IOS router and can be obtained from the official website or via Cisco license manager application. There are multiple types of routers with different features. It is possible to find features you need with the Cisco feature navigator. (Auda 2013)

# 3 GNS3 STRUCTURE

## 3.1 Version 1.5.x

GNS3 is a bundle of different software, in which main executable launches GNS3, which is user-interface, Dynamips and other emulators (if needed) to launch router's OS, with functionality, including virtualizing router ports, RAM, and disks. Currently, multiple Cisco routers, Cisco ASA (with limitations), and multiple Juniper OS devices are supported. Neither official support nor warranties are provided. Other virtual or physical network appliances, including switches and routers, can be connected to the networks created inside GNS3.

GNS3 interconnects virtual routers running in Dynamips emulator and launches console window connected to the router upon double-click in the GNS3 main window.

As a client for console port, PuTTY or SuperPuTTY is used by default. SuperPuTTY is currently in beta, it is a C# written GUI, which allows multiple console windows to be opened in tabs. Console closes when the router is shut down.

## 3.2 Supported hardware

Supported hardware list is located at the GNS3 official community documentation. The list currently includes many of Cisco's routers and other routers. Cisco's switches are not supported. If the user needs to prepare for Cisco's CCNA, CCNP, and CCIE exams, it is advised to use physical switches or other methods of emulation. (GNS3 official documentation 2017) USB network cards can be used, as a non-expensive way of connecting multiple virtual routers to physical switch or switches. GNS3 networks can expand to any real or virtual adapters, which allows to either connect directly to a computer running GNS3 (on windows, via Microsoft loopback adapter driver), virtual machines of any kind running on that PC, any other physical network device, or another PC.

For working with thesis, two routers will be used: Cisco c7200, as a powerful router with many features and ports, and Cisco c2600, as a router with lesser features, which requires less RAM and CPU resources usage.

In this thesis, Cisco 7200 routers are using firmware image c7200-adventerprisek9-mz.152-4.S1.image or image with equal features and PA-2FE-TX modules, which allow adding fast Ethernet interfaces to routers.

To reduce processor and ram usage Cisco C2961 routers with c2691-advipsservicesk9-mz.124-15.T14.image firmware image will be used for endpoint devices, which basic configuration is not requiring features supported by c7200 routers. Those virtual routers use less amount of RAM than c7200, therefore their performance impact, at least on memory, should be lower.

### 3.3 Version 2.x

With version 2.0, compared to versions 1.5.X came possibility to separate user interface and part of the program which emulates routers. As a result, the program may split into two pieces, GNS3 client, which is a primary application with user interface, and virtual machine, which runs router emulation, and handles projects. The connection between the GNS3 client and GNS3 VM is done via network. This approach allows transferring hardware intensive router emulation to a cloud with extensive resources while connecting to VM and using the full functionality of software from even slowest hardware. Other benefits are arising with this approach:

- Multiple clients could control GNS3 at the same time.
- Third parties can make applications controlling GNS3
- Emulation is still possible locally, within VMware virtual machine, which can be downloaded for free.
- VM able to run IOSv and is usually located in places where placing another virtual machine with needed emulated hardware is possible and convenient.
- Projects are automatically saved as the user makes changes to them.

- Multiple users can connect to the same server and edit the same project in real time.
- “Save as you go”- feature, which allows instantly save changes made to projects and snapshots.

Multiple insignificant features were added, including better support for colorblind users, editable node configuration files, usability improvements. (GNS3 official documentation 2017)

Features may be added or deprecated, as software gets updated at a rate of one update per every two weeks. Beta versions come out even more often. 1.X projects cannot be opened in 2.0. Next projects in this thesis will be launched in 2.0 to maximize research done in this thesis. Differences in new versions will not affect work on this thesis.

### 3.4 Saving projects

GNS3 is able to save user created content, as “projects”. Projects include: topology and items location on topology map, connections, clouds, notes, drawings and pictures, which are added to topology, contents of NVRAM. It is not possible to save running routers, like is in some virtualization software, for example, VMware. All devices must be shut down before saving. It was tested, that upon suspending/hibernating Windows host with running GNS3, its functionality is not interrupted. That includes running virtual routers, which were fully functional, after restoring OS.

## 4 USAGE

### 4.1 Client’s main window

In the main window (Figure 2) there is currently opened project’s network topology, as well as multiple controls to modify it. On the left, there is a sidebar, which allows adding devices to topology via drag-and-drop. On the top, there are tools to draw marks on topology when needed. In the center, there is topology to work with. On the right, there are a list of devices in current project with their status

and list of GNS3 servers. On the bottom, there is a console with notifications about the current situation.

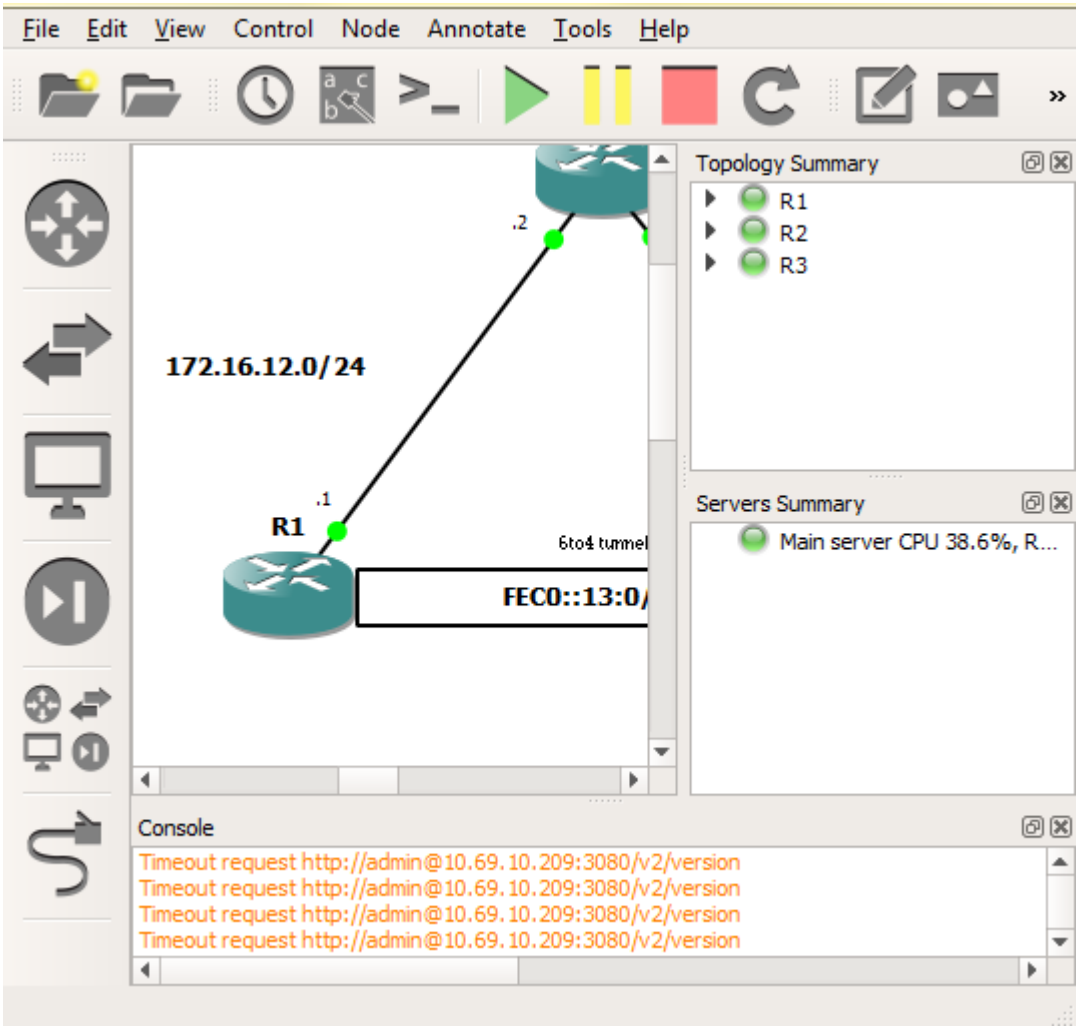



Figure 2. GNS3 user interface

## 4.2 Saving

With version 2.0 any project is saved automatically as changes to shut down devices and topology are added. There is an option to keep the project open when the client disconnects or switches to another project. Snapshots of the project will save its current state. However, snapshot creation and saving are only possible, when all emulated devices in topology are turned off. Closing project without saving will cause all devices to shut down. Saving configurations on IOS routers with

“copy running-config startup-config” command will write emulated NVRAM to disk right away and it will be kept (GNS3 official documentation 2017).

### 4.3 Issues, Duplex mismatch

Duplex mismatch messages may occur on virtual devices in some cases. This issue can be ignored inside the virtual lab; the connection will still work. Command to omit logging: “(config)#no cdp log mismatch duplex”. Another solution: set both interfaces to the same duplex manually or change router’s NIC cards. If workarounds do not work, disabling CDP completely with “(config)#no cdp run” will remove this message.

## 5 BASIC MPLS LAB

As an example of GNS3 possibilities, here is practical Cisco-devices based lab, in which would require a total of 8 physical routers (at least 4 routers and 4 clients of any kind, but that would limit its functionality. It is based on laboratory made by Jeremy Stretch (2011). Additional configuring for testing routing protocols via L3 MPLS, as well as instabilities, leading to wasteful time consumption could occur). Such amount of Cisco routers would require a considerable amount of money to acquire, as well as space, time and electricity, to exist. With GNS3, all these routers and connections between them can be virtualized on a single PC.

Practical usage case: A networking student may practice in building MPLS networks at home.

### 5.1 Building topology

Topology uses 8 routers, 4 Cisco 7200 representing service provider as Pedge1, Pedge2, P1, P2, and 4 Cisco 2691 routers, representing customer edge: CE1A, CE1B, CE2A, CE2B. Topology building process is pictured in Figure 3: router icons represent routers and black lines represent connections or virtual cables between them. Name of topology at the top was added by the creator. Text near cables connecting to routers, represent port numbers to which virtual cables are

plugged-in. This indication can be turned on and off when needed with “show port numbers” button at the top interface bar. In figure 3 this button is pressed in and identification is shown.

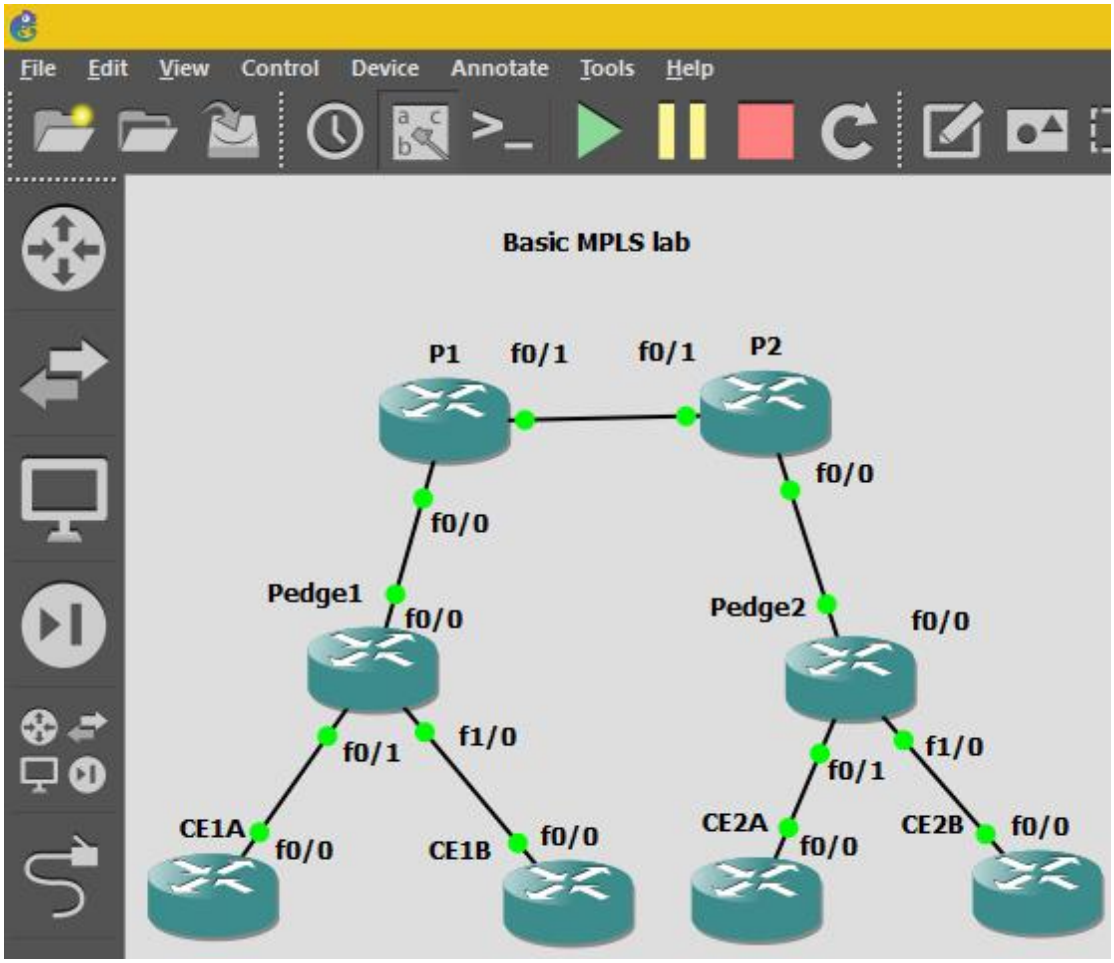



Figure 3. MPLS topology built in GNS3 main window

With GNS3 drawing tools, it is possible to make topologies more human-readable. Additional texts, drawings, pictures, icons can be added to any point in the topology window. Identification of this type may significantly increase the convenience of working with topologies, as well as increase understanding when saved GNS3 projects are shared with others. For example: a teacher may create a topology for students to exercise and write all the tasks right into suitable space in project topology. Students then can return topology with exercise tasks done and additional identification information, if needed. Figure 4 shows capability of

drawing tools to describe routing processes running on specific routers.

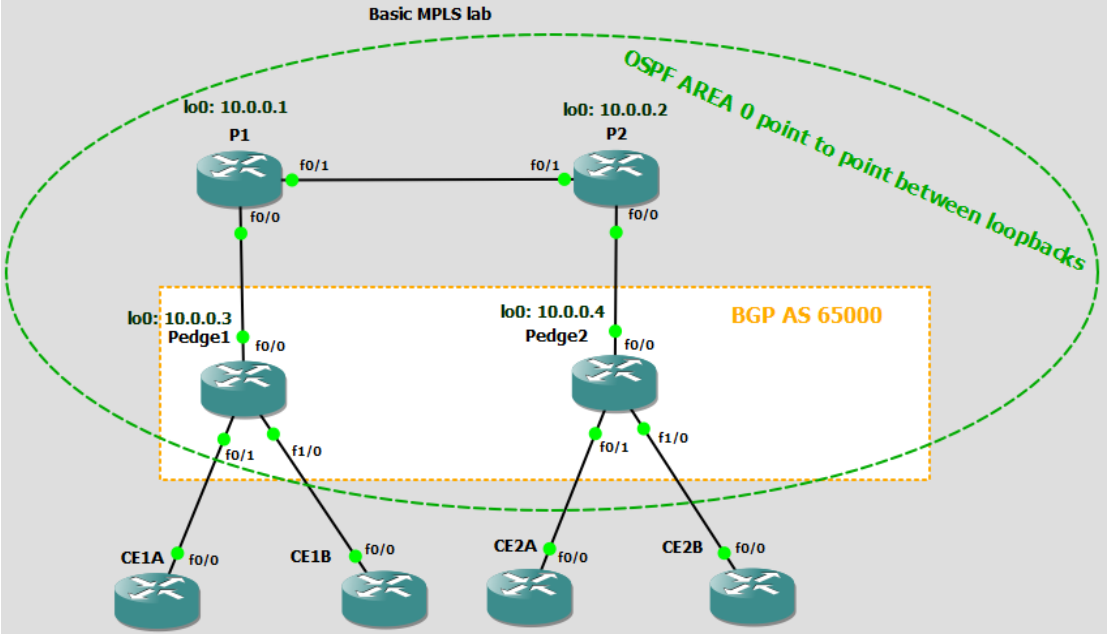



Figure 4. Using GNS3 drawing tools to clarify topologies

Figure 5 shows all IP addresses and connected interfaces, used in this case study.

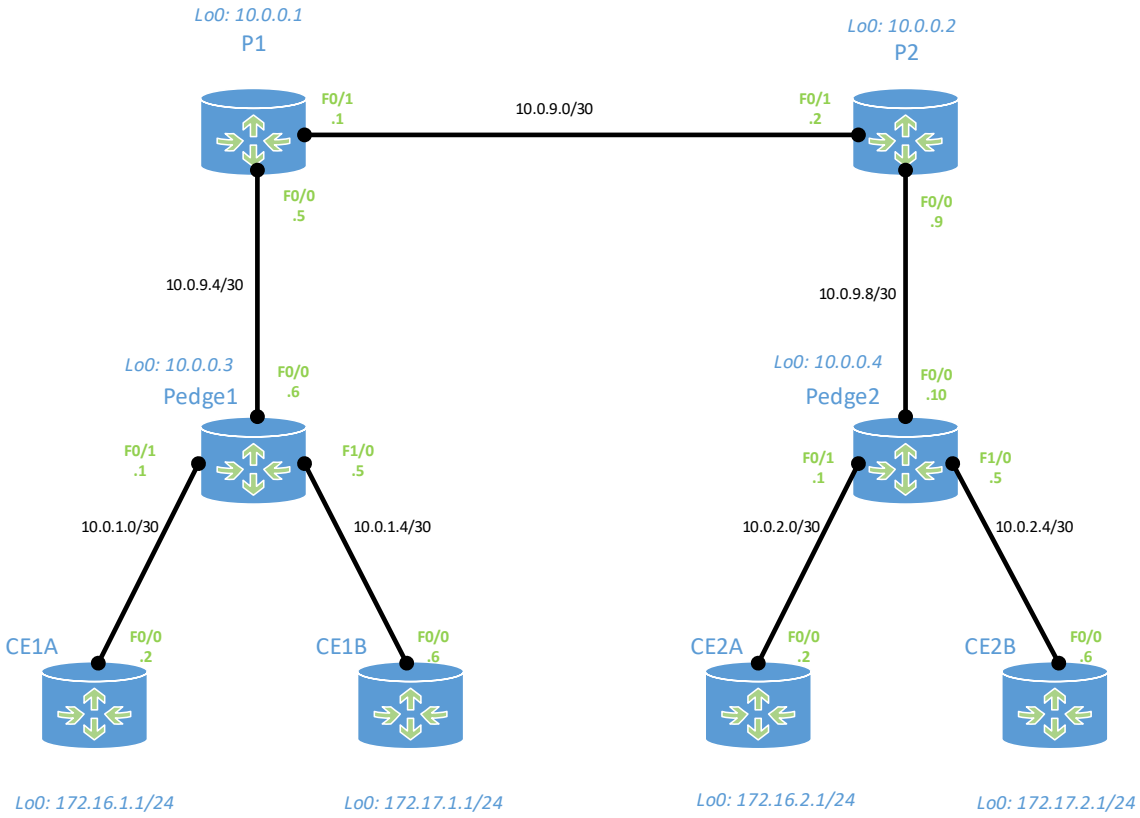


Figure 5. Topology map with all addresses and ports.

Next step is router's configuration. Configuration is performed in the same way as with physical routers. PuTTY may be used as console client and is opened automatically upon double-clicking on the router. There is an option to set custom console client to be used within GNS3. One of the most convenient options is SuperPutty, which can use tabs for easier navigation. This preference is subjective and may vary from user to user.

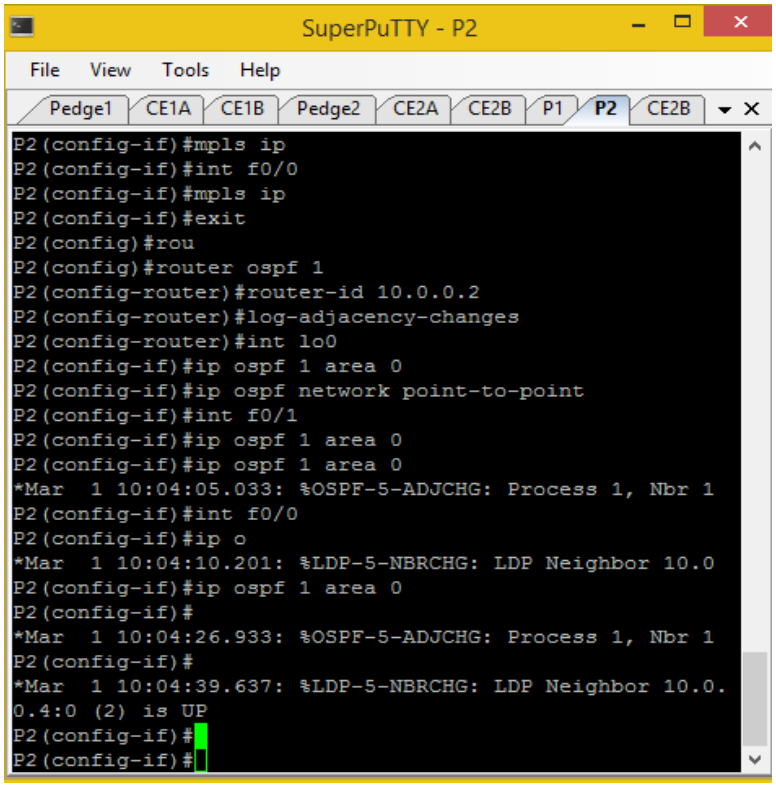



Figure 6. Process of configuring virtual routers via console window using SuperPuTTY client with tabs

In Figure 6 router P2 (Pedge2) is configured with OSPF protocol. Other virtual routers are already configured. OSPF messages show, that OSPF neighbors are up and running. In tabs, there are console windows for all the routers, all of them running and available.

## 5.2 Device roles

Roles of devices in this topology are described in table 1.

| | |
|---|---|
| P1, P2 | Provider core routers. Routers are running OSPF. On all connected interfaces issued “mpls ip” command to participate in label switching. |
| Pedge1, Pedge2 | Provider’s border routers. They are running iBGP sourced from lo0 interfaces to each other. They are also both running OSPF process 1 in area 0, which is active within f0/0 interfaces and lo0 interfaces. There are 2 VRF’s for customer devices ending with “A” and “B” – VRF “Customer_A” and VRF “Customer_B” accordingly. There are 3 OSPF instances on each router: 1 for interconnecting within provider’s own network, 2 for VRF “Customer_A” on interface f0/1 and 3 for VRF “Customer_B” on interface f1/0. Interfaces f0/0 has “mpls ip” command issued. Routers are in vpnv4 peering. |
| CE1A, CE2A | Customer edge devices, running OSPF, in neighborhood relations with provider edge router’s VRF “Customer_A” running OSPF process 2. They are within same MPLS VPN. They have loopbacks0 with IP addresses 172.16.1.1/24 for CE1A and 172.16.2.1/24 for CE2A |
| CE1B, CE2B | Customer edge devices, belonging to another customer, within same MPLS VPN. On Pedge routers, there is |

| | |
|---|---|
| | OSPF process 3 running within VRF “Customer_B”, but on the customer side, OSPF is not configured, these routers will be used for testing later on. They have loopbacks0 with IP addresses 172.17.1.1/24 for CE1A and 172.17.2.1/24 for CE2A |

Table 1. Description of device roles in topology. On the left is device name, on the right – its role

Device roles describe, what function specific device will serve in final topology.

### 5.3 Notes

The MPLS protocol is LDP, default for cisco routers used.
MPLS is layer 3, with OSPF running within VRF’s on provider edge routers.
Method of routing MPLS packets inside provider network is multi-protocol BGP, having address families for each VRF. Here is a configuration on the Pedge1:

```
router bgp 65000
 no synchronization
 bgp log-neighbor-changes
 neighbor 10.0.0.4 remote-as 65000
 neighbor 10.0.0.4 update-source Loopback0
 no auto-summary

 address-family vpnv4
  neighbor 10.0.0.4 activate
  neighbor 10.0.0.4 send-community extended
 exit-address-family

 address-family ipv4 vrf Customer_B
  redistribute ospf 3 vrf Customer_B
  no synchronization
```

```
 exit-address-family

 address-family ipv4 vrf Customer_A
  redistribute ospf 2 vrf Customer_A
  no synchronization
 exit-address-family
```

To reduce the volume of text, examples of configuration will be given only for one router from a group of routers, if the configuration of another router is very similar, and only router specific entries (example: IP addresses) are different and may be understood with the use of topology map.

### 5.4 Configuration

Configuration is performed via console, in the same way, as with the physical routers. Full configuration of Pedge1 and Pedge2 routers is available in this document. (Appendix 1)

#### 5.4.1 Customer edge routers

In this example of CE1A configuration, ports and OSPF routing are configured with the following commands:

```
interface Loopback0
 ip address 172.16.1.1 255.255.255.0

interface FastEthernet0/0
 ip address 10.0.1.2 255.255.255.252

router ospf 1
 log-adjacency-changes
 network 10.0.1.0 0.0.0.3 area 0
 network 172.16.1.0 0.0.0.255 area 0
```

All other customer edge routers are configured in a similar way, with only IP addresses changing accordingly.

### 5.4.2 Provider edge routers.

Provider routers facing customers are configured with VRF, with route distinguisher and route target used for distinguishing and sharing customer's routes in the network. Configuration example from Pedge1 router:

```
ip vrf Customer_A
 rd 65000:1
 route-target export 65000:1
 route-target import 65000:1

ip vrf Customer_B
 rd 65000:2
 route-target export 65000:2
 route-target import 65000:2
```

Configuration of interface with VRF forwarding, each interface is configured in the same way but with different VRF:

```
interface FastEthernet0/1
 description to CE1A VRF Customer_A
 ip vrf forwarding Customer_A
 ip address 10.0.1.1 255.255.255.252
 ip ospf 2 area 0
```

Final topology map with boxes of same color fill representing areas within same MPLS VPN is described in Figure 7, corresponding VRFs are also marked:

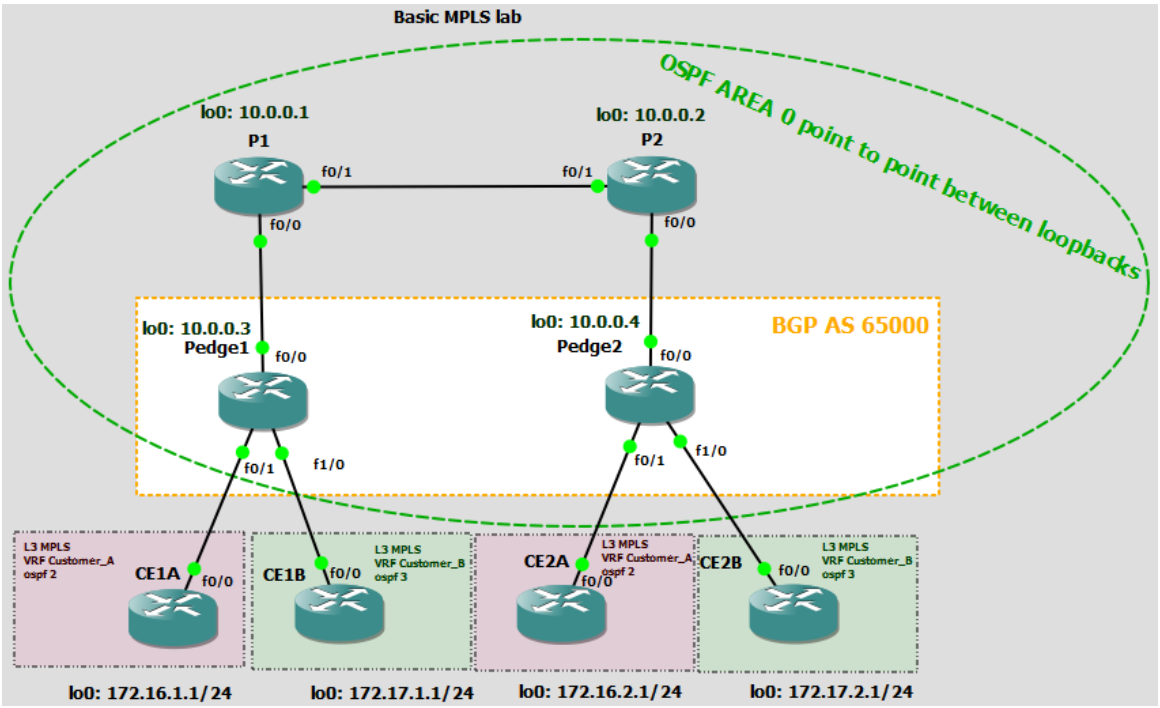



Figure 7. Final topology map with VPNs.

### 5.4.3 Provider's internal routers

P1 and P2 are configured with loopback interfaces, routing and MPLS label switching on all physical connected interfaces:

```
interface Loopback0
 ip address 10.0.0.2 255.255.255.255
 ip ospf network point-to-point
 ip ospf 1 area 0

interface FastEthernet0/0
 ip address 10.0.9.9 255.255.255.252
 ip ospf 1 area 0
mpls ip

interface FastEthernet0/1
 ip address 10.0.9.2 255.255.255.252
 ip ospf 1 area 0
```

```
 duplex auto
 speed auto
 mpls ip

router ospf 1
 router-id 10.0.0.2
```

With configuration complete, topology can be tested.

### 5.5 Testing of topology

In Figure 8 a single ping ICMP packet was sent from CE1A to CE1B located on the different sides of MPLS edges. On CE1B packet debugging is on, incoming and outgoing ICMP packet details are displayed. In this case, MPLS is functioning correctly.

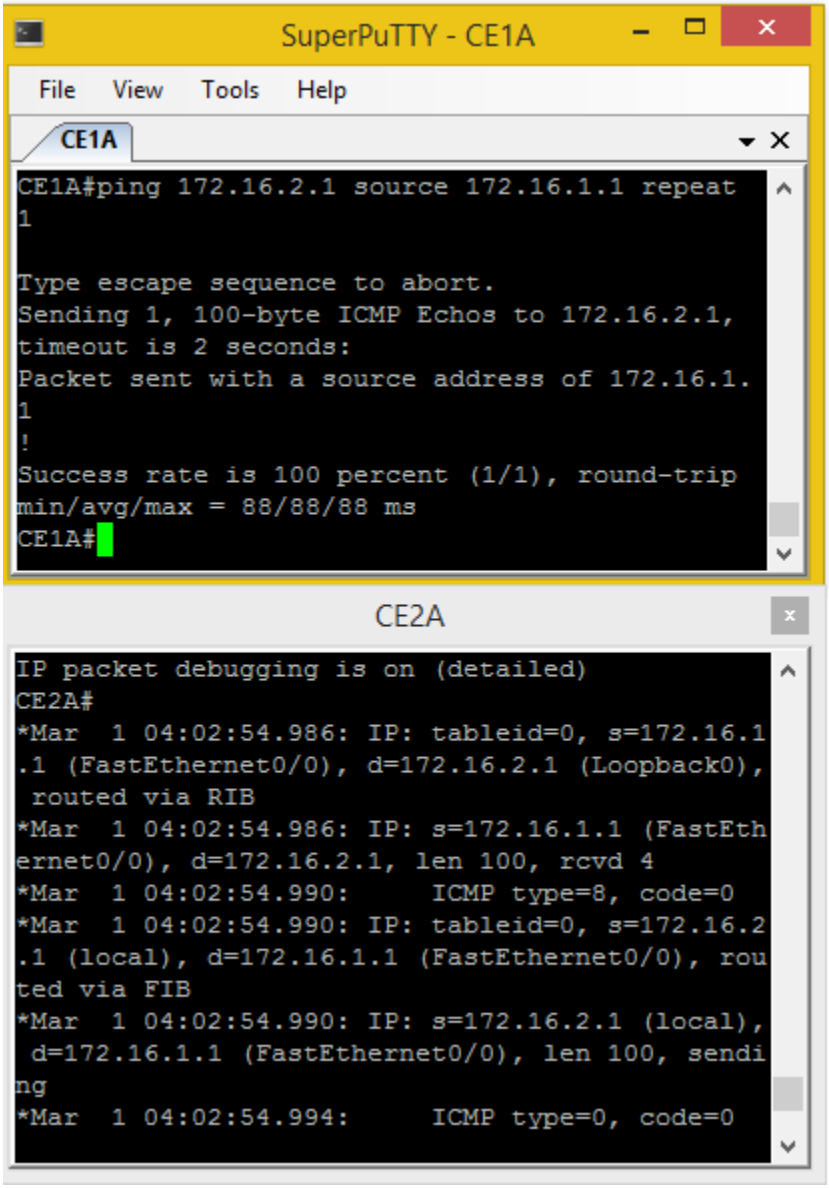



Figure 8. Console output from routers CE1A and CE2A pinging each other.

### 5.6 Capturing packets

Practical usage cases: In the CCNP theoretical exams, packet structure questions may occur. Capturing packets may help students to better understand network protocols.

Security experts may test, what information potential intruder may gain in case of unauthorized access to the network at different points.

Packet capturing is available in every point of GNS3 topology. To test this, let's assume that it is needed to capture ICMP packet from **Error! Reference source not found.**, while it travels in provider's network, encapsulated in LDP. To start capturing at a point inside provider's network (Pedge1 port f0/0 is taken for this example), the user must select the link (Figure 9), then select port (Figure 10) and Wireshark window will open automatically, with capturing mode started.

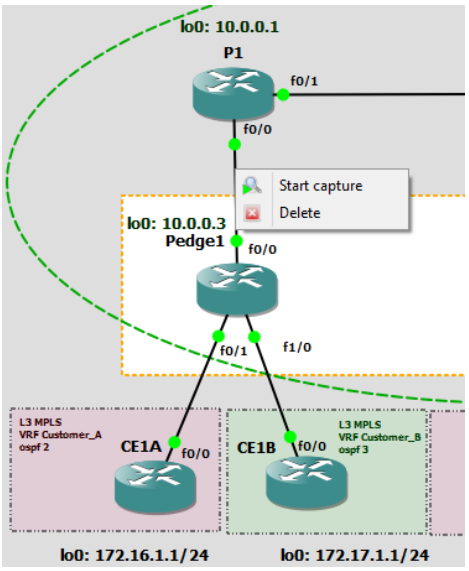



Figure 9. Link selection between Pedge1 and P1

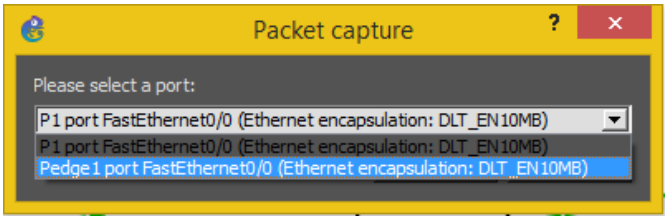



Figure 10. Interface selection for frame capture

For this test, first, single ICMP ping will be performed on CE1A, with the following command: "CE1A#ping 172.16.2.1 source 172.16.1.1 repeat 1"
The packet will be captured and analyzed. Then, single ping from Pedge1 to P1 will be performed and the packet will also be analyzed. The command for this action is:
"Pedge1#ping 10.0.0.1 repeat 1"

These actions will create a total of 4 ICMP packets.

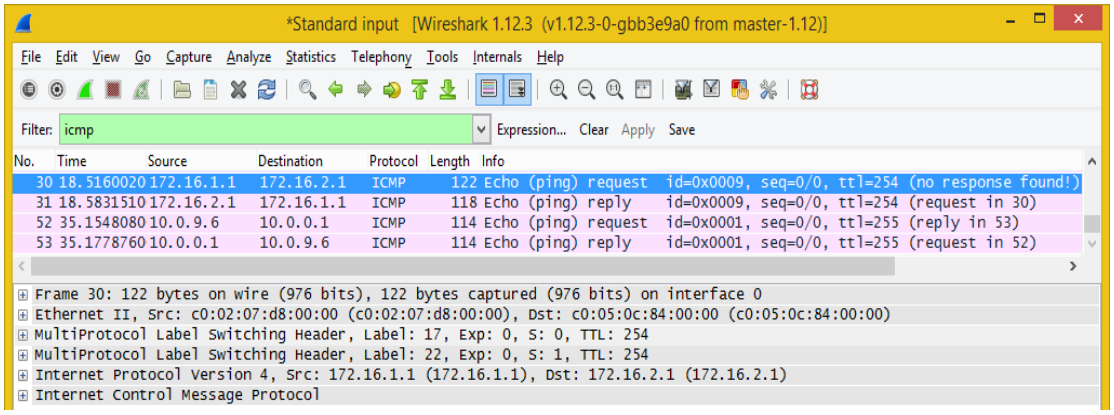



Figure 11. Wireshark window with results of capturing

In Figure 11 Wireshark successfully captured 4 ICMP packets. Upper 2 are LDP encapsulated packets, and lower 2 not, because they are not originating from MPLS VPN. Examination of packets 30 and 31 is in Figure 12:

```
Frame 30: 122 bytes on wire (976 bits), 122 bytes captured (976 bits) on interface 0
Ethernet II, Src: c0:02:07:d8:00:00 (c0:02:07:d8:00:00), Dst: c0:05:0c:84:00:00 (c0:05:0c:84:00:00)
MultiProtocol Label Switching Header, Label: 17, Exp: 0, S: 0, TTL: 254
MultiProtocol Label Switching Header, Label: 22, Exp: 0, S: 1, TTL: 254
Internet Protocol Version 4, Src: 172.16.1.1 (172.16.1.1), Dst: 172.16.2.1 (172.16.2.1)
Internet Control Message Protocol

Frame 31: 118 bytes on wire (944 bits), 118 bytes captured (944 bits) on interface 0
Ethernet II, Src: c0:05:0c:84:00:00 (c0:05:0c:84:00:00), Dst: c0:02:07:d8:00:00 (c0:02:07:d8:00:00)
MultiProtocol Label Switching Header, Label: 22, Exp: 0, S: 1, TTL: 252
Internet Protocol Version 4, Src: 172.16.2.1 (172.16.2.1), Dst: 172.16.1.1 (172.16.1.1)
Internet Control Message Protocol
```

Figure 12. Packets 30 and 31 participating in MPLS VPN

On Pedge1, “mpls forwarding-table” command output is shown on Figure 13:

```
Pedge1#sh mpls forwarding-table
Local  Outgoing    Prefix           Bytes tag  Outgoing    Next Hop
tag    tag or VC   or Tunnel Id     switched   interface
16     Pop tag     10.0.0.1/32      0          Fa0/0       10.0.9.5
17     Pop tag     10.0.9.0/30      0          Fa0/0       10.0.9.5
18     17          10.0.0.4/32      0          Fa0/0       10.0.9.5
19     18          10.0.0.2/32      0          Fa0/0       10.0.9.5
20     19          10.0.9.8/30      0          Fa0/0       10.0.9.5
21     Aggregate   10.0.1.0/30[V]   0
22     Untagged    172.16.1.1/32[V] 26106      Fa0/1       10.0.1.2
23     Aggregate   10.0.1.4/30[V]   0
24     Untagged    172.17.1.1/32[V] 0          Fa1/0       10.0.1.6
Pedge1#
```

Figure 13. MPLS forwarding table on Pedge1

From this output, following conclusion can be made: Frame 30, ICMP ping request packet from CE1A have the tag of 17 and will be sent to 10.0.0.4 which is Pedge2, provider edge router, at another edge of MPLS, where CE2A is located. In frame 31, ICMP reply packet from CE2A router P1 popped its tag (16), because next hop router has tunnel associated with tag 22. When Pedge1 receives LDP packet with tag 22, it will send it to the tunnel with IP 172.16.1.1, untagged, this is where CE1A is.

```
Frame 52: 114 bytes on wire (912 bits), 114 bytes captured (912 bits) on interface 0
Ethernet II, Src: c0:02:07:d8:00:00 (c0:02:07:d8:00:00), Dst: c0:05:0c:84:00:00 (c0:05:0c:84:00:00)
Internet Protocol Version 4, Src: 10.0.9.6 (10.0.9.6), Dst: 10.0.0.1 (10.0.0.1)
Internet Control Message Protocol
```

```
Frame 53: 114 bytes on wire (912 bits), 114 bytes captured (912 bits) on interface 0
Ethernet II, Src: c0:05:0c:84:00:00 (c0:05:0c:84:00:00), Dst: c0:02:07:d8:00:00 (c0:02:07:d8:00:00)
Internet Protocol Version 4, Src: 10.0.0.1 (10.0.0.1), Dst: 10.0.9.6 (10.0.9.6)
Internet Control Message Protocol
```

Figure 14. Pings outside of MPLS VPN

Figure 14 shows, that packet exchange between Pedge1 f0/0 and P1 lo0: are not within MPLS and have no MPLS header.

## 6 SCALABILITY

Because all routers are virtual, a total number of devices is limited by resources of PC it is running on. If one PC's resources are not enough, it is possible to use another instance of GNS3 on another PC and interconnect them via a physical network.

Console connections for controlling routers are also problematic with large physical networks. Console connections would require either console cabling to each router from one PC, one router at a time, or acquiring separate console server device, which have limited number of ports, and requires configuration, or connect multiple devices to one PC with multiple com ports (additional serial controllers may be required), or workarounds, like initially configure routers to work via

SSH (SSH connectivity may fail in case of configuration/network/link issues). Those connection methods are not usually problematic in operational networks, but in network testing situations, where there are multiple device adding, removing and re-configuration, connectivity may be problematic. GNS3 and other network virtualization technics solve this problem.

## 7 CASE STUDY: MPLS INTER-AS NETWORK (OPTION B)

MPLS inter-AS option B is one of CCIE topics. If customer's need in MPLS VPN may exceed one internet service provider's capabilities, then it can be expanded to another provider. However, each provider usually has its own autonomous system with label switching procedure for various reasons.

Solution to this is inter-AS MPLS VPN, which exists in various options and their combinations. Most common options are A, B, C and D.

Option A back to back VRF is a solution with opening VPN at provider border edge. In this situation, both providers' border edges are connected with additional VRF interface or sub-interface within VPN that needs to be expanded over two providers. In this situation, provider border edges see each other in the same way they are seeing customer edge routers and are configured in a similar way as provider edge routers facing VPN customers. This means that traffic between back to back VFR interfaces is plain IP. This solution requires configuration of new VRF and a new interface for each VPN and may cause problems with scalability.

In inter-AS Option B solution, a vpnv4 session is configured between providers' border routers, through which vpnv4 routes and labeled VPN prefixes are transmitted from one AS to another. This means that MPLS traffic between ASBRs is labeled. Unlike option A this solution requires less additional configuration when new VPN's are being added to the configuration. The disadvantage of this method is that MPLS packets are not affected by Quality of Service features of

routers, if such feature implementation is needed. Other options will not be covered in this thesis.

## 7.1 Note on devices reserved for future use

During case study labs, there will be devices with no configuration added into the topology. After the end of this research, topologies will be posted to GNS3 official forums for everyone to practice. The intent is to help others. The author of this thesis was not able to find a single MPLS inter-AS B topology fully build in Cisco IOS routers, but had seen request on discussion boards.

## 7.2 Topology

Topology map with all ports and addresses pictured in Figure 14:

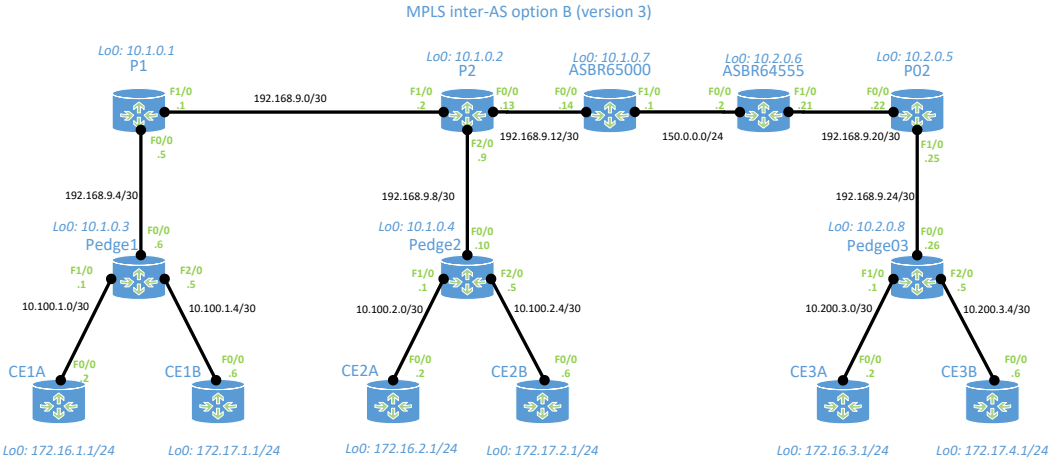



Figure 14. Topology map

GNS3 has enough drawing tools to completely describe topology, as shown in Figure 15:

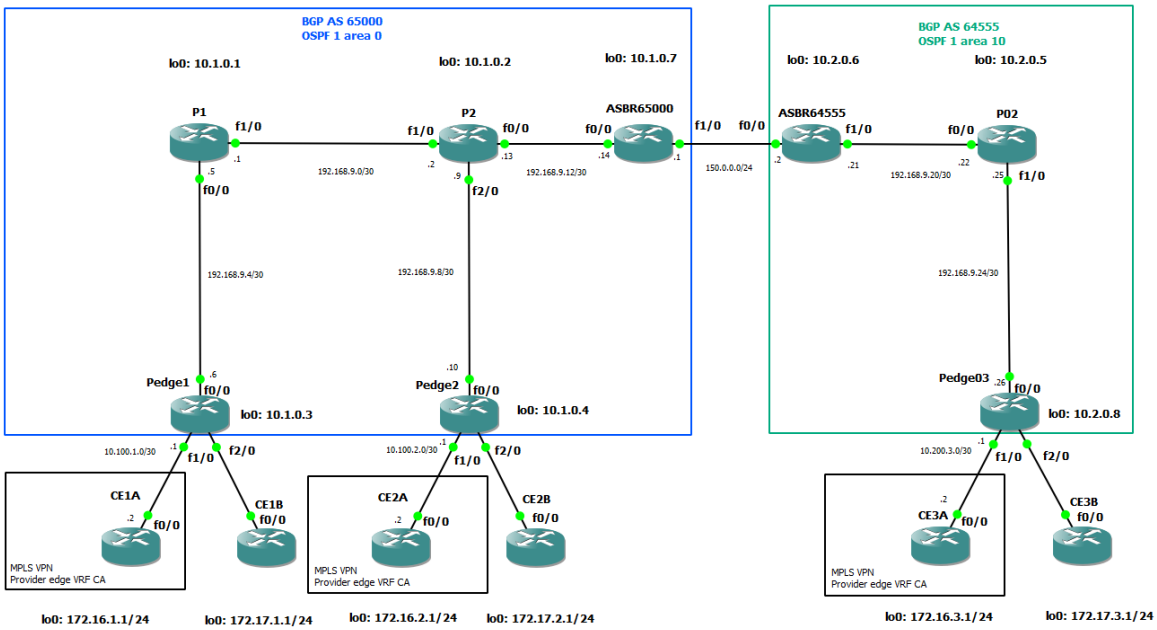



Figure 15. Topology map with marks, as shown in GNS3

With this information, it will be easier for the user to navigate topology.

## 7.3 Device roles

Devices in this topology will have following roles described in in Table 2:

| | |
|---|---|
| ASBR65000 | AS border router for AS 65000, named ASBR65000, which will be connected to P2 router, and will run OSPF, iBGP and eBGP will have a vpnv4 session with ASBR64555. |
| ASBR64555 | AS border router for AS 64555, named ASBR64555, which will be connected to P01 and ASBR65000 routers, and will run iBGP and eBGP. It will have a vpnv4 session with ASBR65000. |

| | |
|---|---|
| P1, P2, P02 | Those providers' internal routers will interconnect others, and perform label switching. They do not run BGP. |
| CE1A, CE2A, CE3A | Group A of customer edge routers, they will have OSPF neighborhood with corresponding routing process on connected provider edge router and will share routes which will be isolated by MPLS VPN. |
| CE1B, CE2B, CE3B | Reserved for future use. |
| Pedge1, Pedge2, Pedge03 | Provider edge routers, MPLS VPN endpoints to customer edge routers, connected to provider network and customer's routers. Pedge1 and Pedge2 are located in AS 65000, while Pedge03 is located in AS 64555. |

Table 2. Device roles in this case study. On the left is device name, on the right – its role

## 7.4 Configuration

### 7.4.1 Customer edge routers

For maximum simplicity, this topology has all customer edge routers have a similar configuration, with one interface towards corresponding provider edge router, OSPF routing and loopback interface. Configuration is based on Cisco's official documentation. (Verma 2016) An example configuration of CE1A:

```
interface Loopback0
 ip address 172.16.1.1 255.255.255.0
 ip ospf 10 area 11

interface FastEthernet0/0
 description to Pedge1
```

```
 ip address 10.100.1.2 255.255.255.252
 ip ospf 10 area 11

router ospf 10
 router-id 172.16.1.1
```

### 7.4.2 Providers' edge routers

Provider edge routers have VRF's corresponding to attached customer VPN, which contains route distinguisher and route target to import and export routes specific to this VPN. VRF named "CA" is chosen for customer edge "A" group routers. Configuration will be examined as is on Pedge1 router:

```
ip vrf CA
 rd 65000:1
 route-target export 65000:1
 route-target import 65000:1
```

An interface facing customer with corresponding VRF forwarding and OSPF routing:

```
interface FastEthernet1/0
 description To CE1A VRF Customer A
 ip vrf forwarding CA
 ip address 10.100.1.1 255.255.255.252
 ip ospf 10 area 11
```

All inside interfaces within provider are configured with MPLS forwarding:

```
interface FastEthernet0/0
 description To P1
 ip address 192.168.9.6 255.255.255.252
 ip ospf 1 area 0
```

```
 mpls ip
```

OSPF processes, 1 for inter-provider routing, 10 for VRF instance “CA”, which re-distributes routes from other endpoints of MPLS VPN from BGP to OSPF:

```
router ospf 10 vrf CA
 router-id 10.100.1.1
 redistribute bgp 65000 subnets

router ospf 1
 router-id 10.1.0.3
```

Configuration of BGP process includes multiprotocol iBGP IPv4 peering, activation of VPNv4 peering, setting next hop for VPN endpoints to self, and redistribution of VPN’s routes learned over OSPF process 10 to be available to other endpoints of MPLS VPN:

```
router bgp 65000
 bgp log-neighbor-changes
 no bgp default ipv4-unicast
 neighbor 10.1.0.4 remote-as 65000
 neighbor 10.1.0.4 update-source Loopback0
 neighbor 10.1.0.7 remote-as 65000
 neighbor 10.1.0.7 update-source Loopback0

 address-family ipv4
  neighbor 10.1.0.4 activate
  neighbor 10.1.0.7 activate
 exit-address-family

 address-family vpnv4
  neighbor 10.1.0.4 activate
  neighbor 10.1.0.4 send-community extended
```

```
 neighbor 10.1.0.4 next-hop-self
 neighbor 10.1.0.7 activate
 neighbor 10.1.0.7 send-community extended
 neighbor 10.1.0.7 next-hop-self
exit-address-family

address-family ipv4 vrf CA
 redistribute ospf 10
exit-address-family
```

Configuration of Pedge03 router is different as there is no other provider-to-customer edge routers, VPN and IP peering is only performed with ASBR64555

### 7.4.3 Providers' internal routers.

Internal routers P1, P2 and P3 are only configured with interfaces, OSPF and label switching. Example of P1 configuration:

```
interface Loopback0
 ip address 10.1.0.1 255.255.255.255
 ip ospf network point-to-point
 ip ospf 1 area 0

interface FastEthernet0/0
 description To Pedge1
 ip address 192.168.9.5 255.255.255.252
 ip ospf 1 area 0
mpls ip

interface FastEthernet1/0
 description To p2
 ip address 192.168.9.1 255.255.255.252
 ip ospf 1 area 0
```

```
 mpls ip

router ospf 1
 router-id 10.1.0.1
```

### 7.4.4 Autonomous system border routers

ASBRs are configured with interfaces facing inside and outside AS, on interface facing inside AS label switching and routing is configured, as for example ASBR65000 configuration:

```
interface FastEthernet0/0
 description To P2
 ip address 192.168.9.14 255.255.255.252
 ip ospf 1 area 0
 mpls ip
```

By default, router drops MPLS packets, if it has no VRF configured for that label. ASBRs have no VRFs. To avoid this behavior, and allow labeled packets from another AS to be received, “mpls bgp forwarding” command is issued on an interface facing another ASBR:

```
interface FastEthernet1/0
 description To ASBR64555
 ip address 150.0.0.1 255.255.255.0
mpls bgp forwarding
```

An OSPF process 1 is configured for internal routing:

```
router ospf 1
 router-id 10.1.0.7
```

Multiprotocol BGP is configured with iBGP and eBGP neighborhoods, IPv4 peering and VPNv4 session over which VPN routing information and labels are exchanged to MPLS tunnel endpoints in same AS with border router, as well as with border router of another AS:

```
router bgp 65000
 bgp log-neighbor-changes
 no bgp default ipv4-unicast
 no bgp default route-target filter
 neighbor 10.1.0.3 remote-as 65000
 neighbor 10.1.0.3 update-source Loopback0
 neighbor 10.1.0.4 remote-as 65000
 neighbor 10.1.0.4 update-source Loopback0
 neighbor 150.0.0.2 remote-as 64555

 address-family ipv4
  neighbor 10.1.0.3 activate
  neighbor 10.1.0.4 activate
  neighbor 150.0.0.2 activate
 exit-address-family

 address-family vpnv4
  neighbor 10.1.0.3 activate
  neighbor 10.1.0.3 send-community extended
  neighbor 10.1.0.3 next-hop-self
  neighbor 10.1.0.4 activate
  neighbor 10.1.0.4 send-community extended
  neighbor 10.1.0.4 next-hop-self
  neighbor 150.0.0.2 activate
  neighbor 150.0.0.2 send-community extended
 exit-address-family
```

## 7.5 Verification and testing

All customer edge routers receive routers from all other parts of MPLS VPN network, and are able to ping each other, as shown in Figure 16 and Figure 17:

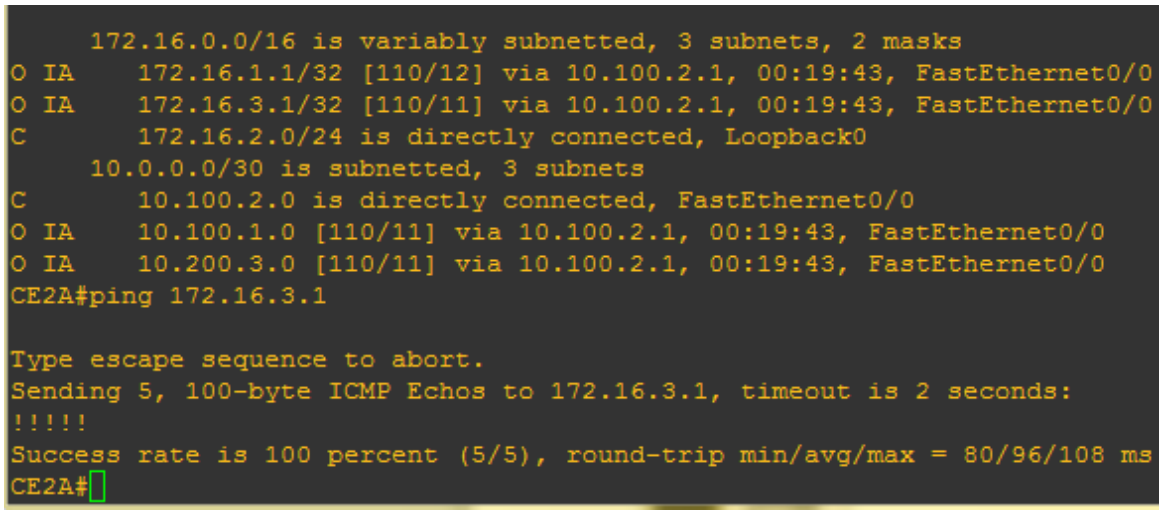



Figure 16. Route map on CE2A and ping issued to a loopback interface of CE3A, with a successful result.

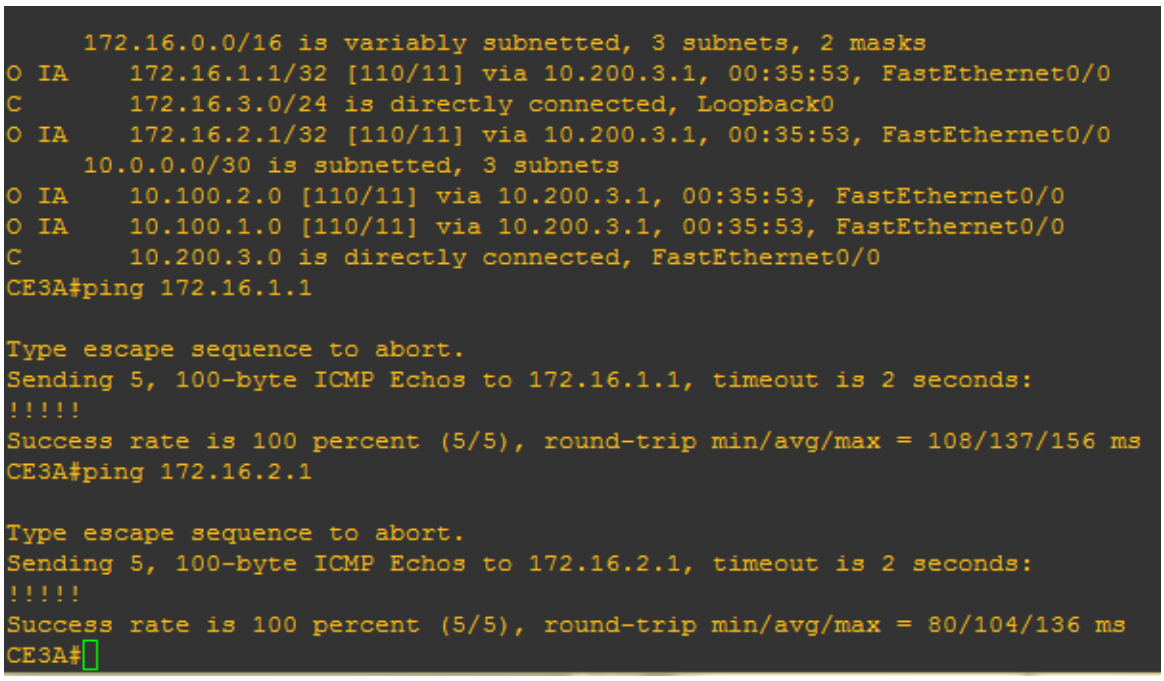



Figure 17. Route map on CE3A and ping issued to CE1A and CE2A loopback interfaces with successful result.

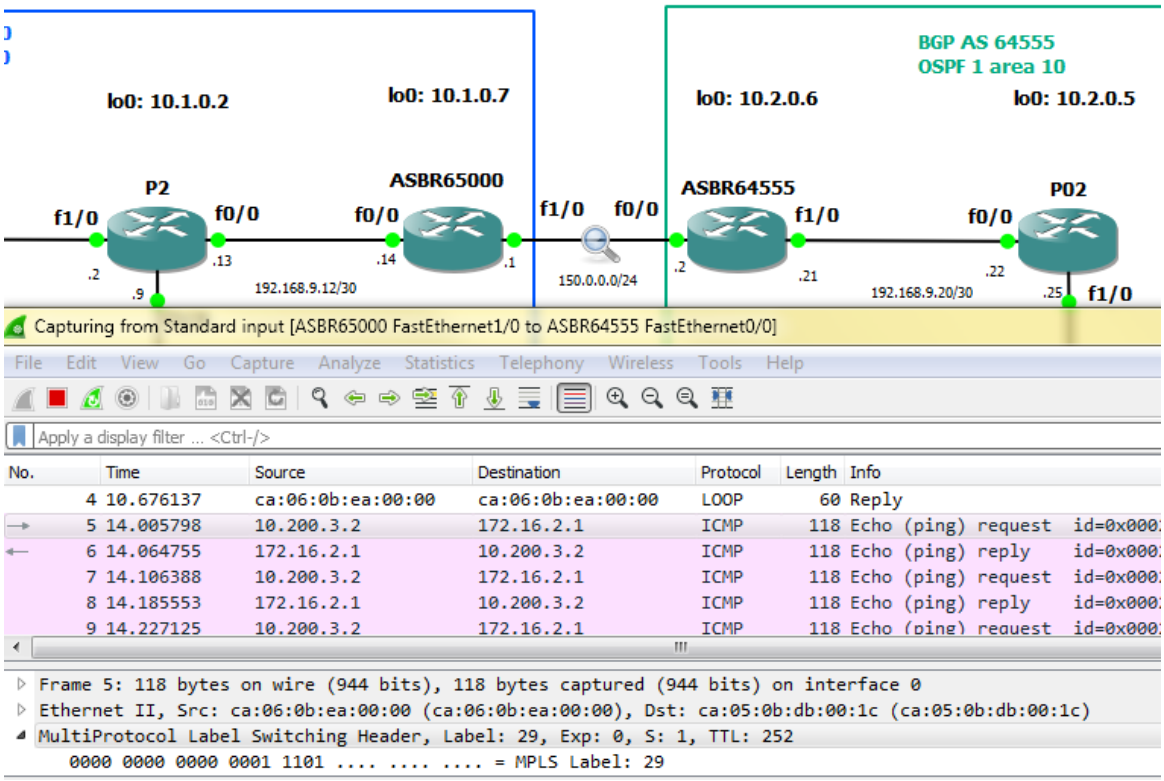



Figure 18. A packet capture between ASBRs

In Figure 18 traffic from one AS to another is labeled. This is the correct operation of Inter-AS MPLS option B.

## 8 CASE STUDY: CCNP LAB IPV6 TUNNELS

Test to determine, if GNS3 is capable of setting environment for CCNP labs, a topology with settings similar to one in CCNP ROUTE 6.0 student guide (2010, 331) will be implemented in GNS3 and its functionality will be tested.

### 8.1 Topology

Figure 19 is topology map with all needed information.

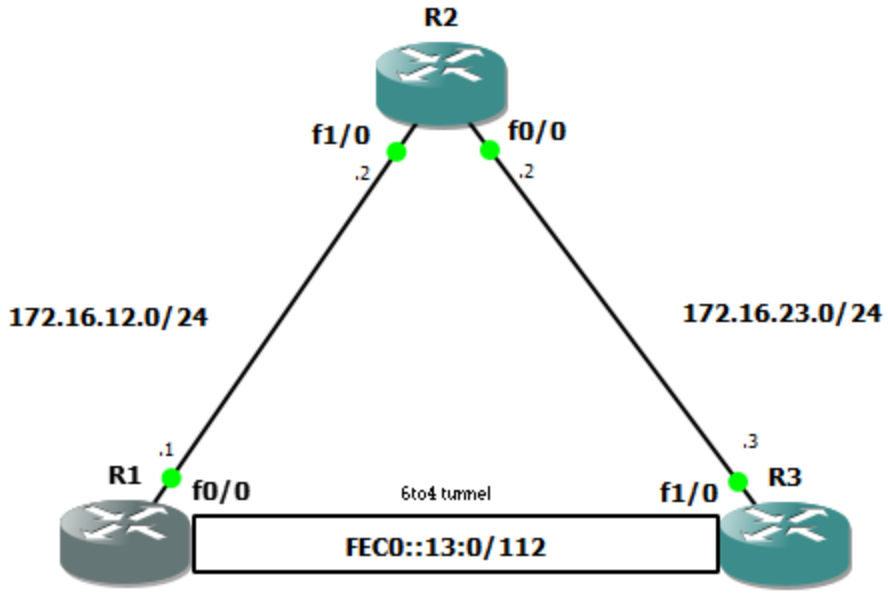



Figure 19. Topology map for this case study

This is topology with 6to4 tunnel. IPv6 traffic cannot navigate through IPv4 networks as-is, however, need of navigating such traffic may occur, as result of gradual switching to newer Internet protocol. To avoid re-configuration on all the devices, a 6 to 4 tunnel may be used.

## 8.2 Device roles

Device roles in this topology are described in Table 3:

| | |
|---|---|
| R1, R3 | IPv6 tunnel endpoints, both connected to R2 with IPv4 protocol only |
| R2 | IPv4 routing |

Table3. Description of device roles in topology. On the left is device name, on the right – its role

### 8.3 Configuration

All router interfaces were configured with IPv4 addresses, except loopback addresses, which have IPv6 addresses for future expansion.
R1 has 3 interfaces, one facing R2 with IPv4 only address, one loopback and one tunnel interface pointing to F1/0 of R3:

```
interface Tunnel0
  ipv6 address FEC0::13:1/112
 tunnel source FastEthernet0/0
 tunnel destination 172.16.23.3
 tunnel mode ipv6ip
```

A loopback interface configured with IPv4 and IPv6:

```
interface Loopback0
 ip address 10.1.1.1 255.255.255.0
 ipv6 address FEC0::1:1/112
```

An EIGRP routing configuration is exactly same on all routers in this topology:

```
router eigrp 1
 network 10.0.0.0
 network 172.16.0.0
 no auto-summary
```

R2 has two IPv4 interfaces configured and EIGRP routing.

R3 has tunnel interface with only IPv6 address pointing to R1 f0/0:

```
interface Tunnel0
 no ip address
 ipv6 address FEC0::13:3/112
 tunnel source FastEthernet1/0
 tunnel destination 172.16.12.1
```

tunnel mode ipv6ip

A loopback interface configured with IPv4 and IPv6:

interface Loopback0

ip address 10.1.3.1 255.255.255.0

ipv6 address FEC0::3:1/112

## 8.4 Verification and additional configuration

To verify connectivity a ping is issued over a tunnel and shown in Figure 20:

```
R1#ping FEC0::13:3

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to FEC0::13:3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 24/27/28 ms
R1#ping FEC0::3:1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to FEC0::3:1, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
R1#
```

Figure 20. Ping command issued on R1 though 6to4 tunnel to other tunnel endpoint successful result and unsuccessful ping of R3 loopback.

Routers have successful IPv6 connectivity over a tunnel. There is no connection between loopbacks, because there are no routing protocols running for IPv6. IPv6 routing can be enabled with:

ipv6 unicast-routing

And EIGRP can be enabled on each interface. Configuration is same on R1 and R3:

interface loopback0

Ipv6 eigrp 100

interface tunnel0

ipv6 eigrp 100

IPv6 EIGRP is shut down by default. To turn it on, “no shutdown” command must be issued under EIGRP with used process number:

```
ipv6 router eigrp 100
no shutdown
```

After routing was enabled, loopbacks have connectivity.

```
R1#ping FEC0::3:1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to FEC0::3:1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 12/18/20 ms
R1#
```

Figure 21. Ping command issued on R1 to routed loopback interface address with successful result

Figure 21 shows, that not available before address is now responding to ping request on R1, as a result of a successful routing.

## 9 CONNECTING GNS3 WITH OTHER DEVICES.

GNS3 VM is able to use connected network adapters to connect emulated network inside GNS3 to any network outside GNS3. To perform connection, additional virtual or physical network adapters are added to VM and they are configured with IP addresses, which then will be used for connected devices inside GNS3. To test this function, a case study with interconnecting devices inside and outside of GNS3 will be implemented in section 9.1

Configuration of Internet adapters is possible within GNS3 VM under “configure networking settings” menu, (Figure 22) this action opens Nano editor with file “/etc/network/interfaces”. Network adapters also need to be started in order to be used, to start them automatically at system boot command “auto <interface name> “issued under each interface. Example: “auto eth3”.

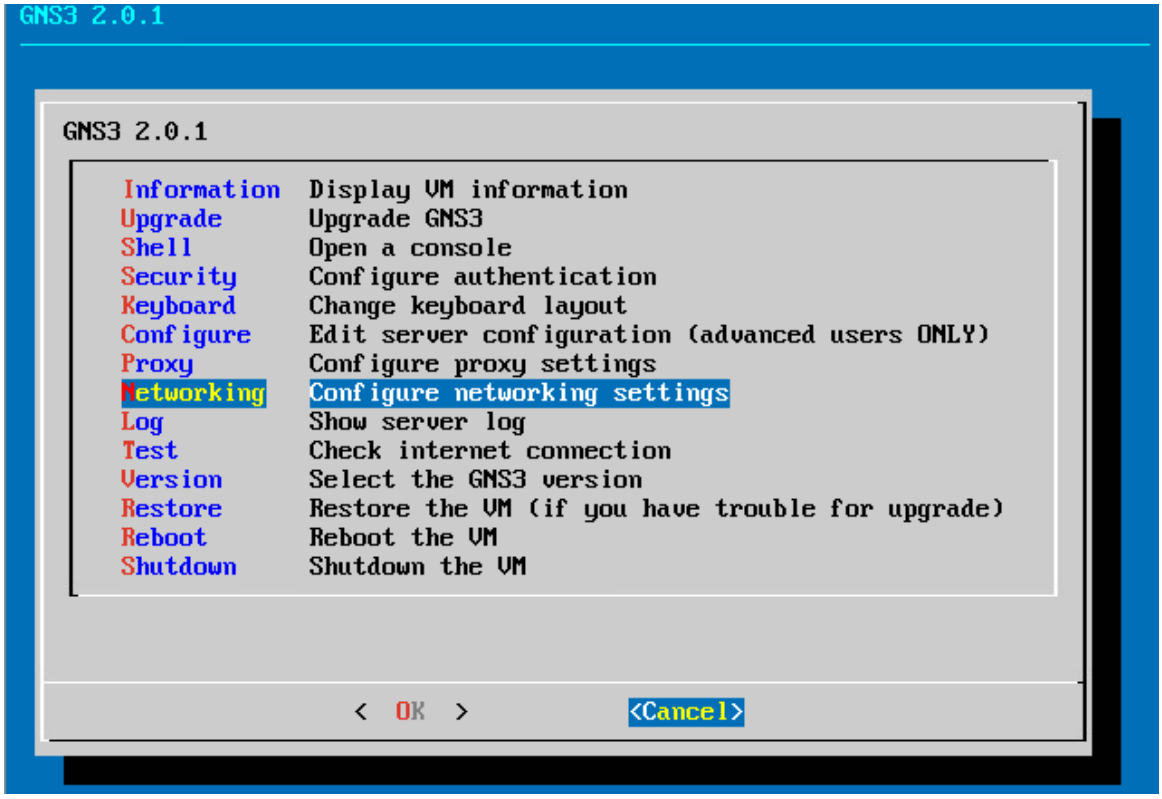



Figure 22. GNS3 VM main menu with option to configure network settings selected

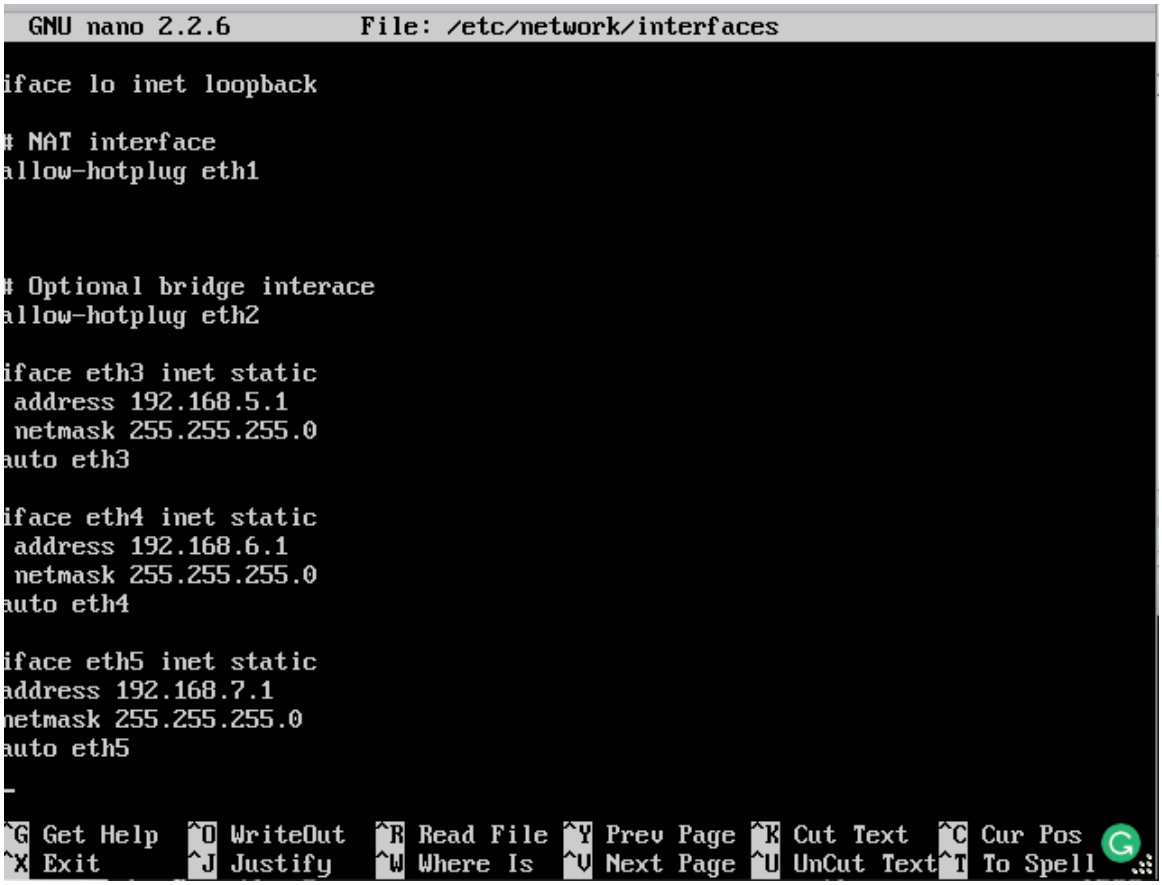



Figure 23. Configuration of network interfaces used to connect to outside word

Each ESXi interface has a separate switch connected only to a needed networking appliance to eliminate any unwanted network traffic, as pictured in Figure 24. It is also possible to use a single interface, and connect multiple devices using NAT. This option requires additional configuration.

| Name | Active.. | VLAN... | Type | vSwitch | VMs |
|---|---|---|---|---|---|
| VM Network | 2 | 0 | Standard port group | vSwitch0 | 5 |
| Management Network | 1 | 0 | Standard port group | vSwitch0 | N/A |
| GNS3 Cloud 1 | 1 | 0 | Standard port group | GNS3 cloud 1 | 2 |
| GNS3 Cloud 2 | 1 | 0 | Standard port group | GNS3 cloud 2 | 2 |
| GNS3 Cloud 3 | 1 | 0 | Standard port group | GNS3 Cloud 3 | 2 |
| GNS3 Cloud 4 | 1 | 0 | Standard port group | GNS3 Cloud 4 | 2 |

6 items

Figure 24. ESXi networking port groups configuration

Port group in ESXi require Promiscuous mode to be set to “Yes” in port group security settings. This is needed for correct GNS3 functionality.

After GNS3 VM with correct port configuration is started, network adapters facing outside may be used with “cloud” device inside GNS3 topology, selected on the screenshot in Figure 25. Cloud icon can then be changed to any other icon for human-readability. Introduced to topology cloud device acts as a point with all GNS3 VM network interfaces configured in Figure 24.

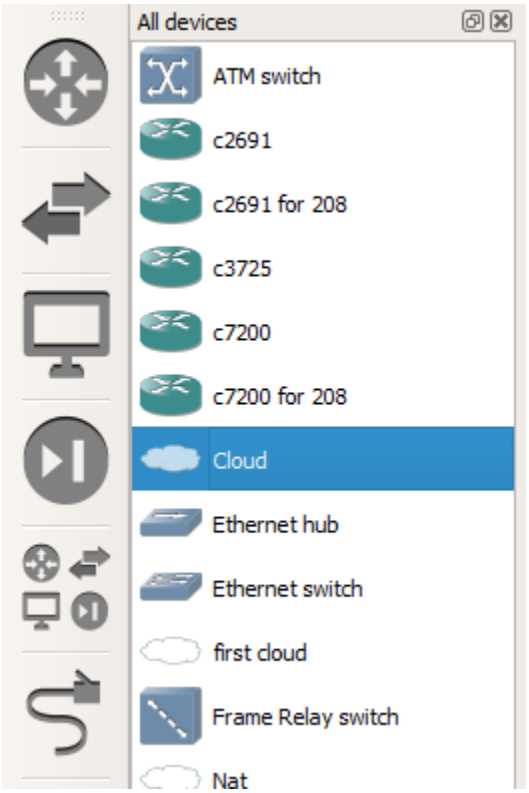



Figure 25. Cloud device in GNS3 interface along other devices

## 9.1 Case study: IPsec VPN tunnel between IOS and Mikrotik RouterOS

In this topology, and IPsec tunnel will be established between Cisco IOS device inside GNS3 and RouterOS in form of Cloud Hosted Router (CHR) running in a virtual machine outside GNS3.

## 9.2 Topology

Figure 26 describes the topology of this case study.

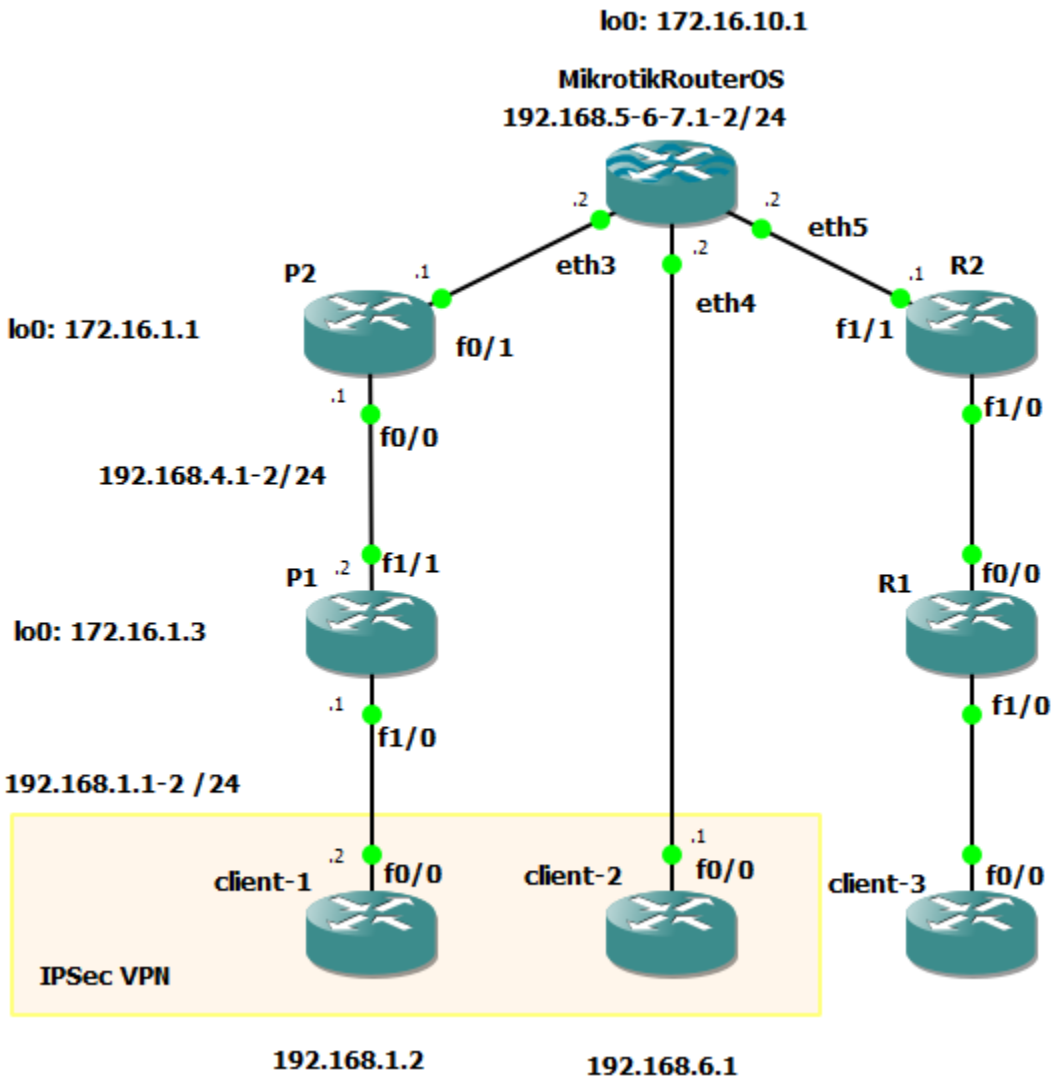



Figure 26. Topology with MikrotikRouterOS outside of GNS3 connected via eth3-5

In Figure 26 topology of this case study is drawn, with cloud icon in place of MikrotikRouterOS changed to router icon. GNS3 drawing tools allow putting all the necessary information into topology map, and changing icons makes it more clear to user.

## 9.3 Device roles

Table 4 describes device roles in this topology:

| | |
|---|---|
| client-1 and client -2 | Endpoint devices, located in networks, interconnected via IPsec tunnel. |
| P1 | Serving as IPsec tunnel endpoint device inside GNS3, Cisco IOS virtualized router |
| MikrotikRouterOS | Serving as IPsec tunnel endpoint device outside GNS3, MikrotikRouterOS virtual router |
| P2 | Routes packets, it does not participating in IPsec configuration. It represents third party network, over which data must be passed securely encapsulated |
| R2, R, client-3 | Reserved for future use, not functioning in this topology. |

Table 4. Description of device roles in topology. On the left is device name, on the right – its role

P1 and MikrotikRouterOS are tunnel endpoint devices; they will encapsulate traffic for networks 192.168.1.0/24 and 192.168.6.0/24 correspondingly. P1 will send encapsulated packets out of f1/1 to eth3 on MikrotikRouterOS and vice versa.
P2 routes packets, it does not participate in IPsec. It represents third party network, over which data must be passed securely encapsulated.

R2, R1 and client-3 are reserved for future use and not used.

### 9.4 Configuration

All devices are set up with IP addresses corresponding to their interfaces.
On links facing outside cloud, IP address is matching with IP address of eth3 interface. This is the configuration of P2 f0/1:

```
interface FastEthernet0/1
 ip address 192.168.5.1 255.255.255.0
```

ip ospf 1 area 0
duplex auto
speed auto

All routers except client-2 are configured to use OSPF as routing protocol to route packets, client-2 uses Mikrotik device as the default gateway. NAT and firewall are not used for simplicity; all devices have unique IP addresses.

### 9.4.1 Configuration of Mikrotik device:

Configuration is made by navigating to specific hierarchical menu levels and adding or removing configuration text commands. Configuration can be also made via web based GUI, called “WebFig”. First step in configuration of IPsec tunnel is a proposal. Proposals are located at “/ip ipsec proposal”.
A proposal with security algorithms used to connect to remote server. It contains authentication and encryption algorithm used for tunnel and additional settings

add name="vpnproposal- auth-algorithms=shal enc-algorithms=aes-256-cbc lifetime=30m pfs-group=none

“print” command prints all items in menu level, as pictured in Figure 27:

```
[admin@MikroTik] /ip ipsec proposal> print
Flags: X - disabled, * - default
 0  * name="default" auth-algorithms=md5 enc-algorithms=des lifetime=30m pfs-group=modp1024

 1    name="vpnproposal" auth-algorithms=sha1 enc-algorithms=aes-256-cbc lifetime=30m pfs-group=none
```

Figure 27. Result of “print” command issued in “/ip ipsec proposal” on Mikrotik device

Under item number 1 is proposal which were configured by previous command. Next step is to configure IPsec policy for tunnel, which contains networks participating in tunnel, addresses of tunnel endpoints, proposal used and other parameters. Policies are located at “/ip ipsec policy”. Adding policy is done with following command:

add src-address=192.168.6.0/24 src-port=any dst-address=192.168.1.0/24 dst-port=any protocol=all action=encrypt level=require ipsec-protocols=esp tunnel=yes sa-src-address=192.168.5.2 sa-dst-address=192.168.4.2 proposal=vpn-proposal priority=0 ph2-count=1

Final step is to configure information about peer. Peer is other endpoint of tunnel, which is P1 interface f1/1. In this configuration key of CiscoCisco777 is used on both ends. Peers are located at “/ip ipsec peer”. Peer is added with following command:

add address=192.168.4.2/32 auth-method=pre-shared-key secret=-CiscoCisco777- generate-policy=no policy-template-group=default exchange-mode=main send-initial-contact=yes nat-traversal=yes proposal-check=obey hash-algorithm=shal enc-algorithm=aes-256 dh-group=modp1024 lifetime=ld dpd-interval=2m dpd-maximum-failures=5

### 9.4.2 Configuration of P1

Other endpoint of tunnel has to mirror settings of P1 is IOS device, first step is to configure ISAKMP policy

```
crypto isakmp policy 1
 encr aes 256
 authentication pre-share
 group 2
```

Configure neighbor, which is the other tunnel endpoint:

```
crypto isakmp key CiscoCisco777 address 192.168.5.2      no-xauth
```

Configure transform set:

```
crypto ipsec transform-set vpntransformset esp-aes 256 esp-sha-hmac
```

```
mode tunnel
```

Configure crypto map

```
crypto map vpnmap 10 ipsec-isakmp
 description map for ipsec vpn
 set peer 192.168.5.2
set transform-set vpntransformset
match address aclforvpn
```

Access list with both networks at tunnel ends:

```
ip access-list extended aclforvpn
 permit ip 192.168.1.0 0.0.0.255 192.168.6.0 0.0.0.255
```

### 9.5 Testing

To test tunnel, ping command issued on client-2 to client1 address, network traffic between those clients is analyzed in point before entering the tunnel and along tunnel path. By analyzing the type of packets it is possible to determine, whenever they are encapsulated into encrypted protocol or not. Packets inside tunnel are encapsulated, and packets outside tunnel are not. Figure 28 describes capture points, one is between client-2 and MikrotikRouterOS, where there is no tunnel, and other is between P1 and P2 where packets should be encapsulated.

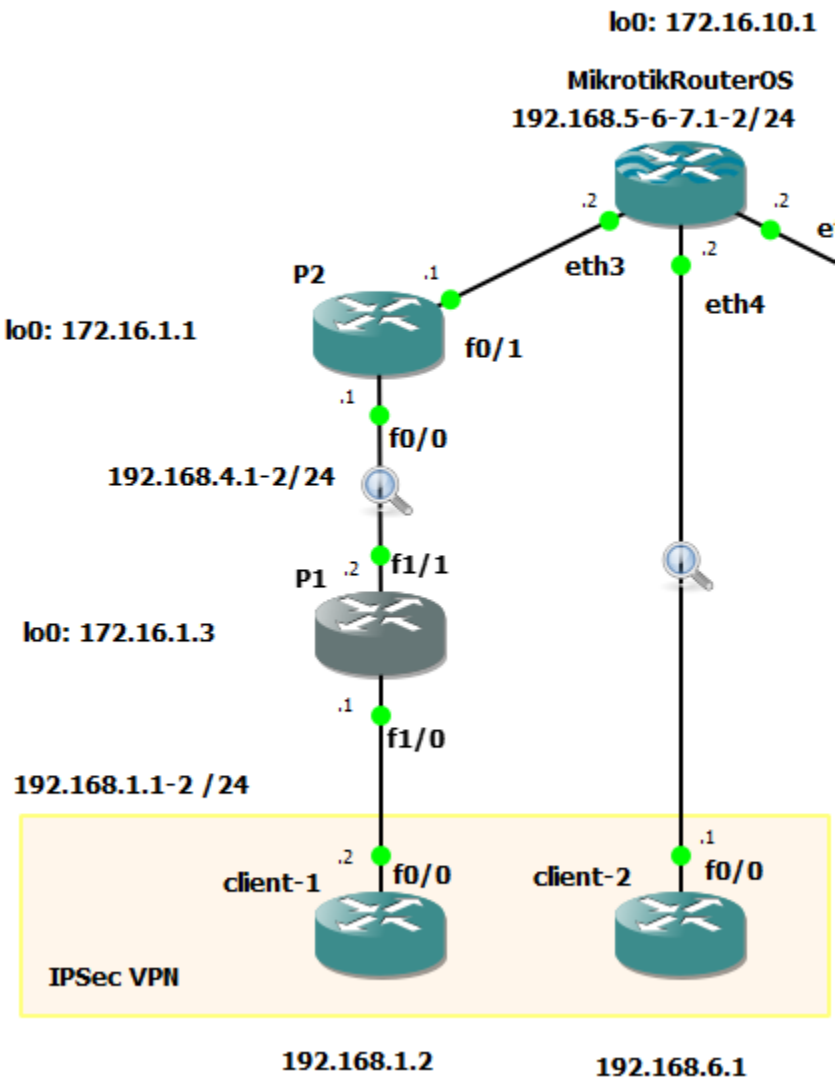
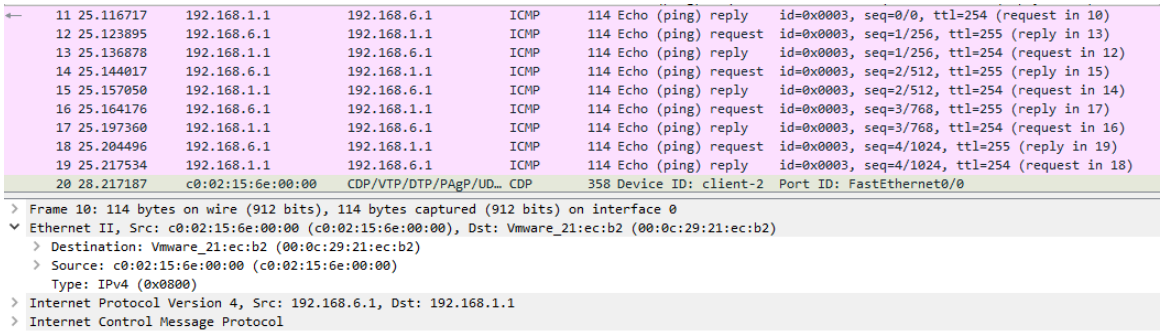



Figure 28. Packet capture points market with magnifying glass icon

```
Sending 5, 100-byte ICMP Echos to 192.168.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 20/26/36 ms
client-2#
```

Figure 29. Ping issued from client-2 to client-1 with successful results. Packets produces by this command will be captured in two points pictured in Figure 27

The output on Figure 29 shows, that there is a connection between routers. Results of packet capture between client-2 and MikrotikRouterOS is shown in Figure 30:



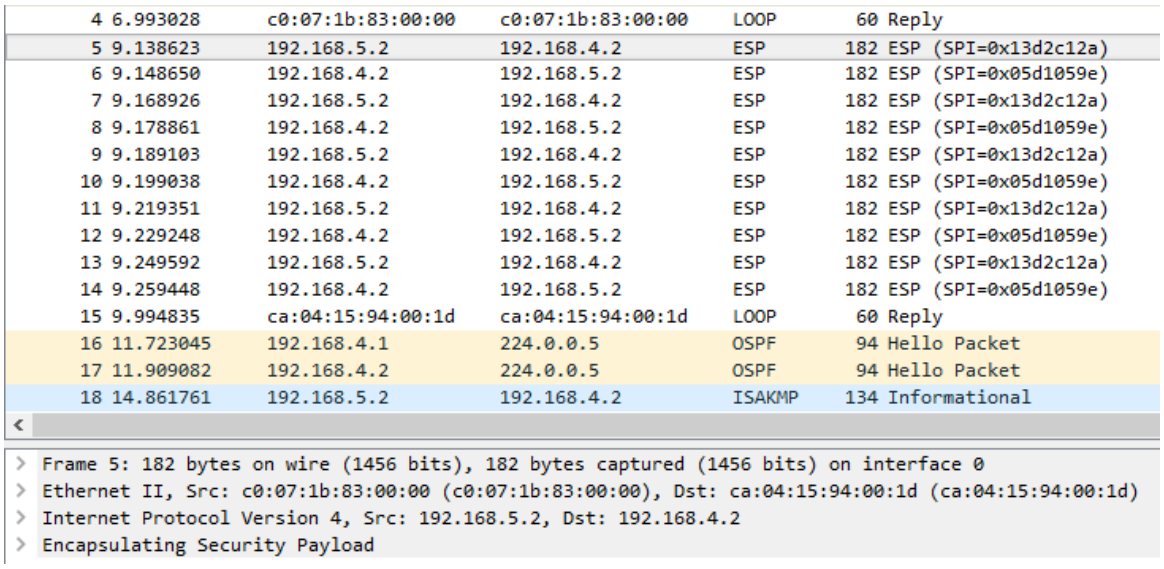
| | | | | | | |
|---|---|---|---|---|---|---|
| 11 | 25.116717 | 192.168.1.1 | 192.168.6.1 | ICMP | 114 | Echo (ping) reply id=0x0003, seq=0/0, ttl=254 (request in 10) |
| 12 | 25.123895 | 192.168.6.1 | 192.168.1.1 | ICMP | 114 | Echo (ping) request id=0x0003, seq=1/256, ttl=255 (reply in 13) |
| 13 | 25.136878 | 192.168.1.1 | 192.168.6.1 | ICMP | 114 | Echo (ping) reply id=0x0003, seq=1/256, ttl=254 (request in 12) |
| 14 | 25.144017 | 192.168.6.1 | 192.168.1.1 | ICMP | 114 | Echo (ping) request id=0x0003, seq=2/512, ttl=255 (reply in 15) |
| 15 | 25.157050 | 192.168.1.1 | 192.168.6.1 | ICMP | 114 | Echo (ping) reply id=0x0003, seq=2/512, ttl=254 (request in 14) |
| 16 | 25.164176 | 192.168.6.1 | 192.168.1.1 | ICMP | 114 | Echo (ping) request id=0x0003, seq=3/768, ttl=255 (reply in 17) |
| 17 | 25.197360 | 192.168.1.1 | 192.168.6.1 | ICMP | 114 | Echo (ping) reply id=0x0003, seq=3/768, ttl=254 (request in 16) |
| 18 | 25.204496 | 192.168.6.1 | 192.168.1.1 | ICMP | 114 | Echo (ping) request id=0x0003, seq=4/1024, ttl=255 (reply in 19) |
| 19 | 25.217534 | 192.168.1.1 | 192.168.6.1 | ICMP | 114 | Echo (ping) reply id=0x0003, seq=4/1024, ttl=254 (request in 18) |
| 20 | 28.217187 | c0:02:15:6e:00:00 | CDP/VTP/DTP/PAgP/UD… | CDP | 358 | Device ID: client-2 Port ID: FastEthernet0/0 |

```
> Frame 10: 114 bytes on wire (912 bits), 114 bytes captured (912 bits) on interface 0
v Ethernet II, Src: c0:02:15:6e:00:00 (c0:02:15:6e:00:00), Dst: Vmware_21:ec:b2 (00:0c:29:21:ec:b2)
    > Destination: Vmware_21:ec:b2 (00:0c:29:21:ec:b2)
    > Source: c0:02:15:6e:00:00 (c0:02:15:6e:00:00)
      Type: IPv4 (0x0800)
> Internet Protocol Version 4, Src: 192.168.6.1, Dst: 192.168.1.1
> Internet Control Message Protocol
```

Figure 30. Packet capture result in-between client-2 to MikrotikRouterOS

ICMP packets generated by “ping” command are shown in figure 30.

Results of packet capture between P1 and P2 is shown in Figure 31:



| | | | | | | |
|---|---|---|---|---|---|---|
| 4 | 6.993028 | c0:07:1b:83:00:00 | c0:07:1b:83:00:00 | LOOP | 60 | Reply |
| 5 | 9.138623 | 192.168.5.2 | 192.168.4.2 | ESP | 182 | ESP (SPI=0x13d2c12a) |
| 6 | 9.148650 | 192.168.4.2 | 192.168.5.2 | ESP | 182 | ESP (SPI=0x05d1059e) |
| 7 | 9.168926 | 192.168.5.2 | 192.168.4.2 | ESP | 182 | ESP (SPI=0x13d2c12a) |
| 8 | 9.178861 | 192.168.4.2 | 192.168.5.2 | ESP | 182 | ESP (SPI=0x05d1059e) |
| 9 | 9.189103 | 192.168.5.2 | 192.168.4.2 | ESP | 182 | ESP (SPI=0x13d2c12a) |
| 10 | 9.199038 | 192.168.4.2 | 192.168.5.2 | ESP | 182 | ESP (SPI=0x05d1059e) |
| 11 | 9.219351 | 192.168.5.2 | 192.168.4.2 | ESP | 182 | ESP (SPI=0x13d2c12a) |
| 12 | 9.229248 | 192.168.4.2 | 192.168.5.2 | ESP | 182 | ESP (SPI=0x05d1059e) |
| 13 | 9.249592 | 192.168.5.2 | 192.168.4.2 | ESP | 182 | ESP (SPI=0x13d2c12a) |
| 14 | 9.259448 | 192.168.4.2 | 192.168.5.2 | ESP | 182 | ESP (SPI=0x05d1059e) |
| 15 | 9.994835 | ca:04:15:94:00:1d | ca:04:15:94:00:1d | LOOP | 60 | Reply |
| 16 | 11.723045 | 192.168.4.1 | 224.0.0.5 | OSPF | 94 | Hello Packet |
| 17 | 11.909082 | 192.168.4.2 | 224.0.0.5 | OSPF | 94 | Hello Packet |
| 18 | 14.861761 | 192.168.5.2 | 192.168.4.2 | ISAKMP | 134 | Informational |

```
> Frame 5: 182 bytes on wire (1456 bits), 182 bytes captured (1456 bits) on interface 0
> Ethernet II, Src: c0:07:1b:83:00:00 (c0:07:1b:83:00:00), Dst: ca:04:15:94:00:1d (ca:04:15:94:00:1d)
> Internet Protocol Version 4, Src: 192.168.5.2, Dst: 192.168.4.2
> Encapsulating Security Payload
```

Figure 31. Packet capture result in-between P1 and P2

In packet capture result there are no ICMP packets; instead, there are ESP (Encapsulated Security Payload) packets between tunnel endpoints. This means that data is encapsulated and encrypted, tunnel is functional.

Cisco crypto engine on P1 shows, that there was encryption and decryption of packets, shown in figure 32:

```
P1# show crypto engine connection active
Crypto Engine Connections

   ID  Type    Algorithm           Encrypt  Decrypt LastSeqN IP-Address
    1  IPsec   AES256+SHA                0       22       22 192.168.4.2
    2  IPsec   AES256+SHA               23        0        0 192.168.4.2
 1001  IKE     SHA+AES256                0        0        0 192.168.4.2
```

Figure 32. Output of “show crypto engine connection active” command on P1

Just as Mikrotik device, any other network device can be connected with GNS3. There was a problem with unexpected disconnection of the IOS device ports pointed to cloud, or in other words, to a Microtik device. This problem disappeared after restart of IOS devices connected to that cloud.

## 10 GNS3 USAGE EXAMPLES

GNS3 can be used to prepare for CCNA/CCNP route exams, with Cisco ASA device added to a GNS3 topology, preparing to Cisco Security courses is also possible. GNS3 community as of 2017 offers paid and free courses for preparation to CCNA, CCNP and many other exams in networking industry. One of the type of courses available are video courses are represented in form of video tutorials with additional material available for course purchaser.

GNS3 is not able to run Cisco IOS for switches. This is due technical difficulties in emulating such hardware. (GNS3 official documentation 2017) GNS3 has an ability to add layer 2 switch (configuration window of which is shown in Figure 33) with basic functionality, including: access ports, VLANs, Dot1Q ports and QinQ ports, but no other configuration of the switch is possible.

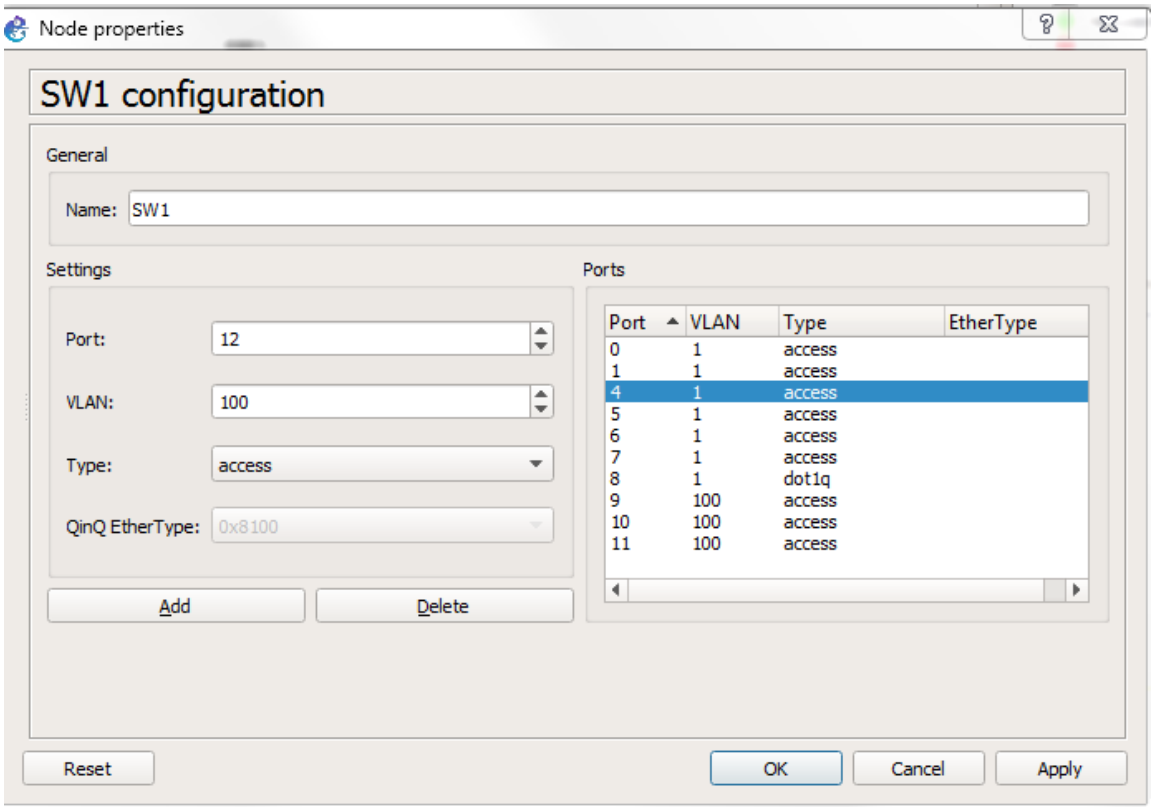



Figure 33. GNS3 built-in switch configuration window

CCNP certifications require switch configuration skills. (Wallace 2014, 35) Whenever GNS3 users want to run topology with Cisco switches, they have to use physical switches, which are connected by physical interfaces, as figure shows.

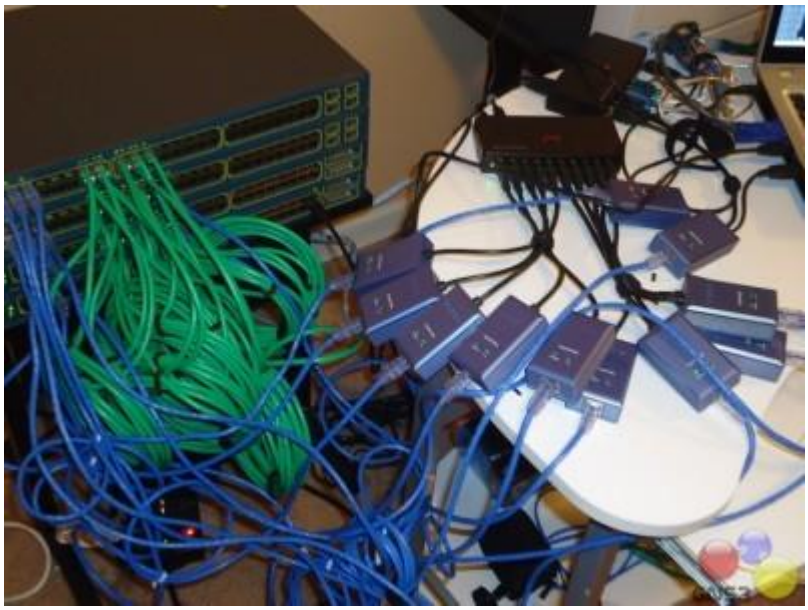


Figure 34. A user preparing to CCIE exams using GNS3, multiple network adapters and physical switches (Ziesemer 2016)

The need of multiple physical devices may cause troubles to end users, because of space and resource necessity, as shown in Figure 34. Techniques like trunking, may be used to reduce the number of physical network adapters needed, however, users must validate that hardware used supports needed technologies, and purchasing specific hardware may trouble user because of its cost and availability.

Students at XAMK may use GNS3 to prepare for exams by running Cisco's case studies at home, which may be more convenient. One of study method in XAMK is projects in which students are given a topic to research, and, possibly implement and test. If implemented in physical hardware, they need time to set up physical links, configure devices at each stage of re-implementing needed test scenario and return physical devices to their initial state after lesson ends. Their capability in creating topologies for different projects expands with GNS3 beyond limitations of the physical hardware.

## 11 COMPARISON TO OTHER NETWORK EMULATORS

There are other products on the network emulation market. Some of them do not specialize on emulation of specific hardware, others do. One of the examples is UNETLAB; it offers nearly same features, including functionality of VM, has a similar interface, but allows more types of router images to be run. As an advantage, there is a web interface, and disadvantage - excessive configuration when adding specific new devices.

NetSim Professional is a more complex solution with tools for traffic engineering, the packet analyzing, protocol development and other tools. It is proprietary and paid software.

Cisco packet tracer, used by Cisco instructors in the learning process is very limited in functionality, but is constantly updated and has tools to check how correct configuration is.

In XAMK University, there is Virtual Laboratory, created by Jaakko Nurmi (2016). It provides an environment where the users can practice configuring networking devices and servers, has web interface and emulates IOS-XR devices. It has disadvantages. First, topology needs to be written by hand instead of GUI, but it's not necessary significant, since students usually open one of pre-configured topologies for their studies. Second is that IOS-XR routers require significant (higher than emulating IOS devices) amount of resources to run. This is not a problem in that particular area, because XAMK has datacenter with significant computational resources.
There are also other solutions for network emulation, they develop and add new features constantly, giving users better and better options for their needs.

## 12 CONCLUSION

During this research, multiple virtual topologies with different devices were tested. All schemes chosen for emulation were working fine with an exception of sudden connection loss with between Mikrotik and IOS devices in chapter 9, which had only temporary effect.

Research shows, that GNS3 is capable of running complex devices interconnected and configured in multiple ways. This was enough for topologies tested during research, but users, who need to emulate hardware unsupported by the GNS3 may need to either emulate it outside of the GNS3's virtual emulated space or use other emulators. Tools supplied with the GNS3 and their integration allows easy packet capture and monitoring.

With a knowledge of what specific router is capable of, and how it performed during test scenarios, it is possible to interpret which use cases it may be capable of participating in. While using IOS alone should be enough for CCNA and CCNP routing exercise practicing, possible usage cases of GNS3 are much wider, because of its capability to successfully connect to outside word and use other networking and endpoint devices.

Other network emulating software exists and plays a completive role on the market. There are solutions for specific tasks, as well as multi-tools, a brief examination of their specifications may give an idea, that they can be used in combination with each other or alone for practicing in configuring computer networks. It is nearly safe to say that for each topic that can be studied, there is a specific tool for practicing in virtually building topic-related scenarios.

Overall, the tendency of virtualizing hardware gives benefits in every area, by saving time, bringing convenience and allowing new configurations, impossible in the past.

**REFERENCES**

Auda Y. 2013. Understating Cisco IOS v15 Licenses IOS-XR WWW document. Available at: https://learningnetwork.cisco.com/docs/DOC-20321 [Accessed 9 June 2017].

Cisco Inc. P. 2010. CCNP ROUTE 6.0 Student Lab Manual. Indianapolis: Cisco press.

GNS3 official documentation 2017. WWW document. Available at: https://docs.gns3.com/ [Accessed 31 May 2017].

Nurmi, J. 2016. Implementation of Nested Virtual Laboratory System. WWW document. Available at: https://www.theseus.fi/bitstream/handle/10024/107061/Nurmi_Jaakko_Thesis.pdf?sequence=1 [Accessed: 12 May 2017].

Stretch, J. 2011 Creating an MPLS VPN. WWW document. Available at: http://packetlife.net/blog/2011/may/16/creating-mpls-vpn/ [Accessed 20 April 2016].

Verma, L. 2016 Configuration and Verification of Layer 3 INTER-AS MPLS VPN Option B using IOS and IOS-XR WWW document. Available at: http://www.cisco.com/c/en/us/support/docs/multiprotocol-label-switching-mpls/mpls/200557-Configuration-and-Verification-of-Layer.html [Accessed 16 May 2017].

Wallace, K. P. 2014. Routing and Switching ROUTE 300-101 Official Cert Guide. Indianapolis: Cisco press.

Ziesemer, M. 2016. USB to Ethernet adapter supporting multiple virtual LANs 2017 WWW document Available at: https://hardwarerecs.stackexchange.com/questions/2422/usb-to-ethernet-adapter-supporting-multiple-virtual-lans [Accessed 31 May 2017].

Appendix 1/1

Routers' Pedge1 and Pedge2 complete useful configuration. Unused connection interfaces were deleted.

```
!      Pedge2

service timestamps debug datetime msec

service timestamps log datetime msec

no service password-encryption
!
hostname Pedge2
!
boot-start-marker
boot-end-marker
!
!
no aaa new-model
memory-size iomem 5
no ip icmp rate-limit unreachable
ip cef
!
!!
ip vrf Customer_A
 rd 65000:1
 route-target export 65000:1
 route-target import 65000:1
!
ip vrf Customer_B
 rd 65000:2
 route-target export 65000:2
 route-target import 65000:2
!
no ip domain lookup
!
multilink bundle-name authenticated
!
!!
archive
 log config
  hidekeys
!
!
!
!
ip tcp synwait-time 5
ip ssh version 1
!
!
interface Loopback0
 ip address 10.0.0.4 255.255.255.255
 ip ospf network point-to-point
 ip ospf 1 area 0
!
interface FastEthernet0/0
 ip address 10.0.9.10 255.255.255.252
 ip ospf 1 area 0
 duplex auto
 speed auto
 mpls ip
```

```
!
no ip address
 shutdown
 clock rate 2000000
!
interface FastEthernet0/1
 description to CE2A VRF Customer_A
 ip vrf forwarding Customer_A
 ip address 10.0.2.1 255.255.255.252
 ip ospf 2 area 0
 duplex auto
 speed auto
!
interface FastEthernet1/0
 description to CE2B VRF Customer_B
 ip vrf forwarding Customer_B
 ip address 10.0.2.5 255.255.255.252
 ip ospf 3 area 0
 duplex auto
 speed auto
!
router ospf 2 vrf Customer_A
 router-id 10.0.2.1
 log-adjacency-changes
 redistribute bgp 65000 subnets
!
router ospf 3 vrf Customer_B
 router-id 10.0.2.5
 log-adjacency-changes
 redistribute bgp 65000 subnets
!
router ospf 1
 router-id 10.0.0.4
 log-adjacency-changes
!
router bgp 65000
 no synchronization
 bgp log-neighbor-changes
 neighbor 10.0.0.3 remote-as 65000
 neighbor 10.0.0.3 update-source Loopback0
 no auto-summary
 !
 address-family vpnv4
  neighbor 10.0.0.3 activate
  neighbor 10.0.0.3 send-community extended
 exit-address-family
 !
 address-family ipv4 vrf Customer_B
  redistribute ospf 3 vrf Customer_B
  no synchronization
 exit-address-family
 !
 address-family ipv4 vrf Customer_A
  redistribute ospf 2 vrf Customer_A
  no synchronization
 exit-address-family
!
ip forward-protocol nd
!
!
no ip http server
no ip http secure-server
!
no cdp log mismatch duplex
!
```

```
control-plane
!
!
```

## !  Pedge1

```
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname Pedge1
!
boot-start-marker
boot-end-marker
!
!
no aaa new-model
memory-size iomem 5
no ip icmp rate-limit unreachable
ip cef
!
!!
ip vrf Customer_A
 rd 65000:1
 route-target export 65000:1
 route-target import 65000:1
!
ip vrf Customer_B
 rd 65000:2
 route-target export 65000:2
 route-target import 65000:2
!
no ip domain lookup
!
multilink bundle-name authenticated
!!
archive
 log config
  hidekeys
!
!!
ip tcp synwait-time 5
ip ssh version 1
!!
interface Loopback0
 ip address 10.0.0.3 255.255.255.255
 ip ospf network point-to-point
 ip ospf 1 area 0
!
interface FastEthernet0/0
 ip address 10.0.9.6 255.255.255.252
 ip ospf 1 area 0
 duplex auto
 speed auto
 mpls ip
!
interface Serial0/0
 no ip address
 shutdown
 clock rate 2000000
!
interface FastEthernet0/1
 description to CE1A VRF Customer_A
 ip vrf forwarding Customer_A
```

```
 ip address 10.0.1.1 255.255.255.252
 ip ospf 2 area 0
 duplex auto
 speed auto
!
shutdown
 clock rate 2000000
!
interface FastEthernet1/0
 description to CE1B VRF Customer_B
 ip vrf forwarding Customer_B
 ip address 10.0.1.5 255.255.255.252
 ip ospf 3 area 0
 duplex auto
 speed auto
!
router ospf 2 vrf Customer_A
 router-id 10.0.1.1
 log-adjacency-changes
 redistribute bgp 65000 subnets
!
router ospf 3 vrf Customer_B
 router-id 10.0.1.5
 log-adjacency-changes
 redistribute bgp 65000 subnets
!
router ospf 1
 router-id 10.0.0.3
 log-adjacency-changes
!
router bgp 65000
 no synchronization
 bgp log-neighbor-changes
 neighbor 10.0.0.4 remote-as 65000
 neighbor 10.0.0.4 update-source Loopback0
 no auto-summary
 !
 address-family vpnv4
  neighbor 10.0.0.4 activate
  neighbor 10.0.0.4 send-community extended
 exit-address-family
 !
 address-family ipv4 vrf Customer_B
  redistribute ospf 3 vrf Customer_B
  no synchronization
 exit-address-family
 !
 address-family ipv4 vrf Customer_A
  redistribute ospf 2 vrf Customer_A
  no synchronization
 exit-address-family
!
ip forward-protocol nd
!
!
no ip http server
no ip http secure-server
!
no cdp log mismatch duplex
!
```