Ivan Paulov

Routing in a Virtualised Environment with RouterOS

Helsinki Metropolia University of Applied Sciences

Bachelor of Engineering

Information Technology

Thesis

16 November 2015

Author(s) Title Number of Pages Date	Ivan Paulov Routing in a Virtualised Environment with RouterOS 54 pages + 2 appendices 16 November 2015
Degree	Bachelor of Engineering
Degree Programme	Information Technology
Specialisation option	Data Networks
Instructor(s)	Erik Pätynen, Senior Lecturer

The purpose of this thesis was to test network implementation in a virtualised environment using virtual machines running Mikrotik RouterOS on top of VMware Fusion hypervisor. The aim was to create and test virtualised networks using the technologies mentioned above in a manner similar to a networking laboratory exercise.

This study starts by describing the basic concepts in computer networking, equipment, protocols and methods. After the concepts have been explained, detailed instructions are given on how to set up a virtualised networking laboratory.

Two laboratory exercises were planned and implemented. As a result, a host of networking technologies were successfully tested to work: dynamic routing, tunnelling, load balancing, static routing, network address port translation and backup routes. Examples of configurations were given for both a graphical user interface and a command line interface interaction. Limitations and problems of the studied technologies for use as a virtualised networking laboratory were identified and elaborated upon.

As a conclusion, the aim of the study was met. RouterOS and VMware Fusion are user friendly and work together without issues. The instructions in this thesis can be used as a guide to create one’s own network implementations. Interesting areas for further research include implementation of switching and/or software defined networking using OpenFlow with RouterOS.

Keywords	RouterOS, OS X, VMware, virtualisation, routing

Contents

Appendices

List of Abbreviations

AS - Autonomous System

BDR - Backup Designated Router

BGP - Border Gateway Protocol

CCITT - International Telegraph and Telephone Consultative Committee

CD - Compact Disc

CDP - Cisco Discovery Protocol

CNA - Cisco Networking Academy

CPU - Central Processing Unit

DHCP - Dynamic Host Control Protocol

DMG - Universal Disk Image Format Apple Disk Image

DNS - Domain Name System

DR - Designated Router

DVD - Digital Versatile Disc

ECMP - Equal Cost Multipath Routing

EGP - External Gateway Protocol

FTP - File Transfer Protocol

GNS3 – Graphical Network Simulator-3

GPS - Global Positioning System

GRE - General Routing Encapsulation

GUI - Graphical User Interface

HDD - Hard Drive

HTTP - Hypertext Transfer Protocol

IDE - Integrated Drive Electronics, also known as Parallel ATA

ICMP – Internet Control Message Protocol

IGP - Internal Gateway Protocol

IOS – Internetwork Operating System

IP - Internet Protocol

IPsec – Internet Protocol Security

ISO - International Organisation for Standardisation

ISP – Internet Service Provider

ITU-T - Telecommunication Standardisation Sector of the International Telecommunication Union

KVM - Kernel Virtualisation Modules

LSA - Link State Advertisement

MAC - Media Access Control

MD5 - Merkle–Damgård Hashing Algorithm 5

MPLS – Multiprotocol Label Switching

NAT - Network Address Translation

NAPT - Network Address Port Translation

NFS - Network File System

NIC - Network Interface Card

OS - Operating System

OSI - Open Systems Interconnection

OSPF - Open Shortest Path First

PC - Personal Computer

PDU - Protocol Data Unit

PoE - Power over Ethernet

RAM - Random Access Memory

RFC - Request for Comments

RIP - Routing Information Protocol

SCSI - Small Computer System Interface

SDN – Software Defined Networking

SIP - Session Initiation Protocol

SNMP - Simple Network Management Protocol

SOHO - Small Office Home Office

SSH - Secure Shell

TCP - Transmission Control Protocol

UDP - User Datagram Protocol

VLAN - Virtual Local Area Network

VM - Virtual Machine

1 Introduction

In this thesis I explore how to comprehensively test a computer network implementation using a virtualised environment based on virtual machines (VMs) running Mikrotik RouterOS. The testing will be done in the form of a laboratory exercise, which is well known to all students attending courses offered by the Cisco Networking Academy (CNA) programme. For the purposes of this thesis no physical network will be built, however. The virtualised environment is built using an identical operating system (OS) that is used on all Mikrotik's physical routers[1]. The virtualised environment precludes the use of certain functionality offered by RouterOS (such as Power over Ethernet (PoE), Global Positioning System (GPS) and Wireless networking related features). However, it does not prevent construction of the common network topologies.

Mikrotik Ltd. is a Latvian company founded in 1996 to develop routers and wireless ISP systems. It is based in Riga and employs (as of November 2015) 160 people. [1] The low price point of Mikrotik's products suggests specialisation in developing markets, but based on my personal experience Mikrotik products are also popular for example in the Czech Republic. In this thesis I use Mikrotik's networking operating system RouterOS of version 6.32.2 which was published in September 2015 and is available for download from Mikrotik's website.

The work starts with a brief review of the fundamental concepts of network technologies with emphasis on technologies used in this thesis. The work is carried out on VMware Fusion, version 8.0.1, running on top of OS X 10.11 El Capitan. The intended outcome of this thesis are laboratory exercises utilising RouterOS and VMware Fusion to demonstrate in practice the concepts elaborated in the theoretical background part of this thesis. Other goals of this work are to describe limitations and distinctive features of this setup for the purpose of creating and running a virtual networking laboratory.

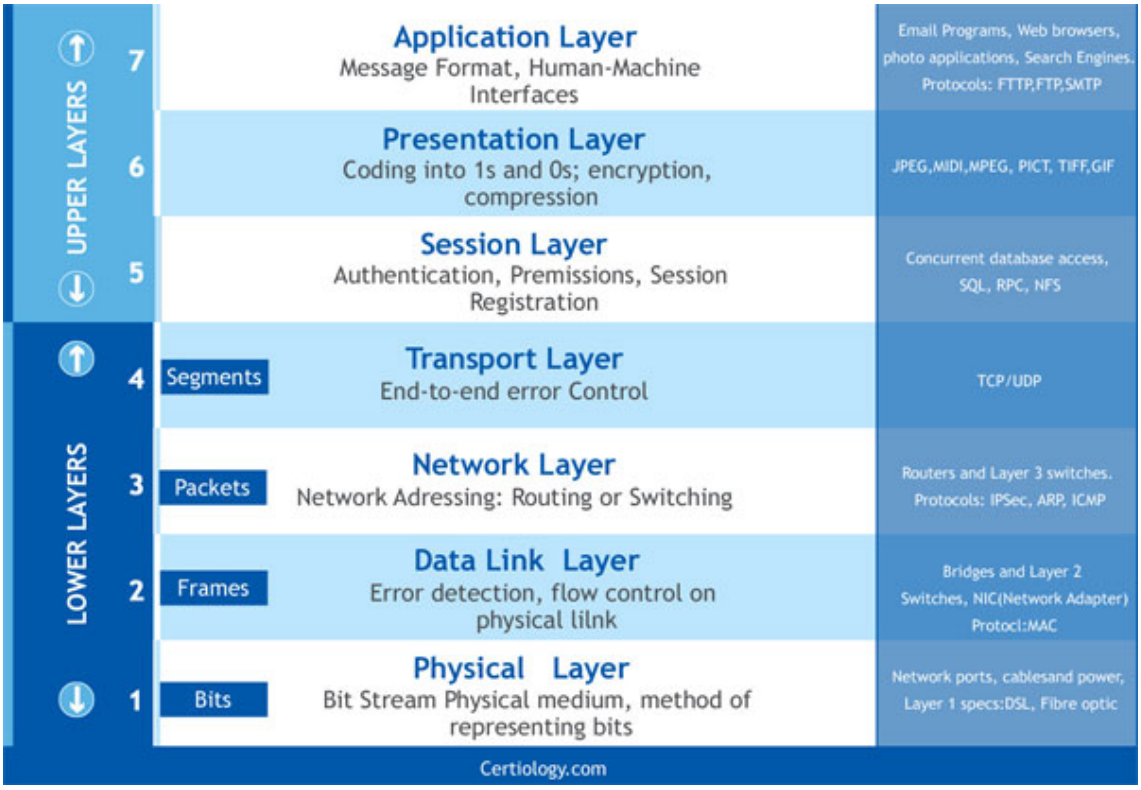
The instructions in this thesis can be used with minor adjustments on Windows and Linux OS based machines running VMware Desktop 10.

[1] Mikrotik SwOS (which stands for Switch OS) is used on all models of switches offered by Mikrotik

2 Theoretical Background

2.1 OSI Reference Model

Open Systems Interconnection (OSI) reference model is a framework for modelling transport of information in the data network using a layered approach. The model was developed at the end of the 1970s independently by two organisations, International Organisation for Standardisation (ISO) and International Telegraph and Telephone Consultative Committee (CCITT), which later became Telecommunication Standardisation Sector of the International Telecommunication Union (ITU-T). The resulting joint standard was published in 1984 as ISO 7498 [2].



7		**Application Layer** Message Format, Human-Machine Interfaces	Email Programs, Web browsers, photo applications, Search Engines. Protocols: FTTP,FTP,SMTP
6		**Presentation Layer** Coding into 1s and 0s; encryption, compression	JPEG,MIDI,MPEG, PICT, TIFF,GIF
5		**Session Layer** Authentication, Premissions, Session Registration	Concurrent database access, SQL, RPC, NFS
4	Segments	**Transport Layer** End-to-end error Control	TCP/UDP
3	Packets	**Network Layer** Network Adressing: Routing or Switching	Routers and Layer 3 switches. Protocols: IPSec, ARP, ICMP
2	Frames	**Data Link Layer** Error detection, flow control on physical lilnk	Bridges and Layer 2 Switches, NIC(Network Adapter) Protocl:MAC
1	Bits	**Physical Layer** Bit Stream Physical medium, method of representing bits	Network ports, cablesand power, Layer 1 specs:DSL, Fibre optic

Figure 1. OSI Reference Model. Copied from Certiology, 2015 [3]

Figure 1 illustrates how data moves through different layers of the OSI model using different protocols. It also illustrates the purposes of lower layers (1-4) which are concerned with routing and delivery of data and upper layers which handle presentation of data, human-machine interfaces and coordination of formats. Every layer has a different purpose and every layer uses the services of the layer directly below it and serves the layer directly above.

The layers and their functions are as follows:

- **Physical layer** is the layer on which bits are transferred in the form of electrical, light or radio signals. This layer deals with physical transport of data and with related media and connectors. Example technologies: Universal Serial Bus, Fibre Channel, WiFi.

- **Data link layer** is responsible for framing of packets into correct form for transport over the physical layer and for detection of errors during the transport. The Ethernet protocol works on this layer.

- **Network layer** is responsible for routing Internet Protocol (IP) packets between the devices on different networks using a variety of protocols. Routers operate on this layer.

- **Transport layer** provides error recovery, flow control and ordered data transfer and segmentation. Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) operate on this layer. [4]

- **Session layer** manages starting, controlling and ending sessions [4, 36] between two end-devices. Network File System (NFS) is one of this layer's protocols [3].

- **Presentation layer** defines and negotiates data formats of transported data. Encryption is the presentation layer's service. [4, 36; 2]

- **Application layer** is the layer closest to the user. It provides an interface for applications needing to communicate over the network. Protocols on this layer include, for example, Hypertext Transfer Protocol (HTTP), Simple Network Management Protocol (SNMP), File Transfer Protocol (FTP) and other. [4, 36-37; 3]

Since the purpose of this thesis is to test the routing functionality of RouterOS, it deals with network layer traffic exclusively.

2.2 Switch

“In communication systems, switch is a mechanical, electromechanical, or electronic device device for making, breaking or changing the connections in or among circuits.” [5, S-34]

Principle of operation

Switch, also called a multiport bridge, operates on the data link layer of the OSI model. Switches forward frames based on their Media Access Control (MAC) address. Any time a switch receives a frame on its interface, it will record the frame’s source MAC address and stores it in its MAC address table. In a case where it receives a frame destined to a host not yet in a MAC table, it will flood the frame from all but receiving interface.

Switches are an important part of modern networks. By virtue of their operating principle they divide collision domains (every switch port creates a collision domain of its own) and allow for full duplex operation of Ethernet networks when each switch port is directly connected to the end device. Switching in a virtual environment differs from switching in a physical environment. These differences are discussed in more depth in section 2.8.

2.3 Router

"Routers connect networks. A router links computers to the Internet, so users can share the connection. A router acts as a dispatcher, choosing the best path for information to travel so it's received quickly." [6]

Routers, operating on the network layer of the OSI reference model, make it possible to transport IP packets between different (sub)networks. In order for data to move from one (sub)network to another, there needs to be a device (or devices) between them providing packet switching and path selection. The main task of a router is to forward packets between different (sub)networks and select a path from a routing table for an IP packet to reach its destination. Routers use the network layer’s logical addressing facility (IP addresses) and other factors (called routing metrics) to construct routing tables on the basis of which they make routing decisions.

Routing

In order for IP packets to reach their destination, they need to be routed either dynamically or statically.

Static routing refers to manually configuring an IP route. A static IP route needs to contain at least two pieces of information: the destination network address and the direction how to get there. The direction can be given in two ways: direction to an outbound interface on the router or as an IP address of the next router on a path to the destination network. The router implicitly positions itself as an origin of the route. The commands that can be used to add a static route are usually a variation of the following:

```
ip route 198.177.24.0 255.255.255.0 11.12.13.14
```

where the sequence of arguments refers to the following scheme:

```
ip route [destination IP address] [destination subnet
mask] [outbound port/interface/next hop IP address]
```

In practice static routing is used for very small networks and special purposes only (since it does not react to changes in the network's physical and logical topology and does not scale well) and routing is a responsibility of dynamic routing protocols. Dynamic routing means that routers use a routing protocol to exchange routing information.

Dynamic routing protocols belong to two broad groups: distance vector routing protocols and link-state routing protocols. Distance vector routing protocols utilise the Bellman-Ford algorithm to calculate the distance between two networks and base the routing decisions on that metric. Hop count is often used as a distance metric. Routers store information about their distance to various networks in the routing table and exchange it with their neighbours. Over time the distance information propagates to all routers and so all routers know the distance to all networks. [7]

Link-state protocols use the shortest-path first algorithm and make decisions on the basis of the shortest path between the networks. Routers using link-state routing protocols keep a complete topology database of the network and regularly check connectivity to its various parts while exchanging information about link states with other routers. The amount of metrics that can be used is wider compared to the distance vector routing protocol. For example, a bandwidth of a link can be a part of a metric. Link-state protocols provide faster convergence rates than distance vector protocols at the price of higher router resource utilisation. [7]

Both of these protocol groups can be further subdivided into internal gateway and external gateway protocols. Internal gateway protocols (IGP) route traffic within autonomous systems (AS). Autonomous systems are (inter)networks under a common administration that share a common routing strategy [8, 732]. Interior gateway routing protocols operate within the confines of a single AS. Open Shortest Path First (OSPF) and Routing Information Protocol (RIP) are examples of interior gateway routing protocols. External gateway routing protocols route traffic between autonomous systems. Border Gateway Protocol (BGP) is an external gateway routing protocol. Routers can run more than one routing protocol at a time, or even several instances of the same routing protocol simultaneously. It is possible to redistribute static routes and dynamic routes between routing protocols. This results in more efficient utilisation of the routers' resources.

2.4 OSPF

OSPF is a link-state IGP routing protocol. It is defined in RFC 1131 and 2328. OSPF is a hierarchical routing protocol with the largest entity being the autonomous system itself. To reduce the need for path recalculation in big networks (and the performance and increased traffic requirements it bears), it is often implemented in multiple small(er) routing areas instead of a single area for the whole AS. Every area has its own ID. All areas need to be connected to the backbone area 0 (also called default area). Routers only recalculate link state databases when topology changes in the area they are part of. Routers can take part in multiple areas, and in that case they are called area border routers and carry multiple topology databases (one per each connected area). Routes can be summarised at the edges of routing areas, which leads to smaller routing tables. [9, 114-115, 167-168]

Routers taking part in the OSPF process discover neighbouring routers using the Hello protocol. Once the neighbour relationship is established between two routers they engage in periodic exchange of link state advertisement (LSA) packets, which contain information about routes, links and their status/changes. As of November 2015 there are 11 types of LSA described in RFCs for OSPF v 1, 2 and 3. Each LSA type has a specific purpose and payload (such as information about link state change or a new route). Areas of different types have been developed in order to minimise the amount of LSA advertisements exchanged in the network. [10]

Table 1 demonstrates the relationship between area types and LSA types allowed in these areas. Note that OSPFv2, which is used in this thesis, only recognises seven LSA types and table 1 refers to those.

Table 1. OSPF Area and LSA Types Relationship. Data gathered from Stretch J. 2008. [11]

Area Type	LSA Types Allowed
Standard area	1, 2, 3, 4, 5
Not-so-stubby area	1, 2, 3, 7
Stub area	1, 2, 3
Totally stubby area	1, 2

OSPF area type defines the relationship of areas to other routes and LSAs define the role of the router in those areas. [11]

2.5 Network Address Translation

Basic Network address translation (NAT) is a method of transparent mapping of addresses in one address space to addresses in another address space (on one-to-one basis). Network address port translation (NAPT) is a method of mapping multiple addresses and their TCP/UDP ports to a single address and its TCP/UDP ports. This allows for hosts on private networks (which are defined in RFC 1918) to connect to the public Internet, that is, to networks and hosts with globally unique routable addresses. These two methods are also called traditional NAT. [12, 27]

Traditional NAT (also called masquerading NAT) works on the premise that not all hosts need an end-to-end Internet connectivity. Those hosts can utilise private IP addresses (which are not globally unique and therefore not routable) and use NAT to communicate with the outer world. A device which provides address translation is a multihomed router with interfaces on both private and public networks. NAPT operates as follows: a host on an internal (usually privately addressed) network initiates a connection to the host on the public Internet. The router providing NAPT records the incoming IP packet’s source IP address and TCP/UDP port number, rewrites those with the IP address of its outbound interface and assigns new port numbers. Then it stores this information in a table of connections. If the host on the public Internet replies to the original sender, NAT Router will look up the connection in the connection table and rewrite the target IP address and port numbers in the IP packet header accordingly. [13]

A traditional NAT operation precludes the possibility of host on the public Internet initiating the connection (the traffic can only return if the connection was made from inside the NATed area, because only then the address translation table record is created). Also, the NAT table records are periodically trimmed to allow for reuse of port numbers. To overcome this limitation a technique called port forwarding or static NAT can be used. In this case a static (permanent) pairing of IP addresses and TCP/UDP ports is created. If the host on the public Internet knows the IP address/TCP/UDP combination of the masqueraded host, it can then initiate the connection (this is used when resources such as web and file servers behind NAT need to be accessed from the outside of the privately addressed network).

NAT is one of the methods that allowed for conservation of IPv4 address space by limiting the number of public IP addresses that an entity on the Internet requires. A downside of NAT is that it is resource-intensive and its utilisation complicates the operation of protocols such as FTP or Session Initiation Protocol (SIP).

2.6 Load Balancing

Load balancing is a process of utilising multiple Internet connections in a way that proportionately distributes Internet traffic between all connections. It can be symmetrical or asymmetrical and it is useful if a host on the network has higher bandwidth requirements than a single connection would satisfy. [14]

Multiple implementations of load balancing are available, both static and dynamic. Load balancing is similar to, but distinct from channel bonding which operates on the network and data link layers of the OSI reference model. In contrast to channel bonding, load balancing operates on the transport layer and utilises IP address TCP/UDP port combinations as a basis of the division of traffic.

The load balancing method used in this thesis is equal cost multipath (ECMP) load balancing. In this method of load balancing, the traffic is spread over multiple equal cost paths (where cost refers to the routing metric used). Default behaviour is round robin balancing, but it can also operate in an asymmetrical mode. RouterOS is capable of dividing traffic on the per connection and per packet basis using ECMP. [15]

In ECMP load balancing the hop (router) makes the load balancing decision independently. As with NAT, ECMP load balancing will cause some protocols to operate suboptimally, or even fail [15].

2.7 Tunnelling and Generic Routing Encapsulation

Tunnelling is a name for a multitude of encapsulation methods. A tunnelling protocol is a protocol that encapsulates another protocol PDU with its own header in order to allow a user access to services not provided or supported by the network. [16] Tunnelling protocols often break OSI hierarchy – for example in the case of general routing encapsulation (GRE) an IP packet can be encapsulated within another IP packet. There are three main uses of tunnelling:

1. Offering services which might not be available on the current network (such as encapsulating IPv6 packets within IPv4 packets)
2. Offering services which might be unsafe to offer otherwise – such as providing a tunnel instead of a company's internal address to remote workers
3. Some tunnelling protocols offer a possibility to encrypt the traffic. In this case tunnels offer a possibility to hide the contents of the network traffic (an example of an encrypting tunnelling protocol: Internet Protocol Security protocol suite (IPsec)).

GRE is a tunnelling protocol developed by Cisco Systems that can encapsulate multiple other protocols' PDUs. It is described in RFC 2784. GRE provides stateless tunnels (stateless in this context refers to the fact that endpoints do not have information about the state of the other endpoint. If one of the endpoints is unreachable, the other one will not automatically shut the tunnel down). [17]

2.8 Virtualisation and Networking in Virtualised Environments

"Virtualisation is abstraction of one computing resource from another computing resource." [18, XXV]

Virtualisation means, for purposes of my thesis, abstraction of the operating system from the underlying hardware – that is a possibility to run multiple operating systems (or multiple instances of the same OS) simultaneously on the same physical host. These operating systems run on top of a virtual machine (VM). From the point of view of the guest OS the VM is a computer – with the central processing unit (CPU), hard disk drive (HDD), random access memory (RAM), network interface card (NIC) included. From the point of view of the host OS, or from the point of view of the program that assigns these computing resources to VMs, the hypervisor, a VM is just a collection of files containing data and configuration settings. A hypervisor is therefore a program that can divide computing resources of the host computer and keep the illusion of being run on dedicated hardware for the guest OS. Two types of hypervisors exist:

1. Type 1 hypervisor: A bare metal hypervisor which runs directly on the hardware, without a need for OS to be installed. Examples: VMware ESXi, Linux Kernel Virtualisation Modules (KVM)
2. Type 2 hypervisor. Runs on top of the host OS, that provides it with input/output (I/O) and memory management support. Examples: VMware Fusion, Oracle VirtualBox [18]

Hypervisors do consume computing resources, but allow for higher utilisation of host machine resources compared to typical single OS/purpose installations.

In order for VMs to communicate a standardised way of connecting needs to be employed. VMs use the same networking paradigms as their physical counterparts with some important differences:

- A VM can have an arbitrary amount of virtual NICs limited only by hypervisor support. These can be added and removed on the fly.
- VMs can not connect to each other directly – in order to facilitate networking a hypervisor needs to control how VMs connect to each other and the outside world. The tool a hypervisor uses for this purpose is a virtual switch (also called vSwitch by VMware).

- A hypervisor can run several vSwitches. vSwitches can have an arbitrary number of virtual switchports limited by the hypervisor and host hardware capabilities.
- vSwitch can connect to the NIC(s) on the host machine in several ways: directly taking control of the interface (cutting out the host OS in the case of type two hypervisors), indirectly using NAT or bridging (in the bridging mode the VM will appear as another host on the network the host computer is connected to), or not at all – in this case the VMs will be internal to the host only.
- VMs can be connected to several vSwitches simultaneously
- vSwitches will drop any frames coming form the outside world with unknown destination MAC addresses (they know exactly what is connected to them so flooding of the frame from all interfaces to find a potential recipient is not necessary) [19]
- Some hypervisors support distributed virtual switches, that is, virtual switches on several host machines that act as a single switch [19]
- VMs can be nested, which means that a VM can run a hypervisor on its own and a VM on top of this hypervisor. Nesting can be several layers deep. Networking in such cases is analogous to the one described above.
- vSwitches can support external virtual local area network (VLAN) tagging, or internal (to VMs) VLAN tagging [19]
- vSwitch on VMware Fusion contains a built-in dynamic host control protocol (DHCP) server to assign and distribute IP addresses to VMs.
- vSwitch contains VMkernel ports which serve the VMs when they need to communicate with the hypervisor directly and when the host needs to communicate with the outside world.
- It is possible to set a virtual NIC of a VM to promiscuous mode in which it will be able to receive (sniff) Ethernet traffic destined to other host(s) on the network
- Enterprise solutions may support features such as Cisco Discovery Protocol (CDP), NIC teaming, load balancing and traffic shaping. [19]

Not all of these points are applicable to all hypervisors. Enterprise solutions support more modes of operation and protocols than desktop level hypervisors.

3 Laboratory Preparation

In this section I will describe in detail the setup of both the (virtual) hardware and the software environment of the virtual networking laboratory. The purpose of this section is to guide the reader through the setup process.

The setup of the virtual routers is shown in two versions: using a graphical user interface (GUI) and using a command line interface. Examples of configuration via GUI are given in the form of screenshots; hence the following three sections of the thesis contain numerous figures. To cut down on the number of figures used, navigation in menus and dropdown lists is described by an italicised text path surrounded by question marks as shown in this example: "*Menu* > *Submenu* > *Setting Window*". The same form of visual distinction is used for describing objects in the windows that need to be manipulated in some way. If the same (or similar) configuration is used multiple times throughout the text, the reader will be referred back to the section where such setup was first described.

CLI configuration examples are given in the form of `monospaced text`. User input commands are indicated in **`bold`**, while the output of these commands and other text printed by the routers is of normal weight.

The methodology used in this work (for the configurations and in the laboratory exercises) is as follows:

1. Input the configuration
2. Verify the configuration
3. Test the configuration (with exceptions)

The laboratory exercise chapters of this thesis are meant to emulate those published by the Cisco Networking Academy for use in their CCNA programme. This, coupled with the use of GUI configuration, leads to frequent use of the imperative in those sections. The laboratory exercises can be easily adapted to other network technology courses.

3.1 Host Machine Specifications

The laboratory is running on Apple MacBook Pro (13 inch, mid 2012 model, further in the text "host") with the following specifications:

- 2,5 GHz Intel Core i5 3210M processor
- 16 GB RAM
- 256 GB Samsung 840 Pro SSD
- OSX 10.11 El Capitan.

This configuration is similar in processing power to Apple MacBook Air 2015 model (base configuration) or upgraded Apple MacBook 2015 model (base configuration). The MacBook Pro Retina models released after 2012 are all more powerful than the host used. [20] This host was chosen because it reflects the processing power available to a student who purchased a laptop computer during the years 2014 and 2015.

The installation and configuration steps in this section of the thesis are meant to be referred to in the later stages of work on an as-needed basis and therefore are not necessarily described using concrete values, which depend on a particular setup of each laboratory exercise.

3.2 Virtualisation Software – VMware Fusion 8 Pro

VMware Fusion Pro is a commercial virtualisation software package. I selected it as a virtualisation platform for this thesis for two reasons:

1. At the time of writing it is available free of charge to students in Metropolia's IT degree programmes as a part of VMware academic partnership.
2. VMware products have a significant market share, thus the probability a student will come into contact with them in working life is high.

The software was procured from http://mcp.metropolia.fi. An evaluation copy (which will remain active for a period of 30 days) can be downloaded from https://www.vmware.com/products/fusion/fusion-evaluation.html. In either case the installation of the downloaded .dmg packages is straightforward: doubleclick on the .dmg package to mount it in the “*Finder.app”* and then doubleclick on the application icon to install. When prompted for the license code, either provide a license obtained from http://mcp.metropolia.fi, or select *“Evaluate VMware Fusion 8 Pro”*. While the installer contains both a basic and a Pro version, only the latter includes the Network Manager feature which I used extensively to create the laboratory network environment.

Configuring VMware Fusion 8 Pro Networking

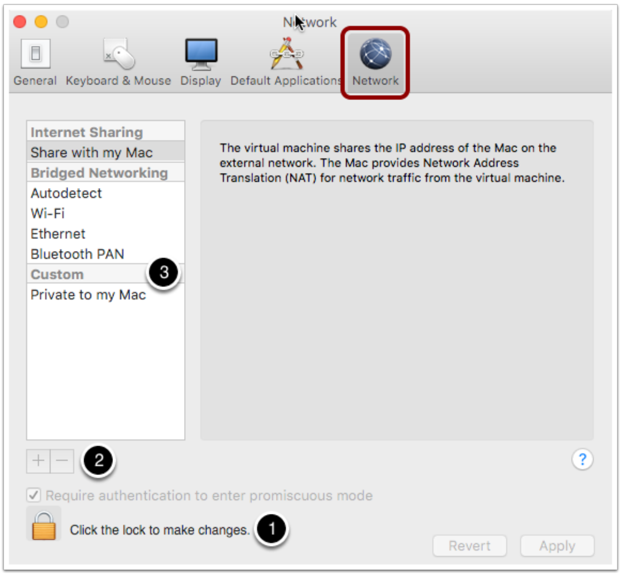
Nodes in the thesis laboratory environment are virtual machines and the links between them are created by virtualisation software. By default, two virtual networks/switches are available to the user: a network which connects to the host’s active Internet connection via NAT (called *“Share with my Mac”* in VMware Fusion 8 Pro preferences) and a host only/private network (called *“Private to my Mac”* in the preferences). For the purpose of building a virtualised networking laboratory multiple networks/switches are needed. New networks can be added and configured in the application preferences. The networking configuration can be found via

- *“VMware Fusion > Preferences… > Network”*
- To add a new network, click on the lock icon and enter the administrator’s password (See figure 2, point 1)
- Then press *"+"* sign to add a new network (point 2)
- In the following window (figure 3) untick *"Connect the host Mac to this network"* (point 1) unless the network is used to simulate the Internet connection or connectivity to the host machine is needed[2].
- Tick *"Provide addresses on this network via DHCP"* (point 2). The addresses for the subnets then can be provided manually. Otherwise they will be provided by a random assignment from a private address space by VMware Fusion.
- Select subnet IP according to the addressing scheme of the given exercise (point 3)
- *"Require authentication to enter promiscuous mode"* should remain ticked.[3]

[2] This is the case when using WinBox software to configure the routers, especially during the initial configuration.

[3] This setting refers to the ability of virtual machines to listen to the network activity of the other nodes on the network (that is to intercept traffic which is not destined to them). It is good practice to limit the ability to snoop the network traffic and allow it only when needed.

- Press *"Apply"* to confirm creation of the new network (point 4).
- The default naming scheme is "*vmnet2*", *"vmnet3"* and so on. The "*vmnet1"* and "*vmnet8"* names are used internally by VMware Fusion.[4]



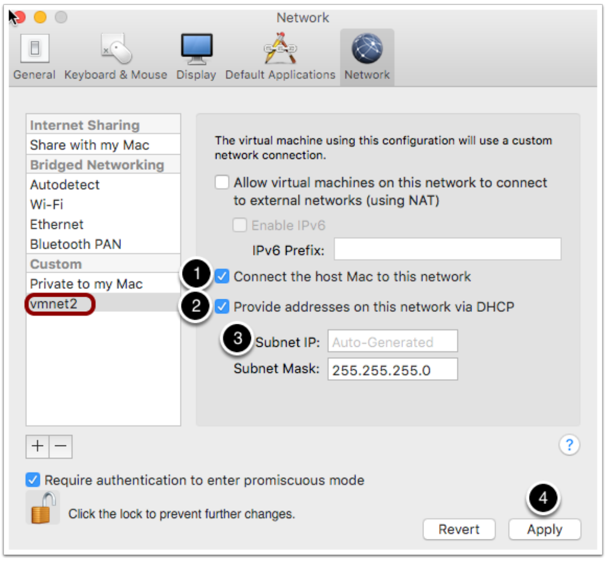
The virtual machine shares the IP address of the Mac on the external network. The Mac provides Network Address Translation (NAT) for network traffic from the virtual machine.

☑ Require authentication to enter promiscuous mode

Click the lock to make changes.

Figure 2. Adding Virtual Networks

- Repeat the process to add networks as required by the network topology/diagram for the laboratory exercise in question.



The virtual machine using this configuration will use a custom network connection.

□ Allow virtual machines on this network to connect to external networks (using NAT)

□ Enable IPv6

IPv6 Prefix:

☑ Connect the host Mac to this network

☑ Provide addresses on this network via DHCP

Subnet IP: Auto-Generated

Subnet Mask: 255.255.255.0

☑ Require authentication to enter promiscuous mode

Click the lock to prevent further changes.

Figure 3. Configuring New Virtual Network

This concludes the network creation phase. The next step is to install and configure RouterOS.

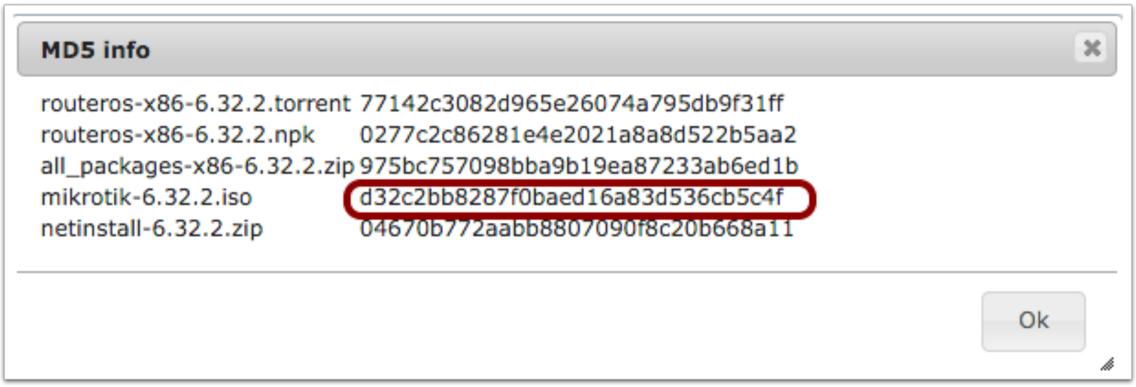
[4] Where "*vmnet1"* is a host only network (called *"Private to my Mac"* in VMware Fusion 8 Pro preferences) and "*vmnet8"* a NAT enabled network (called *"Share with my Mac"* in VMware Fusion 8 Pro preferences)

3.3 Installing and Configuring Mikrotik RouterOS

A demo version of the virtual router software used in this thesis can be downloaded from http://www.mikrotik.com/download. Navigate to *“RouterOS”*, click on *“x86”* and *“ISO Image”*. To verify the integrity of the downloaded ISO image, click on *“MD5”* to reveal the relevant MD5 hash (see figure 4).

Next, compare the hash value obtained from the manufacturer with the file just downloaded:

- Open *“Terminal.app”*
- Run the following command: **`md5 /path/to/the/downloaded.iso`**
- Compare the calculated MD5 hash value with the value provided by the manufacturer. If they match, the downloaded file has not been tampered with during the transit.



routeros-x86-6.32.2.torrent	77142c3082d965e26074a795db9f31ff
routeros-x86-6.32.2.npk	0277c2c86281e4e2021a8a8d522b5aa2
all_packages-x86-6.32.2.zip	975bc757098bba9b19ea87233ab6ed1b
mikrotik-6.32.2.iso	d32c2bb8287f0baed16a83d536cb5c4f
netinstall-6.32.2.zip	04670b772aabb8807090f8c20b668a11

Figure 4. MD5 Hash of RouterOS ISO File

A tampered file means that the file contents have been altered en route to the destination, which can be done purposefully by third parties for purposes such as spreading malware and computer viruses.

Creating RouterOS Virtual Machines

The next step is to create a virtual machine for the RouterOS. The process consists of the following steps:

1. Open VMware Fusion
2. Navigate to “*File > New…”*
3. In *"Select the Installation Method"* window choose *"Create a custom virtual machine"* and press *"Continue"*

4. In *"Choose Operating System"* windows select *"Linux"* and scroll the left side menu down to select *"Other Linux 2.6.x kernel"* and press *"Continue"*
5. In the next window press *"Continue"*
6. The overview window will appear. Since the virtual machine would not boot up with the current settings, press *"Customise Settings"* button.
7. When presented with a file save window, select a fitting name for the virtual machine (Router 1 in this case) and press *"Save".*
8. In the following window click on *"CD/DVD (IDE)"* (see figure 5, point 1). In the window that will pop up click on *"SuperDrive"* and then *"Choose a disc or disc image..."*. If the host machine does not contain SuperDrive press *"Choose a disc or disc image..."* and select the RouterOS image ISO file.
9. Once back to the VM setting window, press *"Hard Disk (SCSI)"* (see figure 5, point 2). There are two important settings to change in the following window: move the *"Disk size:"* slider to the left to shrink the HDD size of the virtual machine to 1GB (see figure 5, point 1). Secondly, press the *"Bus type:"* radio button and select *"IDE".* RouterOS does not contain drivers for *"SCSI"* hard disks and the virtual machine will fail to boot if *"SCSI"* is enabled. To finish setting up the hard disk press *"Apply".*

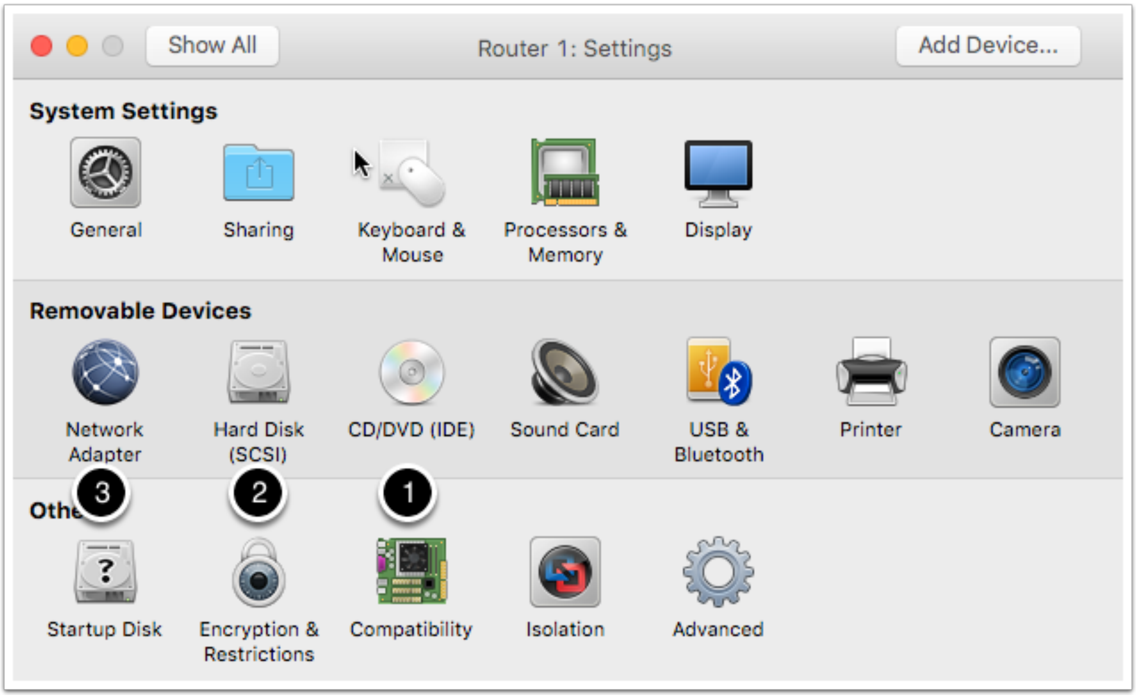
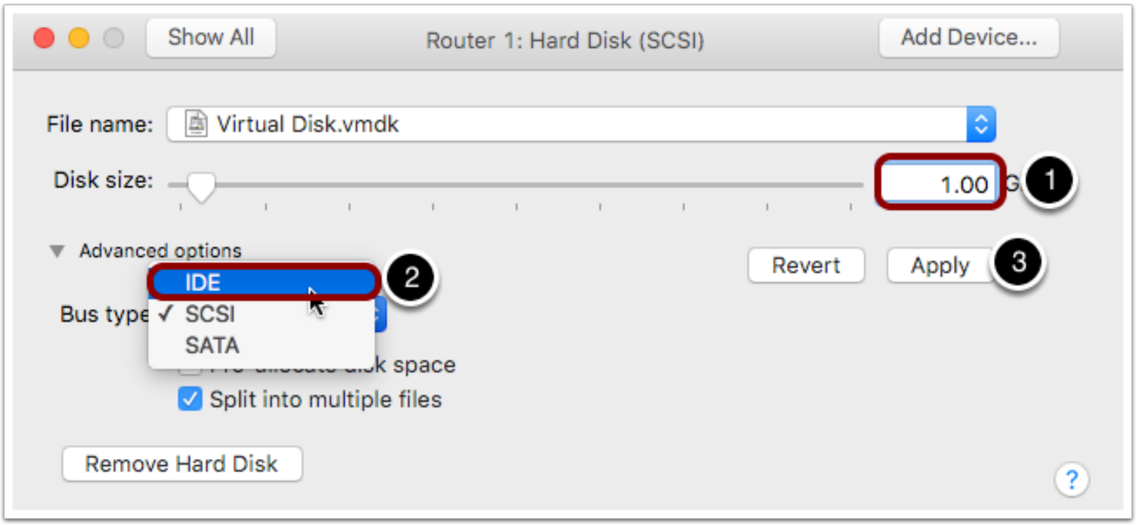


Figure 5. Customising Router OS VM Settings

10. Lastly, the networking configuration needs to be edited. By default, the VM has one virtual NIC available. Real life routers on the other side have multiple NICs

available. To add more NICs press *"Network Adapter"* in the Settings window (See figure 5, point 3). In the following window press *"Add Device…"* in the top right-hand corner. Select *"Network Adapter"* and press *"Add".* Repeat until a sufficient number of network interfaces has been created. New NICs will show up in the Settings window named *"Network Adapter 2"* … *"Network Adapter X"*. Close the Settings window.



Router 1: Hard Disk (SCSI)

File name: Virtual Disk.vmdk

Disk size: 1.00 G

Advanced options

Bus type: ✓ SCSI (IDE, SATA)

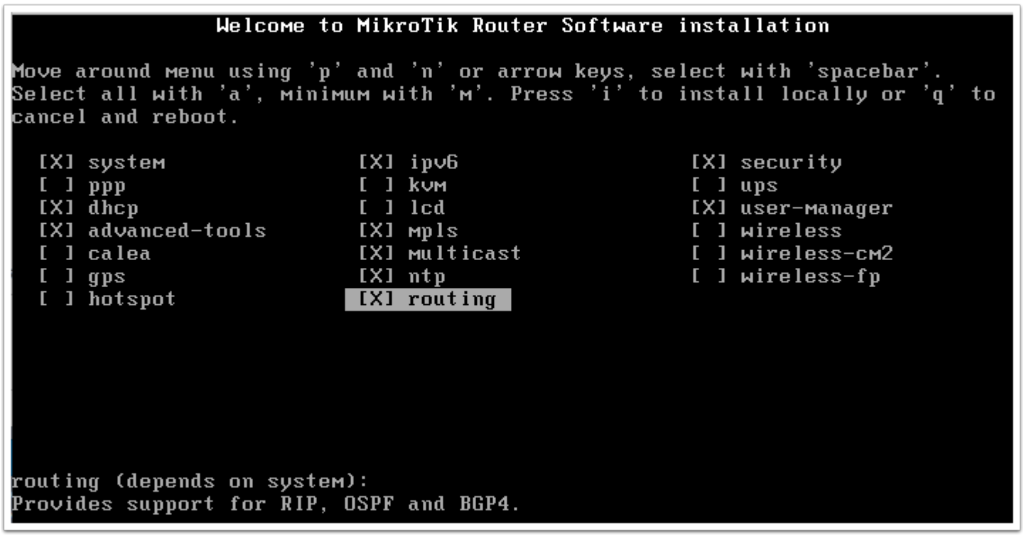
□ Pre-allocate disk space

☑ Split into multiple files

Figure 6. Customising Router OS Hard Disk Settings

The pre-installation configuration is now done and RouterOS can now be installed on the VM.

1. In the *"Virtual Machine Library"* window select the VM just created and press *"Start Up"* button.
2. The VM will boot up and the module installation screen (figure 7) will appear. Select the options according to figure 7. Then press *"i"* to install the system with the selected packages. Confirm the selection by pressing *"y"* twice when prompted.



```
               Welcome to MikroTik Router Software installation

Move around menu using 'p' and 'n' or arrow keys, select with 'spacebar'.
Select all with 'a', minimum with 'm'. Press 'i' to install locally or 'q' to
cancel and reboot.

  [X] system              [X] ipv6                 [X] security
  [ ] ppp                 [ ] kvm                  [ ] ups
  [X] dhcp                [ ] lcd                  [X] user-manager
  [X] advanced-tools      [X] mpls                 [ ] wireless
  [ ] calea               [X] multicast            [ ] wireless-cm2
  [ ] gps                 [X] ntp                  [ ] wireless-fp
  [ ] hotspot             [X] routing




routing (depends on system):
Provides support for RIP, OSPF and BGP4.
```

Figure 7. RouterOS Installation Options

3. After the installation finishes, a prompt to press *"enter"* to restart the VM will appear.
4. The VM will restart and a login screen will appear. The default username and password combination is:

   ```
   Mikrotik Login: admin
   Password: <No password, press "enter" instead>
   ```

5. The option to read the software license will be presented, which can be bypassed by pressing *"n"*. This login screen also shows how much evaluation time the demo version of RouterOS has left. Each user evaluating RouterOS is given 24 hours to do so. Shutting the VM down stops the countdown.
6. The VM can now be shut down. To do so, input the following command in the CLI

   ```
   [admin@Mikrotik] > system shutdown
   ```

   Confirm the command by pressing *"y"*.

3.4 Installing WinBox Software on OSX

The VM that was just created can be accessed and configured in a multitude of ways: by direct connection (serial connection; in this case simulated by VMware Fusion Console window), by SSH (once the IP address of the interfaces has been provided), by WebFig interface (using a built-in WWW server of RouterOS) or by WinBox application for Microsoft Windows provided free of charge by Mikrotik. Each of these methods has distinct advantages and disadvantages. The serial and SSH CLI connections are industry standard. The web interface is often found on consumer grade devices. The configurations in this thesis are done using WinBox software, as this is a feature that sets RouterOS apart from its competitors (they do not offer similarly featured configuration software packages at this price point). Nevertheless, CLI configuration commands are also provided for all of the configurations made.

The WinBox application is only available for Microsoft Windows OS. Unsupported packages that use wrapper software (based on the Wine project) are available. WinBox precompiled for use on OSX can be downloaded from http://www.mikrotik.com/download/share/Winbox.zip. The downloaded zip file can be unzipped by doubleclicking on and dragging and dropping the “WinBox.app” to the *"Applications"* folder on the Mac. The application requires Wine project binaries, which can be downloaded from http://winebottler.kronenberg.org/downloads. Once there, download the development version of the *“WineBottler”* application. Doubleclick on the downloaded .dmg file and drag and drop *"Wine.app"* and *"WineBottler.app"* into the *“Applications”* folder on the Mac. The WinBox application is now ready for use.

3.5 Cloning RouterOS Virtual Machines

Each RouterOS VM represents a single router in the virtual networking laboratory. It is therefore necessary to create additional VMs in order to emulate various networking topologies/scenarios. The simplest way to create more RouterOS VMs is to clone the one just created. To clone the VM do the following:

- In the *"Virtual Machine Library"* window right click on the RouterOS VM and select *"Create Full Clone…"* from the popup menu (see figure 8.).

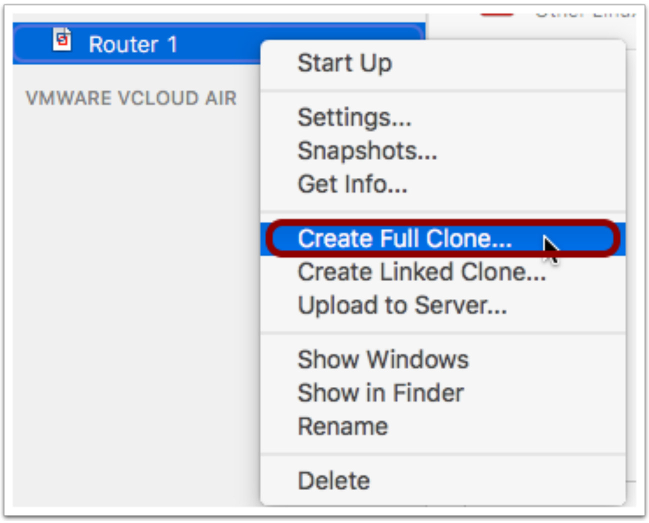
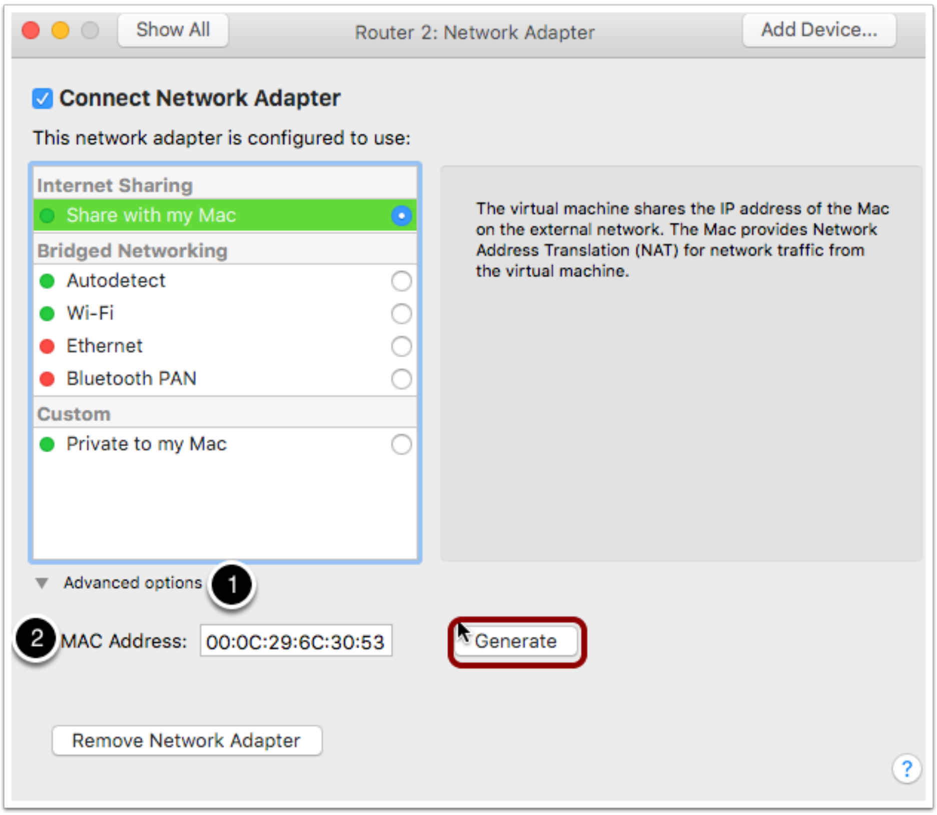


Figure 8. Cloning the RouterOS Virtual Machine

- In the next window will a possibility to name the cloned VM. Give it a descriptive name (such as Router 2 …. Router X) and press *"Save".*
- Create as many clones as required.
- The cloned machines have identical settings and virtual hardware, including MAC addresses of the NICs. If left unchanged this will lead to an inoperable network (since all the machines would have identical unique identifiers on the network). To rectify the situation right click on the newly cloned VM and select *"Settings…"* from the popup menu. Then press *"Network Adapter"*. In the *"Network Adapter"* window press *"Advanced"* (See figure 9, point 1) and then *"Generate"* (figure 9, point 2) to create a new randomised MAC address for the virtual NIC.



☑ Connect Network Adapter

This network adapter is configured to use:

Internet Sharing
- ● Share with my Mac

Bridged Networking
- ○ Autodetect
- ○ Wi-Fi
- ○ Ethernet
- ○ Bluetooth PAN

Custom
- ○ Private to my Mac

The virtual machine shares the IP address of the Mac on the external network. The Mac provides Network Address Translation (NAT) for network traffic from the virtual machine.

▼ Advanced options

MAC Address: 00:0C:29:6C:30:53

Figure 9. Generating New MAC Adresses for Cloned VMs

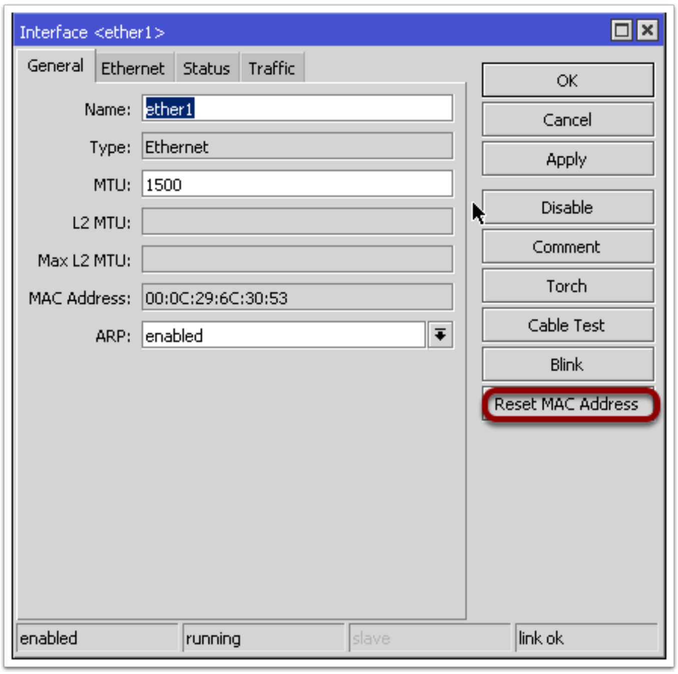
- For the changes of the MAC address to propagate within Router OS settings, start the VM, log in and issue the following command (x is the number of the interface, for example Ethernet1; here in off-by-one form starting with 0):

```
[ admin@MikroTik ] > interface ethernet reset-mac-address x
```

- Verify the updated MAC address by issuing the following command:

```
[ admin@MikroTik ] > interface ethernet print
Flags: X - disabled, R - running, S - slave
#    NAME                MTU  MAC-ADDRESS       ARP
0 R  ether1              1500 00:0C:29:6C:30:53 enabled
```

- Alternatively, the MAC address can be reset by using WinBox. To do so connect to the VM using any of the available interfaces. WinBox offers an option to use the MAC Telnet protocol to perform the configuration, hence allowing remote connection to the machines without Layer 3 connectivity on the directly connected subnets. Then navigate *"WinBox > Interfaces"* and reset the MAC address by clicking on "Reset Mac Address" (refer to figure 10). The change can be verified by inspecting the *"MAC Address"* field to the left of the button just pressed. Note that it is only possible to connect to cloned VMs in their default state one at a time via WinBox, meaning only one running VM at a time. This is due to reliance of WinBox on MAC addresses being unique. Configuration via CLI can be done while all of the cloned VMs are running.



Interface <ether1>

Name: ether1
Type: Ethernet
MTU: 1500
L2 MTU:
Max L2 MTU:
MAC Address: 00:0C:29:6C:30:53
ARP: enabled

enabled | running | slave | link ok

Figure 10. Resetting MAC Address of an Ethernet Interface in WinBox

- Repeat this process for each NIC on each cloned VM.

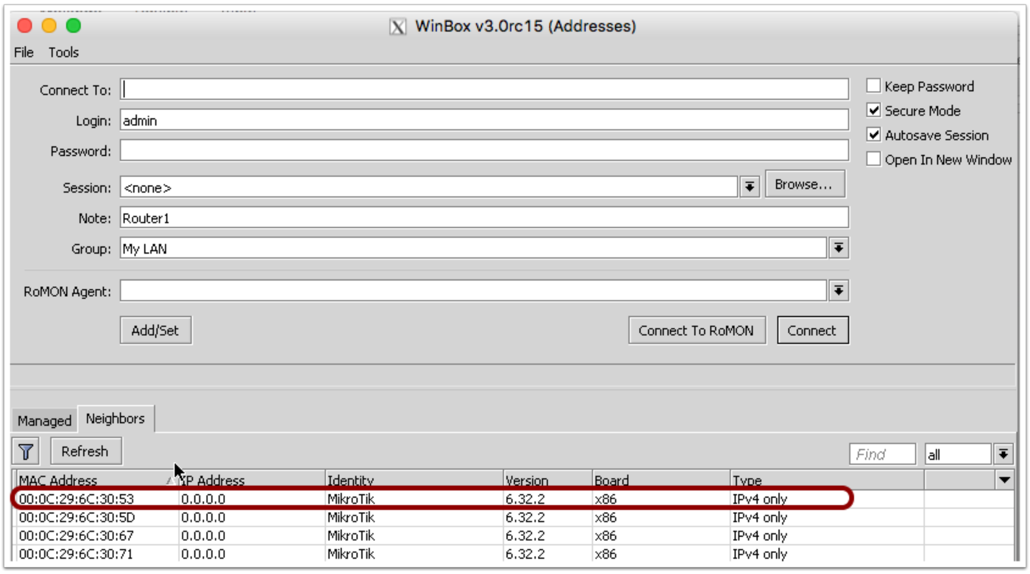
- In case the MAC addresses were not successfully changed, and two or more VMs with identical MAC addresses occur on the same subnet, VMware Fusion will display a warning popup.

3.6 Basic Configuration of RouterOS VMs Using WinBox

Based on the laboratory settings, provide the virtual switches with addresses of the IP subnet. Refer to the "Configuring VMware Fusion 8 Pro Networking" section for more information on the process. To provide for WinBox access from the host machine, the *"Connect the host Mac to this network”* option should be ticked.

While the VMs are powered off, the NICs can be attached to the virtual networks/switches according to network topology diagrams provided for each laboratory. To do so, right click on the VM in the *"Virtual Machine Library"* and select *"Settings…"* from the popup menu. Click on the *"Network Adapter"* icon and in the following window click on the vmnet the machine/NIC combination is supposed to use. Repeat for all the VMs and NICs. Then power up the first VM and start WinBox application.

- Once the VM booted up the MAC addresses of all NICs that are up and running will be displayed in the *"Neighbors"* tab at the bottom of the screen (see figure 11). WinBox can connect to the Router even without layer 3 connectivity. Click on the first line in the *"Neighbors"* tab and press *"Connect"* to log in to the router.



Connect To:
Login: admin
Password:
Session: <none>
Note: Router1
Group: My LAN
RoMON Agent:

□ Keep Password
☑ Secure Mode
☑ Autosave Session
□ Open In New Window

MAC Address	IP Address	Identity	Version	Board	Type
00:0C:29:6C:30:53	0.0.0.0	MikroTik	6.32.2	x86	IPv4 only
00:0C:29:6C:30:5D	0.0.0.0	MikroTik	6.32.2	x86	IPv4 only
00:0C:29:6C:30:67	0.0.0.0	MikroTik	6.32.2	x86	IPv4 only
00:0C:29:6C:30:71	0.0.0.0	MikroTik	6.32.2	x86	IPv4 only

Figure 11. WinBox Default Window

- Dismiss a welcome/license information message by pressing *"OK".*

- Once inside the main WinBox configuration interface for the given router, press *"System"* on the left-hand menu and select *"Identity"* (See figure 12, point 1)

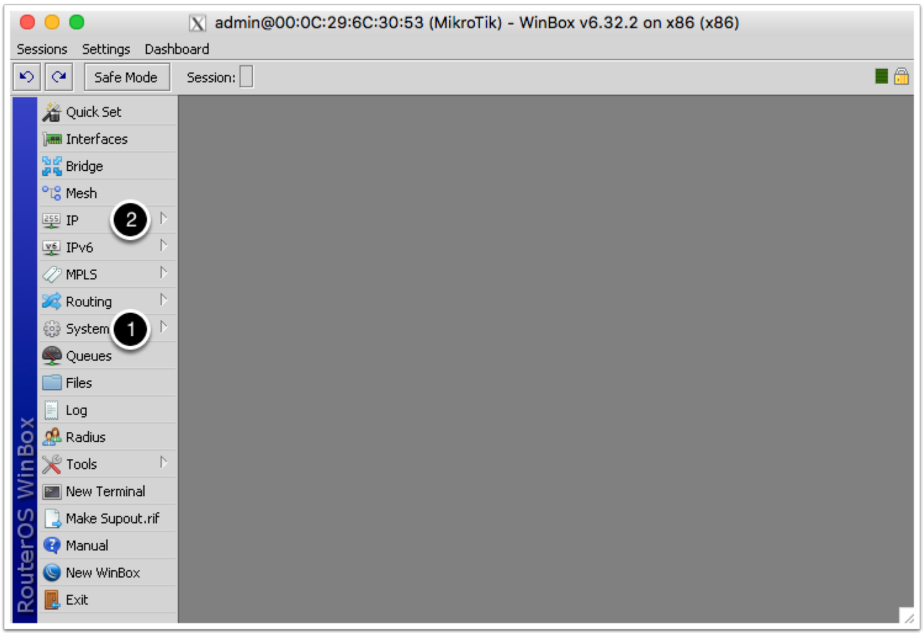
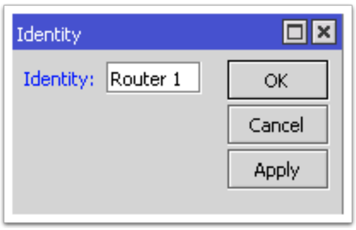


Figure 12. Main Configuration Interface of WinBox

- In the window that will pop up, change the name of the router, so that it reflects the name of the VM (see figure 13). Press *"OK".* The changed name can be verified by revisiting the "Identity" window. The new name will also be visible on top of the WinBox window.



Identity

Identity: Router 1

Figure 13. Setting Router Identity

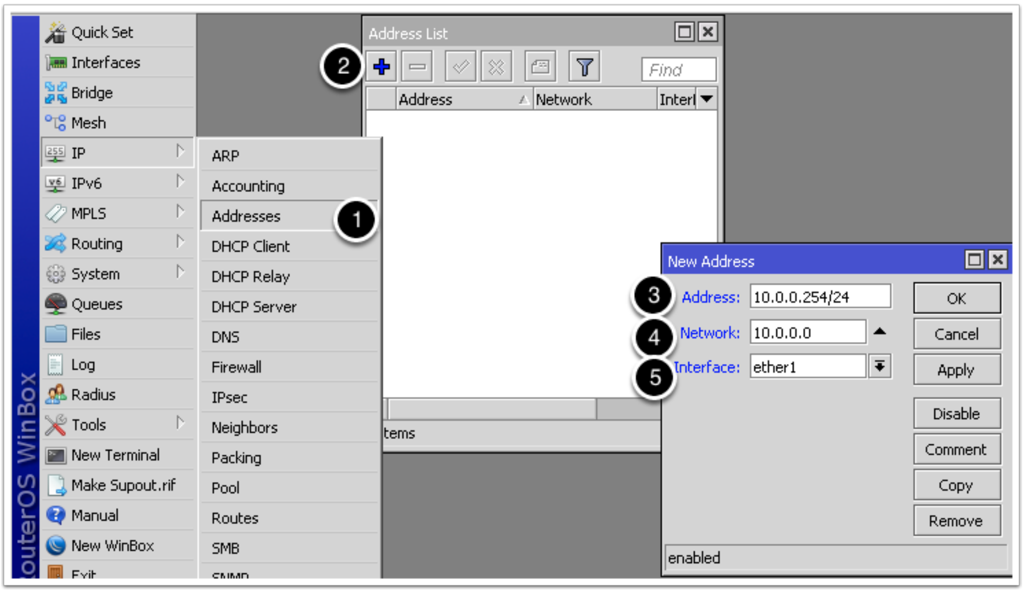
- CLI version of the above configuration is as follows:

```
[ admin@MikroTik ] > system identity set name=”Router x”
```

- Hostname change verification, while the system hostname is displayed as part of the command line, the explicit command to print it is shown below:

```
[ admin@Router x ] > system identity print
name: Router x
```

- After returning to the main configuration window press *"IP"* and then *"Addresses"* (see figure 14, point 1). In the window that pops up press the *"+"* button to add a new IP address to an interface (figure 14, point 2). In the following window provide the interface's IP address in slash notation (point 3), the subnet address (point 4) and select the interface to bear this address (point 5). Press *"Ok"* and repeat for all interfaces as needed.



Address: 10.0.0.254/24
Network: 10.0.0.0
Interface: ether1

Figure 14. Adding IP Addresses to Interfaces

- IP addresses of the interfaces can be set in CLI as follows:

```
[admin@MikroTik] > ip address add address=xxx.xxx.xxx.xx/yy interface=etherx disabled=no
```

- IP addresses of the interfaces can be verified with the following command:

```
[admin@MikroTik] > ip address print
Flags: X - disabled, I - invalid, D - dynamic
 #   ADDRESS            NETWORK         INTERFACE
 0   xxx.xxx.xxx.xxx/yy xxx.xxx.xxx.0      etherx
```

- Other settings which can be set, but are not essential for the purpose of this thesis, are: *"System > Clock"* (to set system time); *"System > NTP Client"* (to keep system time automatically synchronised); *"System > Password"* (to change system password) and *"System > Users"* (to add non-administrative users to the system and/or edit their user rights and configuration views). This completes the basic configuration that is identical for both laboratory exercises.

4 Laboratory Exercise 1

The purpose of this laboratory exercise setup is to emulate a typical SO/HO environment with a small number of subnets and end user devices, where dynamic routing protocols are not necessary due to small size of the network. This laboratory exercise will start with setting up of the virtual topology followed by configuring of static routing, implementating of a backup route, static load balancing and will finish with configuring NAT and testing of the whole setup.

4.1 Laboratory Design and Tasks

A virtualised laboratory network for exercise 1 consists of three router VMs and a Guest PC VM. The Guest PC is also a RouterOS VM with significantly pared down configuration. Internet connectivity is provided by a built-in NAT of vmnet2. Between routers R1 and R2 there are redundant links which will be utilised for a backup route and load balancing properties. Note that the assigned address for the default route of the whole network is not a NAT gateway (as the first address on each vmnet is always the host machine’s address. Instructions on how to reassign this address in VMware Fusion are provided in section 4.6.

The network topology for laboratory exercise 1 is depicted in figure 15 below.

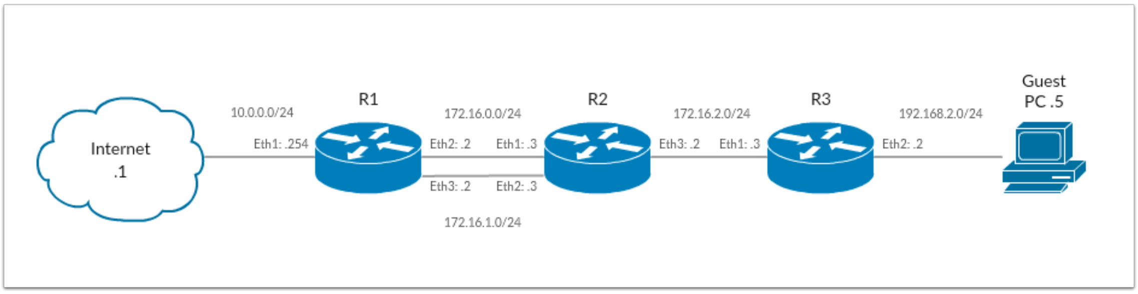


Figure 15. Network Topology for Laboratory 1

The addressing scheme for the hosts forming laboratory 1 topology can be found in table 2.

Table 2. Addressing Scheme for Laboratory 1

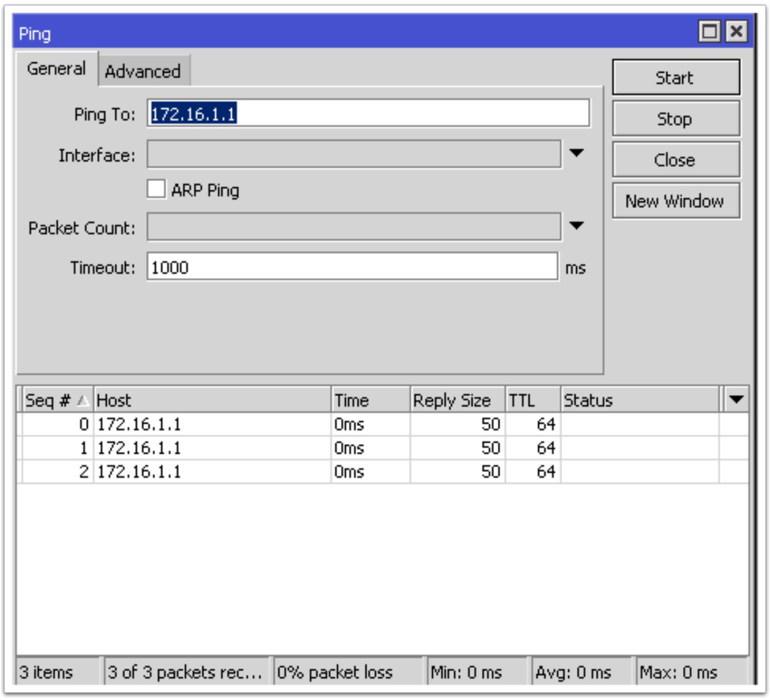
Device	Interface	IP Address	Default Gateway	Port Number	vmnet number
R1	Ethernet 1	10.0.0.254/24	-	1	2
	Ethernet 2	172.16.0.2/24	-	2	3
	Ethernet 3	172.16.1.2/24	-	3	4
R2	Ethernet 1	172.16.0.3/24	-	1	3
	Ethernet 2	172.16.1.3/24	-	2	4
	Ethernet 3	172.16.2.2/24	-	3	5
R3	Ethernet 1	172.16.2.3/24	-	1	5
	Ethernet 2	192.168.2.2/24	-	2	11
Guest PC	Ethernet	192.168.2.5/24	192.168.2.2/24	-	11

Finish the laboratory tasks in the following order:

1. Create the network according to the network topology diagram and assign the correct vmnets to correct interfaces. Then configure RouterOS VMs with appropriate IP addresses.
2. The guest PC is emulated by RouterOS VM with one Ethernet interface.
3. Configure static routing for the network. Configure the default route. Test end-to-end connectivity with ICMP Ping.
4. Configure backup (floating static) route between R1 and R2.
5. Configure the static load balancing between the aforementioned routers.
6. Configure Network Address Translation on eth1 interface of router R1.
7. Test that all of the above works as intended.

4.2 Laboratory Preparation

Network creation and configuration is to be done according to the network addressing scheme and instructions found in sections 2.2 and 2.6. Correctness of the configuration can be tested using ping *("WinBox > Tools > Ping"*, refer to figure 16, alternatively using CLI `[admin@Router x] > ping xxx.xxx.xxx.xxx`) for adjacent interfaces of each respective Router VM. To check whether the VM has connectivity to the host machine ping the first available address in each subnet (for example for subnet 192.168.0.0/24 ping 192.168.0.1). Pinging the host machine might fail if *"Firewall"* and some of its advanced settings have been enabled on the host machine. To make sure that pings will pass through, go to *"System Preferences > Security & Privacy > Firewall* (press the lock icon in the lower left-hand corner and provide the administrator's password*) > Firewall Options"* and in the following window untick *"Enable Stealth Mode"* (see figure 17).



Ping To: 172.16.1.1

Interface:

☐ ARP Ping

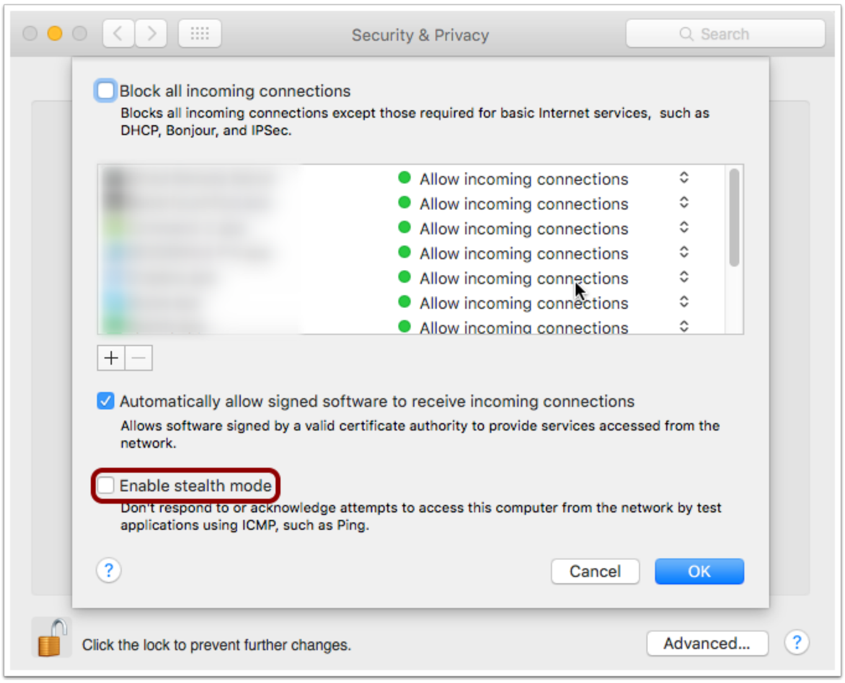
Packet Count:

Timeout: 1000 ms

Seq #	Host	Time	Reply Size	TTL	Status
0	172.16.1.1	0ms	50	64	
1	172.16.1.1	0ms	50	64	
2	172.16.1.1	0ms	50	64	

3 items | 3 of 3 packets rec... | 0% packet loss | Min: 0 ms | Avg: 0 ms | Max: 0 ms

Figure 16. The Ping Tool



□ Block all incoming connections
Blocks all incoming connections except those required for basic Internet services, such as DHCP, Bonjour, and IPSec.

● Allow incoming connections
● Allow incoming connections
● Allow incoming connections
● Allow incoming connections
● Allow incoming connections
● Allow incoming connections
● Allow incoming connections

☑ Automatically allow signed software to receive incoming connections
Allows software signed by a valid certificate authority to provide services accessed from the network.

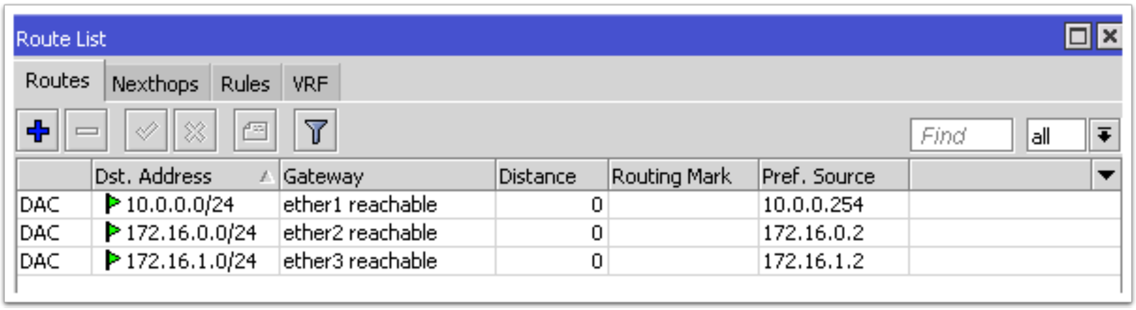
□ Enable stealth mode
Don't respond to or acknowledge attempts to access this computer from the network by test applications using ICMP, such as Ping.

Figure 17. Disabling Stealth Mode of OSX Firewall

Once all of the interfaces on all RouterOS VMs are correctly configured, proceed with the configuration of static routing. The model configurations for all hosts are provided in the appendices.

4.3 Configuring Static Routing on RouterOS

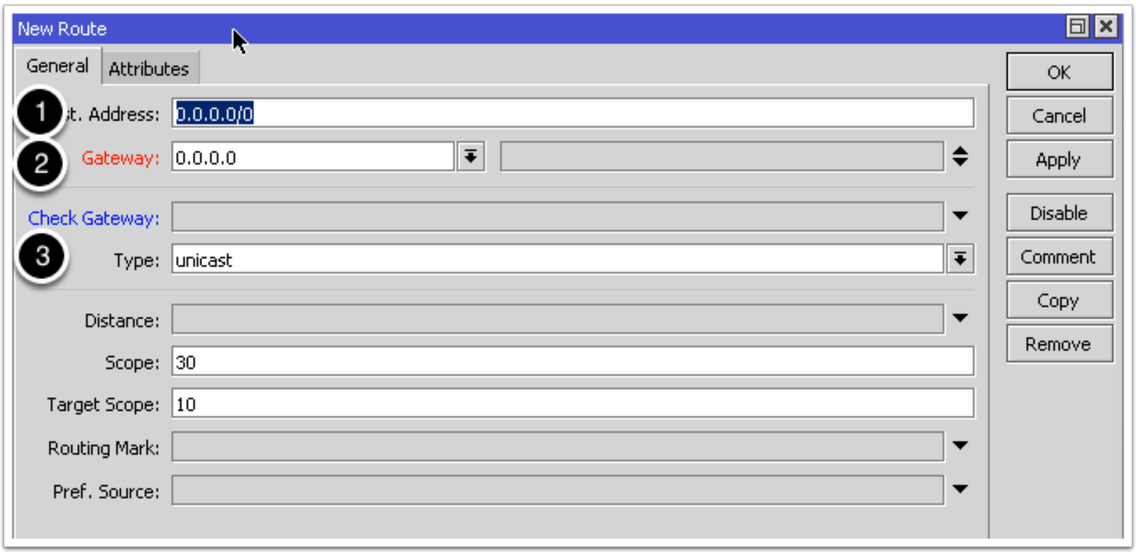
The static routes can be configured in *"WinBox> IP > Routes"*. *"Route List"* window will pop up (see figure 18) and it will be prepopulated with routes to the networks configured on directly attached interfaces.



	Dst. Address	Gateway	Distance	Routing Mark	Pref. Source
DAC	10.0.0.0/24	ether1 reachable	0		10.0.0.254
DAC	172.16.0.0/24	ether2 reachable	0		172.16.0.2
DAC	172.16.1.0/24	ether3 reachable	0		172.16.1.2

Figure 18. Routing Table View

New routes can be added to the routing table by pressing *"+"* button in the top left-hand corner. *"New Route"* window will pop up in which a static route can be configured (see figure 19).



t. Address: 0.0.0.0/0
Gateway: 0.0.0.0
Check Gateway:
Type: unicast
Distance:
Scope: 30
Target Scope: 10
Routing Mark:
Pref. Source:

Figure 19. Creating New Static Routes

At first, provide the network address and mask of the distant network (point 1), then gateway (next hop) address or interface (point 2, interface selection available in the drop-down menu). Lastly it is possible to check whether the gateway is up by executing a ping directly from this window using *"Check Gateway"* function, selecting a ping from the drop-down menu (point 3) and pressing *"Apply"*. Administrative distance of the route can be set in the *"Distance"* field. The new route is confirmed by pressing *"OK".*

Routes can also be created via CLI:

```
[admin@Router 1] > ip route add gateway=172.16.1.2 dst-
address=172.16.2.0/24
```

and verified by displaying the routing table:

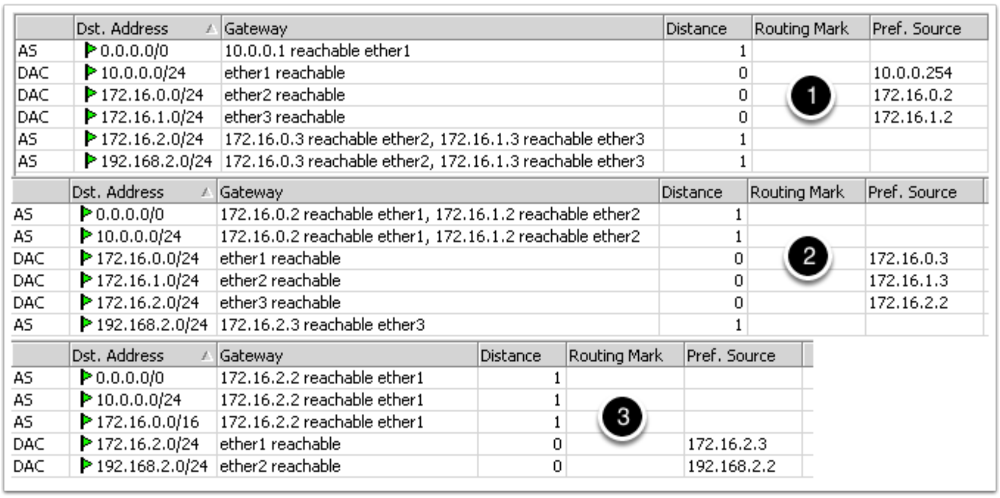
```
[admin@Router 1] > ip route print
```

Default routes are added as follows:

```
[admin@Router 1] > ip route add gateway=10.0.0.1
```

Once the routes are created (see figure 20 for the route tables of all routers; the configuration commands for CLI are included in the appendices), test them with ping. There are some idiosyncrasies of using vmnets: if *"Connect the host Mac to this network"* is ticked during its creation, it will set the host's IP address to the first usable address in the given subnet. A second idiosyncrasy is that these addresses are only reachable from the given vmnet – if pinged from the Guest PC or any routers apart from the routers with the directly connected interfaces on a given vmnet, the ping will fail.

Another problem that can be encountered is creating routes with interfaces as the destination address instead of the next hop address. This setup failed on all RouterOS machines[5]. Only when next the hop address was provided the routing was successful. The routing tables of all the routers are included in appendix 1.2.



	Dst. Address	Gateway	Distance	Routing Mark	Pref. Source
AS	0.0.0.0/0	10.0.0.1 reachable ether1	1		
DAC	10.0.0.0/24	ether1 reachable	0		10.0.0.254
DAC	172.16.0.0/24	ether2 reachable	0		172.16.0.2
DAC	172.16.1.0/24	ether3 reachable	0		172.16.1.2
AS	172.16.2.0/24	172.16.0.3 reachable ether2, 172.16.1.3 reachable ether3	1		
AS	192.168.2.0/24	172.16.0.3 reachable ether2, 172.16.1.3 reachable ether3	1		

	Dst. Address	Gateway	Distance	Routing Mark	Pref. Source
AS	0.0.0.0/0	172.16.0.2 reachable ether1, 172.16.1.2 reachable ether2	1		
AS	10.0.0.0/24	172.16.0.2 reachable ether1, 172.16.1.2 reachable ether2	1		
DAC	172.16.0.0/24	ether1 reachable	0		172.16.0.3
DAC	172.16.1.0/24	ether2 reachable	0		172.16.1.3
DAC	172.16.2.0/24	ether3 reachable	0		172.16.2.2
AS	192.168.2.0/24	172.16.2.3 reachable ether3	1		

	Dst. Address	Gateway	Distance	Routing Mark	Pref. Source
AS	0.0.0.0/0	172.16.2.2 reachable ether1	1		
AS	10.0.0.0/24	172.16.2.2 reachable ether1	1		
AS	172.16.0.0/16	172.16.2.2 reachable ether1	1		
DAC	172.16.2.0/24	ether1 reachable	0		172.16.2.3
DAC	192.168.2.0/24	ether2 reachable	0		192.168.2.2

Figure 20. All Static Routes. Note the load balancing on routers 1 and 2.

Figure 20 shows routing tables after the load balancing has been implemented between routers R1 and R2 (refer to the section 4.5 for more details on its implementation).

4.4 Configuring Backup Routes on RouterOS

The backup routes (in the case of static routes also called "floating static routes") can be created if there are multiple routes to a destination. In this network topology there are two links between routers R1 and R2. Since they are identical when it comes to bandwidth capacity and the number of hops, adding the same static routes for both links would lead to routing loops. If, for example, a new route is added on router R2 to network 10.0.0.0/24, it will be deactivated by default by RouterOS (this can be observed by studying routing tables in appendix 1.2). RouterOS will also automatically activate those routes if the active static routes fail. In other words, RouterOS treats the route that was added second as backup route. This might not be a desired behaviour if multiple backup routes are present and some are preferred to others. To change the backup route's preference, change its distance value. To do so, navigate to *"WinBox > IP > Routes"* press *"+"* button, then fill in the route information according to figure 21.

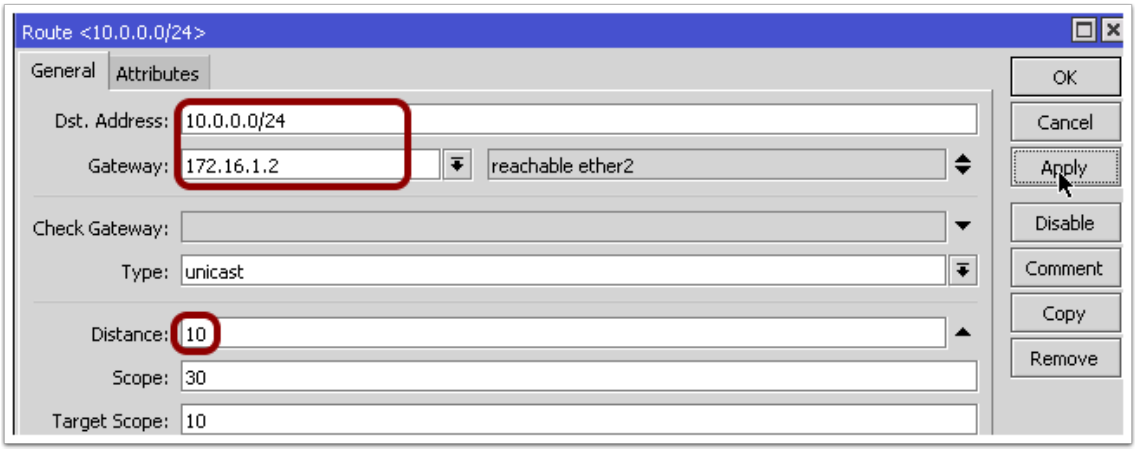
In CLI, edit the existing static routes:

```
[admin@Router 1] > ip route edit number=5 distance
```

[5] Pinging the default gateway of the directly connected subnet/next hop succeeded. However, the pings did not propagate past first hop.

The text editor will open, in which the default value (1) can be changed to "10" in this case. Save the change by pressing *"Ctrl-o"*. Route's new administrative distance value can be verified by:

```
[admin@Router 1] > ip route print
```



Dst. Address: 10.0.0.0/24

Gateway: 172.16.1.2 reachable ether2

Check Gateway:

Type: unicast

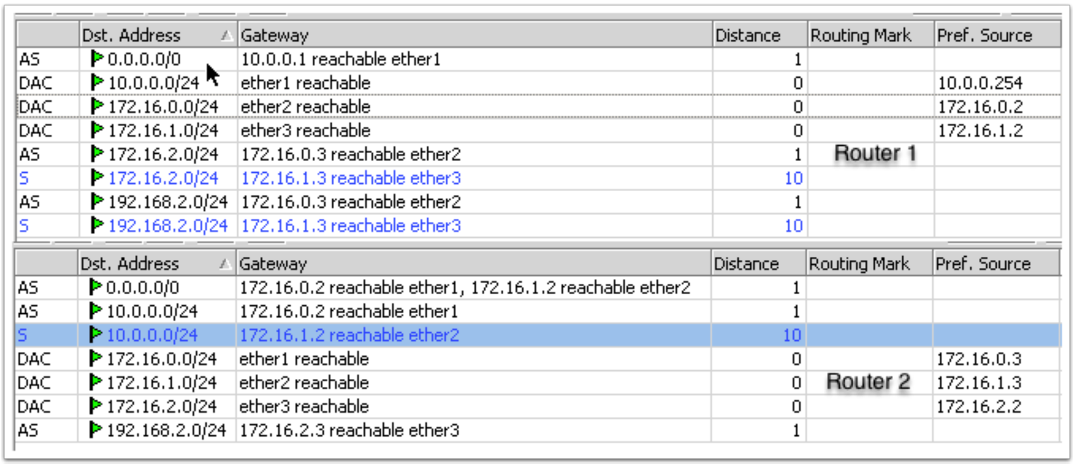
Distance: 10

Scope: 30

Target Scope: 10

Figure 21. Changing Route's Administrative Distance

Once the routes are set on both routers the configuration should look as in figure 22:



	Dst. Address	Gateway	Distance	Routing Mark	Pref. Source
AS	0.0.0.0/0	10.0.0.1 reachable ether1	1		
DAC	10.0.0.0/24	ether1 reachable	0		10.0.0.254
DAC	172.16.0.0/24	ether2 reachable	0		172.16.0.2
DAC	172.16.1.0/24	ether3 reachable	0		172.16.1.2
AS	172.16.2.0/24	172.16.0.3 reachable ether2	1		
S	172.16.2.0/24	172.16.1.3 reachable ether3	10		
AS	192.168.2.0/24	172.16.0.3 reachable ether2	1		
S	192.168.2.0/24	172.16.1.3 reachable ether3	10		

	Dst. Address	Gateway	Distance	Routing Mark	Pref. Source
AS	0.0.0.0/0	172.16.0.2 reachable ether1, 172.16.1.2 reachable ether2	1		
AS	10.0.0.0/24	172.16.0.2 reachable ether1	1		
S	10.0.0.0/24	172.16.1.2 reachable ether2	10		
DAC	172.16.0.0/24	ether1 reachable	0		172.16.0.3
DAC	172.16.1.0/24	ether2 reachable	0		172.16.1.3
DAC	172.16.2.0/24	ether3 reachable	0		172.16.2.2
AS	192.168.2.0/24	172.16.2.3 reachable ether3	1		

Figure 22. Configuration of Backup Routes

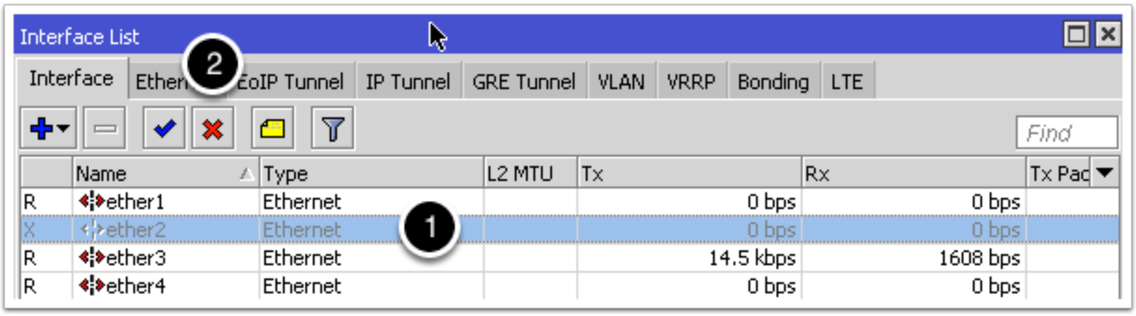
The backup routes are indicated by a higher administrative distance and the fact that they are not in use (letter A is missing in their status window). To test the backup route, disable the Ethernet 2 interface on router R1 by following *"WinBox > Interfaces"*, then select Ethernet 2 and disable it by clicking on the *"x"* button (see figure 23 points 1 and 2).

To disable an interface in CLI, use the following command:

```
[admin@Router 1] > interface ethernet disable ether2
```

and verify it by typing the following:

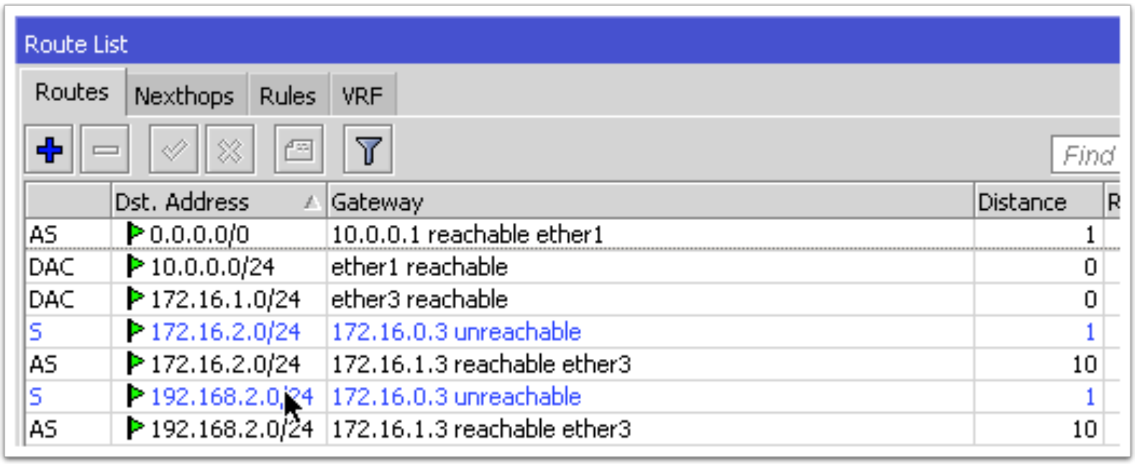
```
[admin@Router 1] > interface ethernet print
```



	Name	Type	L2 MTU	Tx	Rx	Tx Pac
R	ether1	Ethernet		0 bps	0 bps	
X	ether2	Ethernet		0 bps	0 bps	
R	ether3	Ethernet		14.5 kbps	1608 bps	
R	ether4	Ethernet		0 bps	0 bps	

Figure 23. Disabling Network Interface

After a while the change will propagate to the routing table (see figure 24):



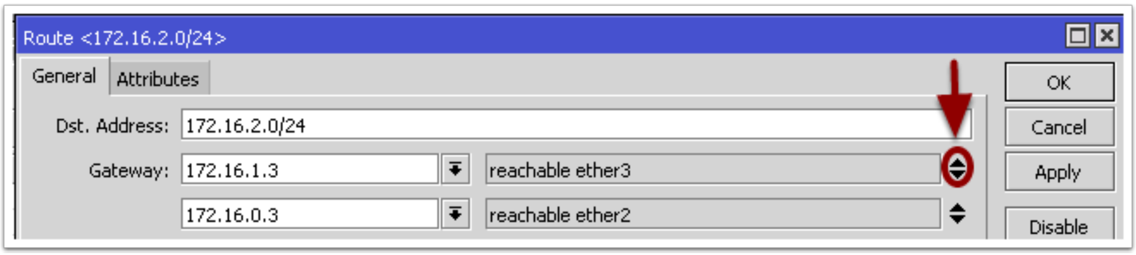
	Dst. Address	Gateway	Distance	R
AS	0.0.0.0/0	10.0.0.1 reachable ether1	1	
DAC	10.0.0.0/24	ether1 reachable	0	
DAC	172.16.1.0/24	ether3 reachable	0	
S	172.16.2.0/24	172.16.0.3 unreachable	1	
AS	172.16.2.0/24	172.16.1.3 reachable ether3	10	
S	192.168.2.0/24	172.16.0.3 unreachable	1	
AS	192.168.2.0/24	172.16.1.3 reachable ether3	10	

Figure 24. Routing Table with Active Backup Routes

The original route is inactive and the backup route took over. Its functionality can be proven by pinging router R3 from router R1 and vice versa.

4.5 Configuring Manual Load Balancing on RouterOS

It is possible to have both of the routes from the previous example active at the same time. This usually indicates a form of load balancing is set up between the links. RouterOS supports various methods of load balancing. The next example deals with configuration of Equal Cost Multi Path (ECMP) routing on RouterOS. The configuration is shown in figure 25.



Dst. Address: 172.16.2.0/24

Gateway: 172.16.1.3 reachable ether3

172.16.0.3 reachable ether2

Figure 25. Adding Multiple Next Hops to a Static Route

After reactivating of interface Ethernet 2 on router R1 and deletion of backup routes, add the additional next hop to the remaining route by clicking on a small triangle button to the right of the first *"Gateway"* field. Input the address of the other hop on for all the routes on both routers (the correct configuration can be seen in figure 20). Note that with this setup, the load on the link will be balanced 50:50. It is possible to add identical next hop multiple times which will lead to more traffic being sent down that particular link (in the case of figure 25, if one more "Gateway" line with the next hop address of 172.16.0.3 was added, the final traffic distribution would be 1:2).

The CLI commands to configure load balancing are as follows:
Use following command to delete a route:

```
            [admin@Router 1] > ip route remove numbers=5,7
```

Route numbers can be obtained from the routing table. Display it with the command:

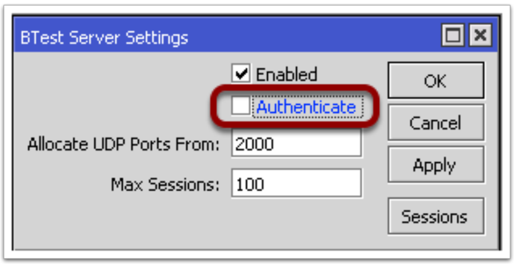
```
            [admin@Router 1] > ip route print
```

Edit remaining routes for ECMP as follows:

```
            [admin@Router 2] > ip route edit number=0 gateway
```

In the editor that opens up, add a comma (",") followed by the IP address of an additional hop. Verify by viewing the routing table. Configure ECMP on all routes between R1 and R2 (example configuration in appendix 1.3).

The operation of ECMP routing can be verified by a built-in bandwidth tester tool. On Router 1 navigate *"WinBox > Tools > BTest Server",* untick the authentication requirement (see figure 26) and press *"Apply".*



BTest Server Settings

☑ Enabled

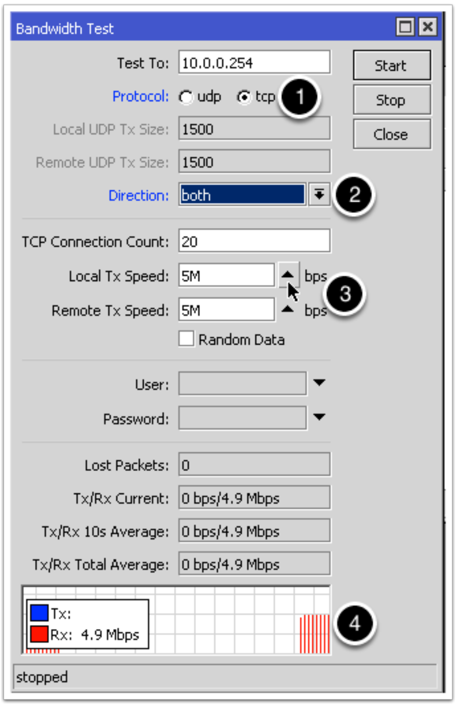
□ Authenticate

Allocate UDP Ports From: 2000

Max Sessions: 100

Figure 26. Setting up Bandwidth Test Server

On Router 2 VM navigate *"WinBox > Tools > Bandwidth Test"* and configure the test according to figure 27.



Test To: 10.0.0.254
Protocol: ○ udp ● tcp
Local UDP Tx Size: 1500
Remote UDP Tx Size: 1500
Direction: both
TCP Connection Count: 20
Local Tx Speed: 5M bps
Remote Tx Speed: 5M bps
□ Random Data
User:
Password:
Lost Packets: 0
Tx/Rx Current: 0 bps/4.9 Mbps
Tx/Rx 10s Average: 0 bps/4.9 Mbps
Tx/Rx Total Average: 0 bps/4.9 Mbps
🟦 Tx:
🟥 Rx: 4.9 Mbps
stopped

Figure 27. Setting up Bandwidth Test

Note that TCP testing is used (point 1), throughput in both directions is tested (point 2), and to conserve resources on the host machine the maximum bandwidth used is 5 Mbps in both directions (point 3). The bandwidth tester operation can be verified by a progress graph at the bottom of the window (point 4). The test is started by pressing *"Start"*.

Now it is possible to display which interfaces are utilised on R1. Navigate *"WinBox > Interfaces"* and in the first window doubleclick on interfaces *“ether 2”* and *“ether 3”.* In the windows that pop up, click on the *"Traffic"* tab. The traffic is now routed through both interfaces (see figure 28). The traffic is being received by interface *“ether 2”* and transmitted by interface *“ether 3”*.

CLI configuration of bandwidth server is as follows:

```
[admin@Router 1] > tool bandwidth-server set authenticate=no
```

Bandwidth test can be configured with the following command:

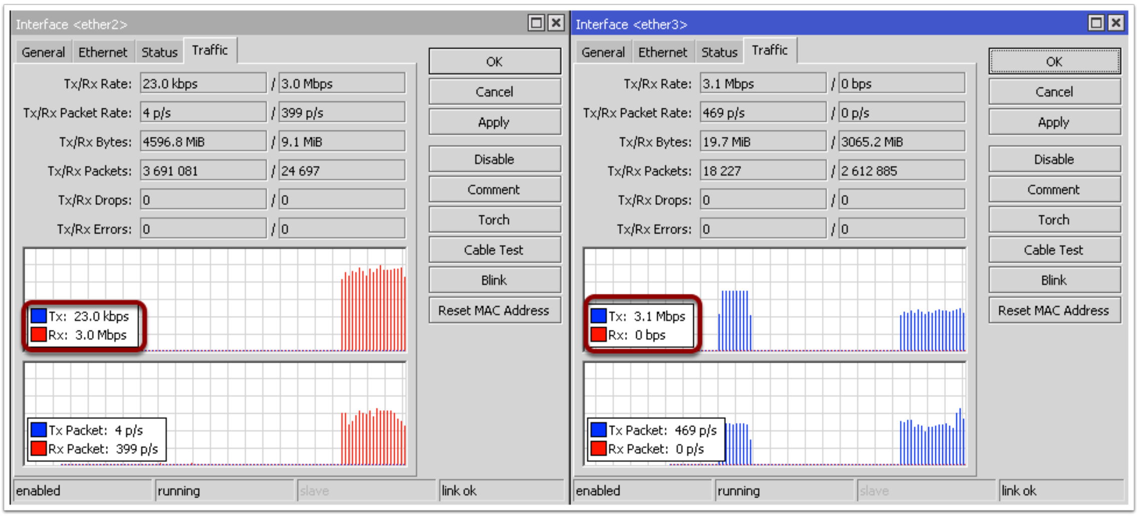
```
[admin@Router 2] > tool bandwidth-test direction=both protocol=tcp local-tx-speed=5M remote-tx-speed=5M address=10.0.0.254
```

The above commands can be verified in Router 2 as follows:

```
[admin@Router 1] > tool torch ether2
```

and

```
[admin@Router 1] > tool torch ether3
```



Tx/Rx Rate: 23.0 kbps / 3.0 Mbps
Tx/Rx Packet Rate: 4 p/s / 399 p/s
Tx/Rx Bytes: 4596.8 MiB / 9.1 MiB
Tx/Rx Packets: 3 691 081 / 24 697
Tx/Rx Drops: 0 / 0
Tx/Rx Errors: 0 / 0

🟦 Tx: 23.0 kbps
🟥 Rx: 3.0 Mbps

🟦 Tx Packet: 4 p/s
🟥 Rx Packet: 399 p/s

enabled | running | slave | link ok

Tx/Rx Rate: 3.1 Mbps / 0 bps
Tx/Rx Packet Rate: 469 p/s / 0 p/s
Tx/Rx Bytes: 19.7 MiB / 3065.2 MiB
Tx/Rx Packets: 18 227 / 2 612 885
Tx/Rx Drops: 0 / 0
Tx/Rx Errors: 0 / 0

🟦 Tx: 3.1 Mbps
🟥 Rx: 0 bps

🟦 Tx Packet: 469 p/s
🟥 Rx Packet: 0 p/s

enabled | running | slave | link ok

Figure 28. Verifying ECMP Routing Functionality

Figure 28 clearly shows that the bandwidth tester traffic circulates between routers R1 and R2 using both links on which load balancing was configured.

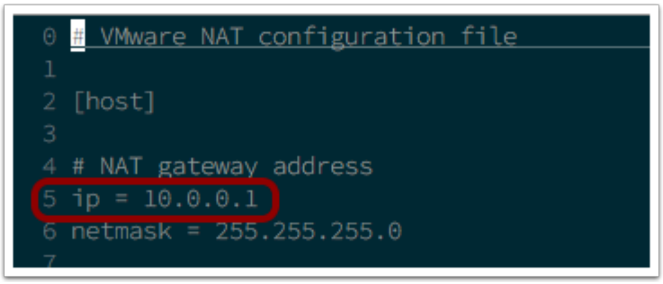
4.6 Configuring NAT on RouterOS

In this part of the laboratory exercise NAT filtering will be added to the *"ether1"* interface of router 1. The vmnet 2 network will be edited in a way that will allow its nodes to access the Internet. To do so, go to *"VMware Fusion > Preferences > Network"*, select vmnet 2 and change the following settings: tick *"Allow virtual machines on this network to connect to external networks (using NAT)"* and untick *"Connect the host Mac to this network"* (refer to figure 3 for more details).

In order for routing outside the network to work, a change in the vmnet 2 configuration file needs to be made. The address that is used by the default route to point to the Internet is the address of the host Mac, which does not route the traffic further. To change default gateway address of vmnet's own NAT run the following command in the *"Terminal.app"*:

```
sudo vim /Library/Preferences/VMware\ Fusion/vmnet2/nat.conf
```

Then change the IP address of the NAT gateway on the fifth line from 10.0.0.2 to 10.0.0.1 (see figure 29).



```
# VMware NAT configuration file

[host]

# NAT gateway address
ip = 10.0.0.1
netmask = 255.255.255.0
```

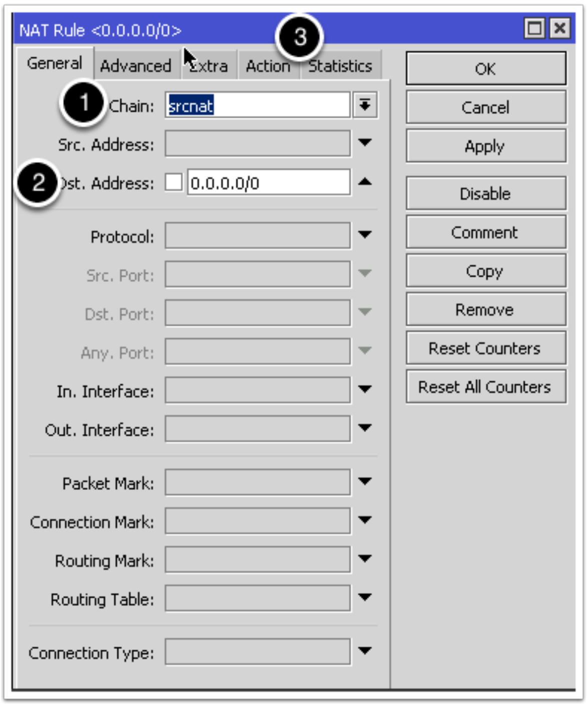
Figure 29. Changing IP Address of vmnet NAT Default Gateway

Once the address is changed, shut down all virtual machines and restart VMware Fusion. After starting the virtual machines, the updated IP address will be in use. Pinging Google's DNS server (IP address 8.8.8.8) is now possible from R1, but not from other nodes in the network (VMware Fusion only allows nodes on vmnet 2 to access it). To enable other nodes to access the Internet, NAT needs to be enabled at the Ethernet 1 interface of router R1.

To set up NAT on R1, go to *"WinBox > IP > Firewall"* and select the *"NAT"* tab in the window that pops up. Press *"+"* button. *"NAT Rules"* window shows up and the correct NAT rule chain will be preselected (translate the source IP addresses, see figure 30, point 1). Change the destination address to all addresses (0.0.0.0/0) and move to the *"Action"* tab (point 3). In *"Action"* drop-down menu select *"Masquerade",* press *"OK".* NAT configuration is now done, test by pinging Google DNS from the Guest PC VM (see figure 31).

NAT can be configured with the following command:

```
[admin@Router 1] > ip firewall nat add action=masquerade chain=srcnat
```



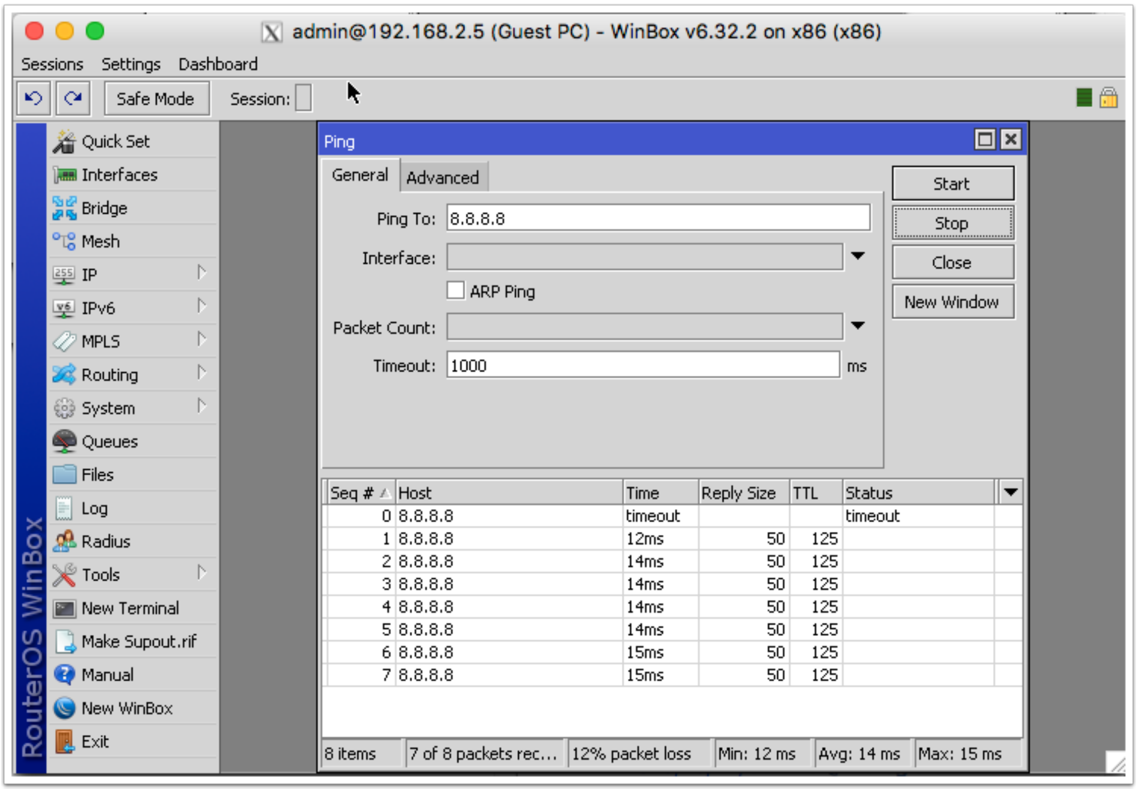
NAT Rule <0.0.0.0/0>

Chain: srcnat

Dst. Address: □ 0.0.0.0/0

Figure 30. Creating NAT Rules

Note that while the ping will work, the traceroute out of the VMware Fusion environment will fail.



admin@192.168.2.5 (Guest PC) - WinBox v6.32.2 on x86 (x86)

Ping

Ping To: 8.8.8.8

Interface:

□ ARP Ping

Packet Count:

Timeout: 1000 ms

Seq #	Host	Time	Reply Size	TTL	Status
0	8.8.8.8	timeout			timeout
1	8.8.8.8	12ms	50	125	
2	8.8.8.8	14ms	50	125	
3	8.8.8.8	14ms	50	125	
4	8.8.8.8	14ms	50	125	
5	8.8.8.8	14ms	50	125	
6	8.8.8.8	15ms	50	125	
7	8.8.8.8	15ms	50	125	

8 items | 7 of 8 packets rec… | 12% packet loss | Min: 12 ms | Avg: 14 ms | Max: 15 ms

Figure 31. Proof of NAT Operation

5 Laboratory Exercise 2

The second laboratory exercise emulates a large network that has been split into smaller areas to lessen the load link state database recalculations inflict on routers. During this laboratory exercise the virtual topology will be set up, OSPF will be configured on all routers taking part in it following with a configuration of a passive OSPF interface and will be wrapped up by creating a GRE tunnel and testing the whole setup. The instructions also cover creation of loopback interfaces as this is done in a significantly different manner on RouterOS in comparison with Cisco IOS.

5.1 Laboratory Design and Tasks

Laboratory network comprises four RouterOS VMs in the role of a router and one as a Guest PC (for testing of the GRE tunnel operation). Links between the routers are formed by different vmnets. This laboratory exercise does not require Internet connectivity. Note that with the exception of router R1 all other routers also belong to non-backbone areas. Guest PC is omitted from the topology diagram for the sake of clarity. Nevertheless, it is one of the hosts forming area 4 network.

The network topology of laboratory exercise 2 network can be seen in figure 32.

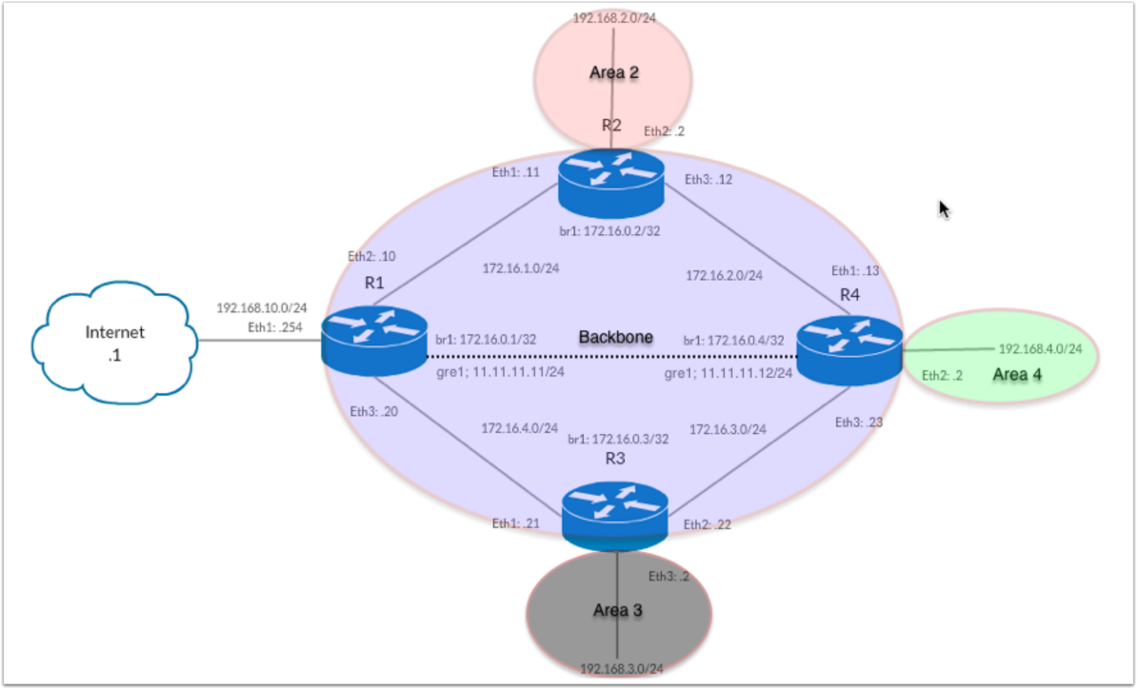


Figure 32. Laboratory 2 Network Topology

The addressing scheme for the laboratory can be found in table 3.

Table 3. Addressing Scheme of Laboratory 2

Device	Interface	IP Address	Area	vmnet
R1	br1 (loopback)	172.16.0.1/32	Backbone	-
	eth1	192.168.10.0/24	-	2
	eth2	172.16.1.10/24	Backbone	3
	eth3	172.16.4.20/24	Backbone	6
	gre-tunnel1	11.11.11.11/24		
R2	br1 (loopback)	172.16.0.2/32	Backbone	-
	eth1	172.16.1.11/24	Backbone	3
	eth2	192.168.2.2/24	Area 2	4
	eth3	172.16.2.12/24	Backbone	5
R3	br1 (loopback)	172.16.0.3/32	Backbone	-
	eth1	172.16.4.21/24	Backbone	6
	eth2	172.16.3.22/24	Backbone	7
	eth3	192.168.3.2/24	Area 3	9
R4	br1 (loopback)	172.16.0.4/32	Backbone	-
	eth1	172.16.2.13/24	Backbone	5
	eth2	192.168.4.2/24	Area 4	10
	eth3	172.16.3.23/24	Backbone	9
	gre-tunnel1	11.11.11.12/24		
Guest PC	eth1	192.168.4.5/24	Area4	10

The laboratory tasks are as follows:

1. Create the network according to the diagram and assign the correct vmnets to the correct interfaces. Then configure RouterOS VMs with correct IP addresses.
2. Configure dynamic routing for the network using OSPF. Create and redistribute a default route. Test end-to-end connectivity with ICMP ping.
3. Configure a GRE tunnel between ether1 interface on R4 and ether2 interface on R1.
4. Test that all of the above works as intended.

5.2 Laboratory Preparation

Network creation and configuration is to be done according to the network addressing scheme and instructions found in sections 3.2 and 3.6. Correctness of the configuration can be tested using ping *("WinBox > Tools > Ping"*, refer to figure 16) for adjacent interfaces of each respective router VM. For the purposes of the OSPF election process it is beneficial to have a permanent router ID. By default, OSPF selects the interface with the highest numerical IP address as its ID. Regular interfaces bearing IP addresses might fail and then new designated router (DR) and backup designated router (BDR) elections take place. This can be prevented if there is a loopback interface on the router bearing the desired IP address. To add a loopback interface, create a new bridge via *"WinBox > Bridge"*, then, in the next window press *"add"*, rename the bridge to *"br1"* and press *"OK"* to save the changes. IP address can be assigned to the newly created bridge interface via *"WinBox > IP > Addresses"*. Detailed instructions are in section 3.6. Create and configure loopback interfaces for all four routers. CLI configuration is as follows:

```
          [admin@Router 1] > interface bridge add name=br1
          [admin@Router 1] > ip address add ad-
dress=172.16.0.1/24 interface=br1
```

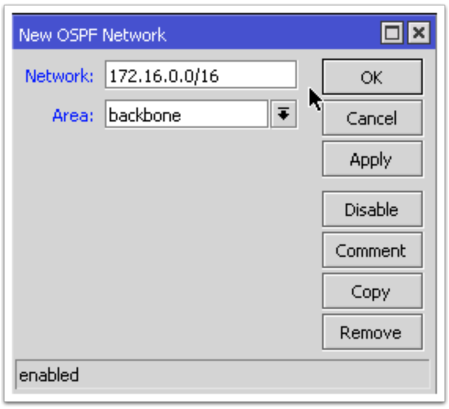
Verify the configuration with the following command:

```
          [admin@Router 1] > interface bridge print
          [admin@Router 1] > ip address print
```

5.3 Configuring Dynamic Routing on Router OS

The advantage of dynamic routing is that the network reacts to the changes in topology autonomously without a need for the administrator's interaction, once configured. A downside is that dynamic routing protocols use up computing resources, memory and bandwidth of the routers and data links.

To start setting up dynamic routing using OSPF, navigate *"WinBox > Routing > OSPF"* and then select *"Instance"* tab. Once there doubleclick on the *"default instance"* and provide IP address of the br1 interface as Router ID (see figure 34, point 1). Select to always redistribute default route on router R1 (as type 1, refer to figure 34, point 2).



New OSPF Network

Network: 172.16.0.0/16

Area: backbone

enabled

Figure 33. Adding Networks to OSPF Process

Next click on *"Networks"* tab and *"+"* button to add a new network to OSPF process. It is possible to use summary routes here. Settings for router R1 are presented in figure 33. Once the network is added to the OSPF process, the interfaces will be added to the process automatically.

CLI version of OSPF instance setup is as follows:

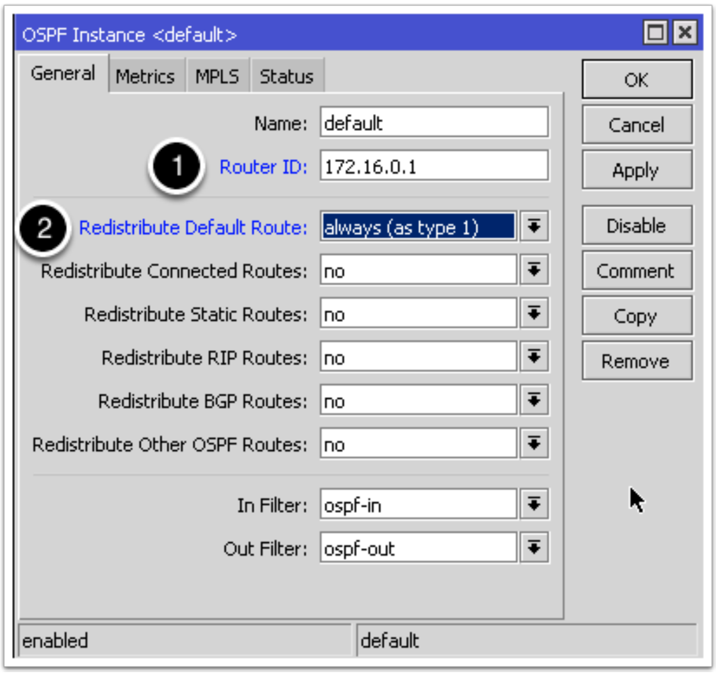
```
[admin@Router 1] > routing ospf instance set [ find
default=yes ] distribute-default=always-as-type-1
router-id=172.16.0.1
```

Adding network to OSPF process can be accomplished with the following command:

```
[admin@Router 1] > routing ospf network add area=back-
bone network=172.16.0.0/16
```

Verify the configuration with the following command:

```
[admin@Router 1] > routing ospf instance print
[admin@Router 1] > routing ospf network print
```



Name: default

Router ID: 172.16.0.1

Redistribute Default Route: always (as type 1)

Redistribute Connected Routes: no

Redistribute Static Routes: no

Redistribute RIP Routes: no

Redistribute BGP Routes: no

Redistribute Other OSPF Routes: no

In Filter: ospf-in

Out Filter: ospf-out

enabled | default

Figure 34. Setting OSPF Router ID

To create a new OSPF area press *"Areas"* tab. Press *"+"* button and in the *"New OSPF Area"* window rename the new area according to the IP addressing scheme (see figure 35, point 1). Set *"Area ID"* last digit to the number of the area according to the IP addressing scheme (point 2). The Mikrotik Area ID differs from Cisco's implementation by having form of dotted decimal (like IPv4 addresses). Cisco uses plain decimals. Add the networks (and interfaces) to the freshly created area by going to the *"Networks"* tab and repeating the steps according to figure 32 for the networks in this area. Create areas on the routers that are connected to them and add networks to the OSPF process on all routers. Verify the functionality of routing by pinging all network interfaces on all routers.

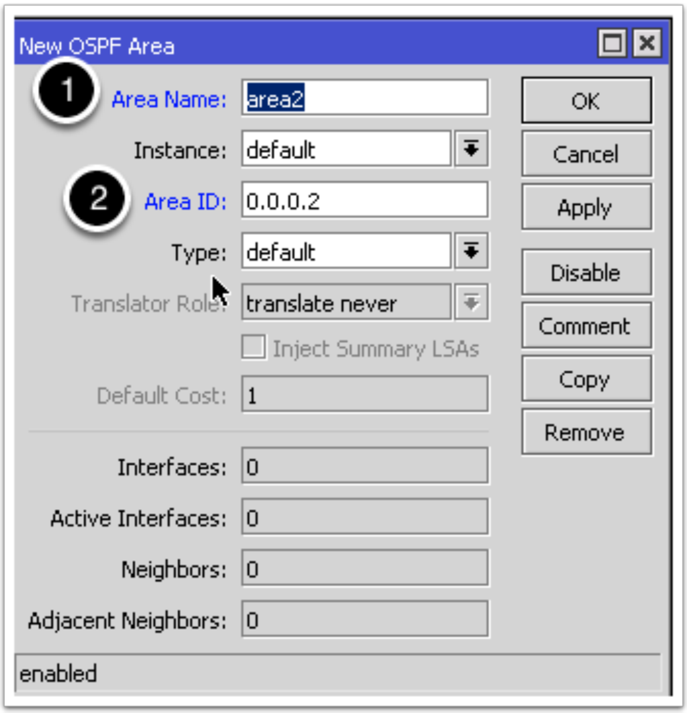
In the *"New OSPF Area"* window it is also possible to create Stubby and Not So Stubby areas by choosing those area types from the *"Type"* drop-down menu. Mikrotik RouterOS does not offer support for Cisco's proprietary Totally Stubby and Totally Not So Stubby areas.

CLI setup of new OSPF areas is as follows:

```
[admin@Router 2] > routing ospf area add area-
id=0.0.0.2 name=area2
```

The configuration can be verified with the following command:

```
[admin@Router 2] > routing ospf area print
```



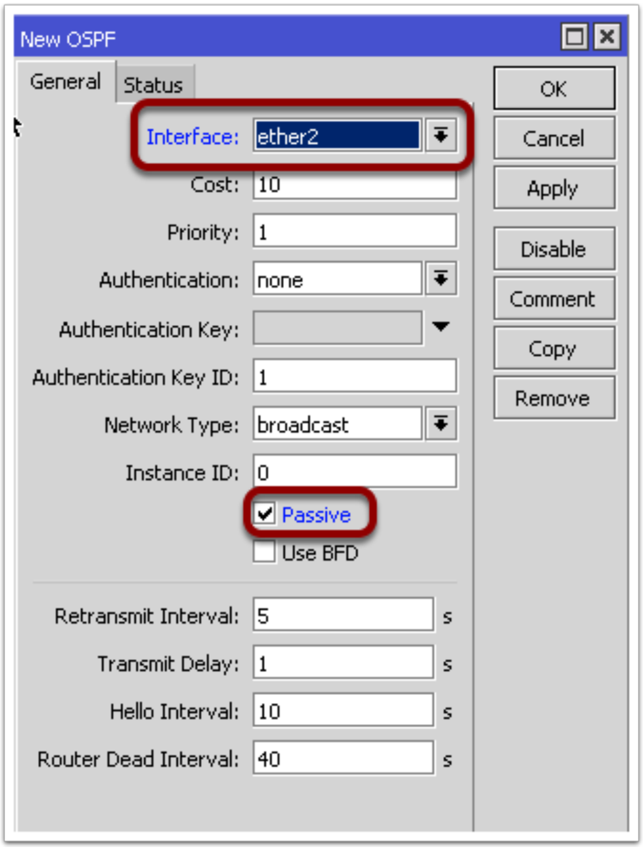
Area Name: area2
Instance: default
Area ID: 0.0.0.2
Type: default
Translator Role: translate never
□ Inject Summary LSAs
Default Cost: 1
Interfaces: 0
Active Interfaces: 0
Neighbors: 0
Adjacent Neighbors: 0
enabled

Figure 35. Creating New OSPF Area

The current dynamic routing configuration assumes that all of the OSPF areas take part in the OSPF process and that the connected networks contain other routers that might participate in the OSPF process. This might not always be the case. In the case of a host only network or a customer’s network it is prudent not to extend the dynamic routing process to those parts of the network (because of the waste of bandwidth and processing power and security implications such as rogue router participating in the OSPF process and so on).

5.4 Configuring Passive OSPF Interfaces in RouterOS

The aforementioned issues can be remedied by making the customer facing interface passive – its address will still be routed, but the interface will not take part in any OSPF process originating in that subnet. To configure an interface as passive select *"WinBox > Routing > OSPF > Interfaces"* and press *"+”* button, then choose a suitable interface from drop-down menu and tick the *"Passive"* option (see figure 36). Confirm by pressing *"OK"*.



Interface: ether2
Cost: 10
Priority: 1
Authentication: none
Authentication Key:
Authentication Key ID: 1
Network Type: broadcast
Instance ID: 0
☑ Passive
□ Use BFD
Retransmit Interval: 5 s
Transmit Delay: 1 s
Hello Interval: 10 s
Router Dead Interval: 40 s

Figure 36. Configuring Passive Interface

The *“Priority”* value in the *"New OSPF"* window can be used to increase or decrease the priority of that interface in DR and BDR elections. The higher the value, the higher the probability of becoming a DR. The *"Authentication"* drop-down menu offers a possibility of authentication of LSAs with MD5 hashes. Authentication key can be set in the field below. If configured, the router will discard all LSAs with nonmatching MD5 hashes. LSAs can be viewed in *"WinBox > Routing > OSPF > LSA"* tab (see figure 37).

The CLI setup of passive interfaces can be accomplished with this command:

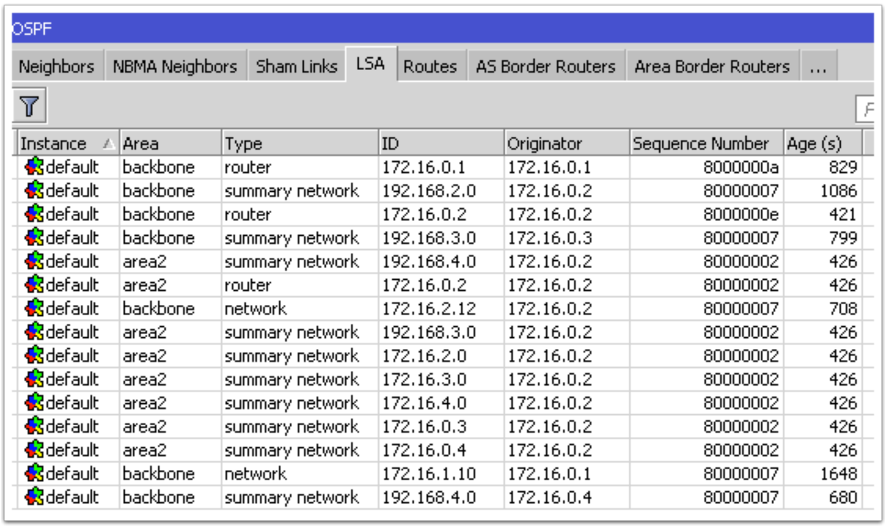
```
[admin@Router 2] > routing ospf interface add interface=ether2 network-type=broadcast passive=yes
```

And verified by the following command:

```
[admin@Router 2] > routing ospf interface print
```

The CLI command to view LSAs is as follows:

```
[admin@Router 2] > routing ospf lsa print
```



Instance	Area	Type	ID	Originator	Sequence Number	Age (s)
default	backbone	router	172.16.0.1	172.16.0.1	8000000a	829
default	backbone	summary network	192.168.2.0	172.16.0.2	80000007	1086
default	backbone	router	172.16.0.2	172.16.0.2	8000000e	421
default	backbone	summary network	192.168.3.0	172.16.0.3	80000007	799
default	area2	summary network	192.168.4.0	172.16.0.2	80000002	426
default	area2	router	172.16.0.2	172.16.0.2	80000002	426
default	backbone	network	172.16.2.12	172.16.0.2	80000007	708
default	area2	summary network	192.168.3.0	172.16.0.2	80000002	426
default	area2	summary network	172.16.2.0	172.16.0.2	80000002	426
default	area2	summary network	172.16.3.0	172.16.0.2	80000002	426
default	area2	summary network	172.16.4.0	172.16.0.2	80000002	426
default	area2	summary network	172.16.0.3	172.16.0.2	80000002	426
default	area2	summary network	172.16.0.4	172.16.0.2	80000002	426
default	backbone	network	172.16.1.10	172.16.0.1	80000007	1648
default	backbone	summary network	192.168.4.0	172.16.0.4	80000007	680

Figure 37. Link State Advertisements

The numerical values of the LSA types shown in Figure 37 are as follows: router LSA is type one, network LSA is type 2 and summary network LSA is type three (refer to table 1 for more details on LSA type and area type relationship).

5.5 Setting up GRE Tunnelling on RouterOS

For the purposes of this part of the laboratory, networks 192.168.10.0/24 and 192.168.2.0/24 will be considered as being part of the same company's internal network. As there is a section of the public Internet between them, a tunnel will be necessary to connect them. A guest PC is necessary to verify the functionality of the tunnel. It is possible to reuse the Guest PC VM from the previous laboratory exercise, by changing the vmnet it is connected to, the IP address of the ether1 interface and the default route's next hop address.

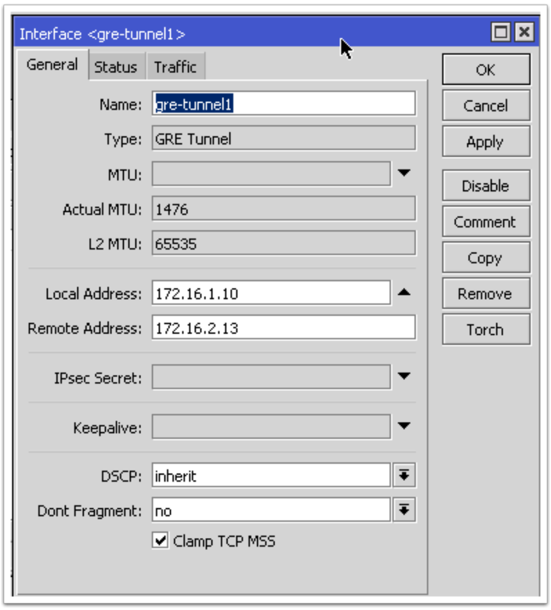
In order to create the tunnel, it is necessary to create endpoint interfaces on routers 1 and 4. To do so, navigate *"WinBox > Interfaces > GRE Tunnel"* and press "+", then add the local and remote interface's IP addresses according to figure 38.

Creating GRE tunnel interface can be accomplished in CLI with the following command:

```
[admin@Router 1] > interface gre add !keepalive local-address=172.16.1.10 name=gre-tunnel1 remote-address=172.16.2.13
```

Verify the configuration by issuing the following command:

```
[admin@Router 1] > interface gre print
```



Interface <gre-tunnel1>

Name: gre-tunnel1

Type: GRE Tunnel

MTU:

Actual MTU: 1476

L2 MTU: 65535

Local Address: 172.16.1.10

Remote Address: 172.16.2.13

IPsec Secret:

Keepalive:

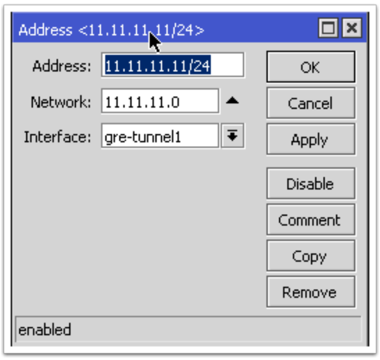
DSCP: inherit

Dont Fragment: no

☑ Clamp TCP MSS

Figure 38. Creating New GRE Tunnel Interface

Next the tunnel needs to be provided with the IP address: *"WinBox > IP > Address"*, then press *"+"* and give the interface a new IP address according to the network’s topology (see figure 39).



Address <11.11.11.11/24>

Address: 11.11.11.11/24

Network: 11.11.11.0

Interface: gre-tunnel1

enabled

Figure 39. Adding IP Address to GRE Tunnel Interface

This can be done in CLI as follows:

Add the IP address using the following command:

```
[admin@Router 1] > ip address add address=11.11.11.11/24 interface=gre-tunnel1
```

Verify the configuration by issuing the following command:

```
[admin@Router 1] > ip address print
```

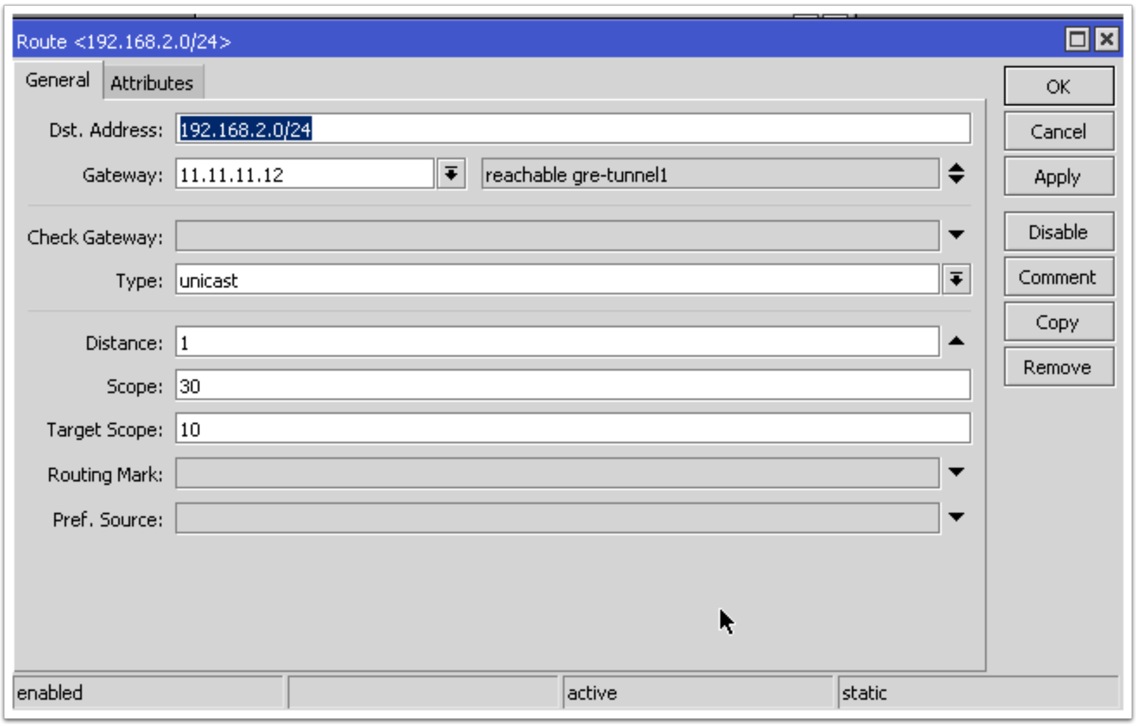
The last step is to create a route to the other end of the tunnel. A static route will suffice. Navigate "WinBox > IP > Routes", press "+" and input values from figure 40. Note that this and the previous steps need to be done at both ends of the tunnel for it to work.

CLI configuration on static routing through GRE tunnel is as follows:

```
[admin@Router 1] > ip route dst-address=192.168.2.0/24
gateway=11.11.11.12
```

It can be verified by the following command:

```
[admin@Router 1] > ip route print
```



Route <192.168.2.0/24>

Dst. Address: 192.168.2.0/24
Gateway: 11.11.11.12 reachable gre-tunnel1
Check Gateway:
Type: unicast
Distance: 1
Scope: 30
Target Scope: 10
Routing Mark:
Pref. Source:

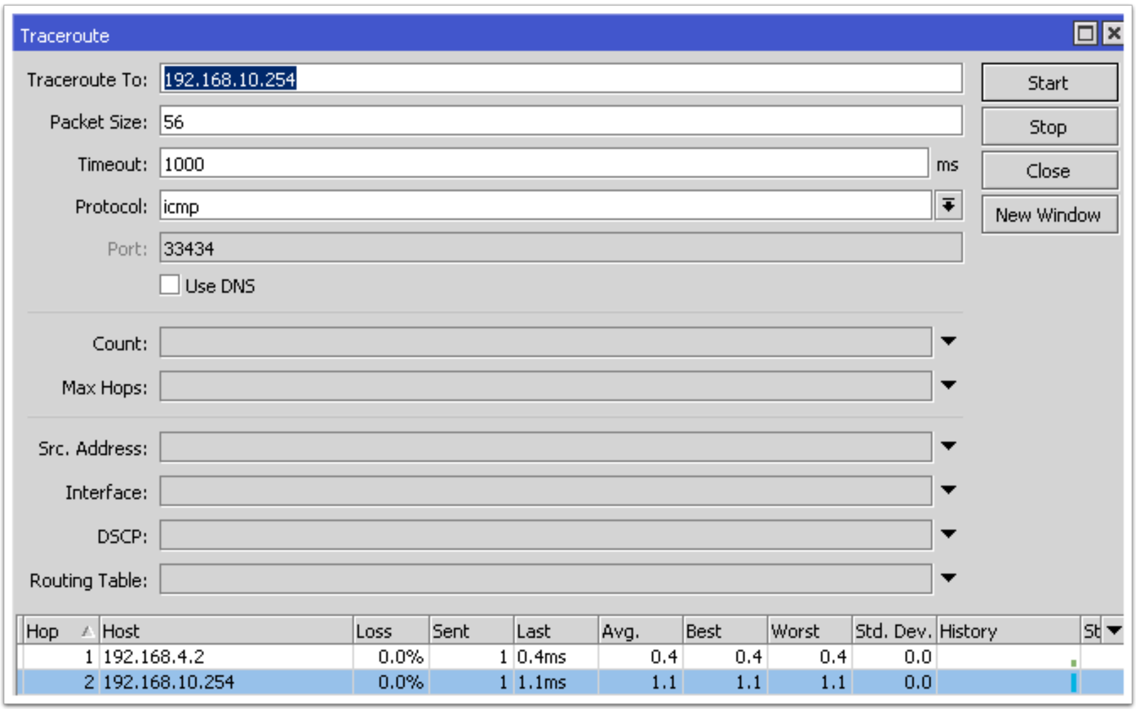
enabled | active | static

Figure 40. Creating a Route Through GRE Tunnel

Once the tunnel is configured on both routers, it will become active. To test its functioning, traceroute from the Guest PC to Router's 1 *"ether 1"* interface (IP address 192.168.10.254). The traceroute tool is available from *"WinBox > Tools > Traceroute".* The only value that needs to be provided is the destination address, which in this case is 192.168.10.254. The results of the traceroute command can be seen in figure 41. Two hops indicate that the traffic of the traceroute command indeed travelled through the tunnel. If this was not the case, the number of hosts traversed would be three: R4 > R2 > R1.

CLI traceroute command takes the following form on RouterOS:

```
[admin@Guest PC] > tool traceroute address=192.168.10.254
```



Traceroute To: 192.168.10.254

Packet Size: 56

Timeout: 1000 ms

Protocol: icmp

Port: 33434

□ Use DNS

Count:

Max Hops:

Src. Address:

Interface:

DSCP:

Routing Table:

Hop	Host	Loss	Sent	Last	Avg.	Best	Worst	Std. Dev.	History	St
1	192.168.4.2	0.0%	1	0.4ms	0.4	0.4	0.4	0.0		
2	192.168.10.254	0.0%	1	1.1ms	1.1	1.1	1.1	0.0		

Figure 41. Tracing Route Through GRE Tunnel

Note that in section 2.7 it was noted that GRE tunnels are stateless. That means, that if one end of the tunnel is down, the traffic sent through the tunnel will be blackholed. Mikrotik RouterOS offers a solution for this problem by letting the endpoints exchange “keepalive” messages. [21]

6 Discussion

The aim of the thesis was successfully reached – it is possible to create and test the functionality of arbitrarily complicated virtualised network topologies using RouterOS and VMware Fusion 8.0 Pro. The configuration files and configuration commands issued on the virtual routers can be used on physical devices without further adjustments. The network topology and its operation can be investigated using the same tools and methods as in the physical world.

There were, however, distinct problems related to the use of virtualisation and a virtualised environment. Due to the implementation of virtualisation it is not possible to create a pure peer-to-peer relationship between two VMs. Every network interface of a VM needs to be connected to some vSwitch/vmnet and only in a case where the two hosts are connected to the same vmnet they might communicate. Direct connections between hosts are impossible.

In a case where the vSwitches/vmnets are connected to the host machine, only hosts directly connected to the given vSwitch can ping it regardless of routing being configured or not. NAT masquerading allows for the host to be probed also from other vSwitches. ICMP traffic seems to be filtered when connecting the VMs to the real world using VMware Fusion’s built-in NAT interface. Pings will pass through to the physical world, and have a similar round trip time to those issued from the host machine. However, traceroute packets will not pass through. This makes verification and testing of the network topology implementation trickier.

I found out that RouterOS is not without quirks either. It uses autonegotiation to set the speed of its Ethernet interfaces. In my environment it invariably failed. Even explicit setting of the speed of the interface does not guarantee successful limitation of the traffic on that particular interface. The bandwidth tester feature will run at maximum achievable speed (in my case at hundreds of MB/s) regardless of the set speed of the interface. The speed can be successfully limed using a bandwidth limiter or by configuring the maximum data rate for the bandwidth tester function. Another problem I experienced was non-functioning static routes if the outbound interface was selected as the next hop address. If the next hop’s IP address was provided, routing worked normally.

While RouterOS can be run entirely without a graphical user interface (GUI), this is not possible at the time with VMware Fusion. The built-in `vmrun` command will provide (after modification of the system path) an easy access to a host of VM management functions, with a notable exception of VM creation from CLI. This can be sidestepped by creating and modifying a VM template (which is possible to do in CLI). However, it is not a trivial operation (compared to the guided VM creation process of the VMware Fusion GUI application) and so the use of GUI is required at least for some VM creation process steps.

While the goal of the thesis was met, it does not represent the whole array of networking techniques and tools required to run modern networks. The scope of this thesis was limited to routing in IPv4 only and topics such as IPv6 and switching were omitted. Switching was provided by VMware Fusion and only the most basic switching was used (that means no VLANs were utilised, no spanning tree protocol (STP) configuration was provided and so on).

With all the aforementioned limitations in mind the results of this thesis can be used as a foundation for more complicated/diverse network virtualisation tasks. Running five RouterOS VMs simultaneously had a negligible impact on the performance of the host machine and so more sophisticated network topologies are possible.

7 Conclusion

The goal of this thesis, to implement and test a computer network using Mikrotik RouterOS and VMware Fusion, was met. RouterOS proved to be straightforward to use (even though the syntax differs from Cisco IOS which is the most widely used) and all of the networking techniques, protocols and technologies I chose to use in the laboratory exercises were successfully implemented.

RouterOS and VMware Fusion can be used to develop and test network implementations with some limitations. If the limitations outlined in chapter 6 are not acceptable, networking specific virtualisation/emulation software such as open source GNS3 (Graphical Network Simulator-3) can be used instead of VMware Fusion.

This thesis aims to provide a foundation on which further exploration of RouterOS possibilities can be based. Topics for further research include IPv6 implementation, advanced switching or multiprotocol label switching (MPLS) and BGP implementation to simulate Internet service provider (ISP) networks. Moreover, OpenFlow is supported since RouterOS version 6 and so RouterOS switches can be used as a part of software defined networks (SDN) which is a technology that has been receiving considerable attention recently and is undergoing rapid development.

References

1 Mikrotik Ltd. About Us. [online]. Mikrotik Ltd: Riga, Latvia. 2015.
URL: http://www.mikrotik.com/aboutus. Accessed: 3 October 2015.

2 International Organization for Standardisation. International Standard ISO/IEC 7498-1: Information Technology – Open Systems Interconnection – Basic Reference Model: The Basic Model. Genève, Switzerland: ISO/IEC; 1994.

3 Certiology.com. OSI Layer Model [online]. 2015.
URL:http://www.certiology.com/wp-content/uploads/2014/03/OSI-Layer-Model.jpg Accessed: 10 October 2015.

4 Odom W. Official Cert Guide Cisco CCENT/CCNA ICND 100-101. Indianapolis, IN: Cisco Press; 2013.

5 National Telecommunication Information Administration. Telecommunications: Glossary of Telecommunications. Government Institutes; 1997. Switch. P. S-34.

6 Cisco Inc. Technical Support. What is a Network Switch vs. a Router? [online]. Cisco Inc.: San Jose, CA.
URL:http://www.cisco.com/cisco/web/solutions/small_business/resource_center/articles/connect_employees_and_offices/what_is_a_network_switch/index.html. Accessed: 20 October 2015.

7 Kozierok CM. Overview Of Key Routing Protocol Concepts: Architectures, Protocol Types, Algorithms and Metrics [online]. 20 September 2005.
URL:http://www.tcpipguide.com/free/t_OverviewOfKeyRoutingProtocolConceptsArchitecturesP-3.htm. Accessed: October 20 2015.

8 Cisco Systems et. al. Internetworking Technologies Handbook, 4th ed. Indianapolis, IN: Cisco Press; 2006.

9 Lammle T, Odom S, Wallace K. CCNP Routing Study Guide. Alameda, CA: Sybex Inc; 2001.

10 Cisco Systems Inc. Open Shortest Path First [online]. Cisco Inc.: San Jose, CA: 2012.
URL: http://docwiki.cisco.com/wiki/Open_Shortest_Path_First. Accessed: October 20 2015.

11 Stretch J. OSPF area types [online]. Raleigh, NC: 2008.
URL: http://packetlife.net/blog/2008/jun/24/ospf-area-types/. Accessed: October 20 2015.

12 Javvin Technologies Inc. Network Protocols Handbook. Javvin Technologies Inc.; 2005. NAT: Network Address Translation P. 27.

13 Lucas MW. Networking for Systems Administrators. Detroit, MI: Tilted Windmill Press; 2015.

14 Discher S. Load Balancing Using PCC & RouterOS [online]. ISP Supplies; 2012. URL: http://mum.mikrotik.com/presentations/US12/steve.pdf. Accessed: October 21 2015.

15 Mikrotik Wiki. ECMP Load Balancing with Masquerade [online]. Riga, Latvia: 2013.
URL: http://wiki.mikrotik.com/wiki/ECMP_load_balancing_with_masquerade. Accessed: October 22 2015.

16 SecPoint Ltd. What is Tunnelling Protocol? [online]. Copenhagen, Denmark: SecPoint Ltd.
URL: https://www.secpoint.com/what-is-tunneling-protocol.html. Accessed: October 23 2015.

17 Juniper Networks Inc. Understanding Generic Routing Encapsulation [online]. Sunnyvale, CA; 2012.
URL: http://www.juniper.net/techpubs/en_US/junos12.1/topics/concept/gre-tunnel-services.html. Accessed: October 23 2015.

18 Lowe S, Marshall N. et al. Mastering VMware vSphere 5.5. Indianapolis, IN: Sybex; 2014.

19 Wahl C, Pantol S. Networking for VMware Administrators. Palo Alto, CA: VMware Press/Pearson Publishing; 2014.

20 Primate Labs Inc. Mac Benchmarks [online]. Toronto, Canada; 2015.
URL: https://browser.primatelabs.com/mac-benchmarks. Accessed: November 1 2015.

21 Mikrotik Wiki. Manual: Interface/GRE [online]. Riga, Latvia: 2015.
URL: http://wiki.mikrotik.com/wiki/Manual:Interface/Gre. Accessed: November 2 2015.

Laboratory Exercise 1 – Configurations and Route Tables

Appendix 1.1 Configurations for Exercise 1.1

Router 1

```
/ip address
add address=10.0.0.254/24 interface=ether1 network=10.0.0.0
add address=172.16.0.2/24 interface=ether2 network=172.16.0.0
add address=172.16.1.2/24 interface=ether3 network=172.16.1.0
/system identity
set name="Router 1"
```

Router 2

```
/ip address
add address=172.16.0.3/24 interface=ether1 network=172.16.0.0
add address=172.16.1.3/24 interface=ether2 network=172.16.1.0
add address=172.16.2.2/24 interface=ether3 network=172.16.2.0
/system identity
set name="Router 2"
```

Router 3

```
/ip address
add address=172.16.2.3/24 interface=ether1 network=172.16.2.0
add address=192.168.2.2/24 interface=ether2 network=192.168.2.0
/system identity
set name="Router 3"
```

Guest PC

```
/ip address
add address=192.168.2.5/24 interface=ether1 network=192.168.2.0
/ip route
add distance=1 gateway=ether1
```

```
/system identity
set name="Guest PC"
```

Appendix 1.2 Configurations and Routing Tables After Configuring Static Routing

Router 1

```
/ip route
add gateway=10.0.0.1
add dst-address=172.16.2.0/24 gateway=172.16.1.3
add dst-address=172.16.2.0/24 gateway=172.16.0.3
add dst-address=192.168.2.0/24 gateway=172.16.1.3
add dst-address=192.168.2.0/24 gateway=172.16.0.3
```

Router 2

```
/ip route
add gateway=172.16.0.2
add gateway=172.16.1.2
add dst-address=10.0.0.0/24 gateway=172.16.0.2
add dst-address=10.0.0.0/24 gateway=172.16.1.2
add dst-address=192.168.2.0/24 gateway=172.16.2.3
```

Router 3

```
/ip route
add gateway=172.16.2.2
add dst-address=10.0.0.0/24 gateway=172.16.2.2
add dst-address=172.16.0.0/24 gateway=172.16.2.2
add dst-address=172.16.1.0/24 gateway=172.16.2.2
```

Guest PC

```
/ip route
add gateway=192.168.2.2
```

Routing tables:

Router 1

```
[admin@Router 1] > ip route print
Flags: X - disabled, A - active, D - dynamic,
C - connect, S - static, r - rip, b - bgp, o - ospf, m - mme,
B - blackhole, U - unreachable, P - prohibit
 #      DST-ADDRESS        PREF-SRC        GATEWAY
DISTANCE
 0 A S  0.0.0.0/0                          10.0.0.1
1
 1 ADC  10.0.0.0/24        10.0.0.254      ether1
0
 2 ADC  172.16.0.0/24      172.16.0.2      ether2
0
 3 ADC  172.16.1.0/24      172.16.1.2      ether3
0
 4 A S  172.16.2.0/24                      172.16.1.3
1
 5   S  172.16.2.0/24                      172.16.0.3
1
 6 A S  192.168.2.0/24                     172.16.0.3
1
 7   S  192.168.2.0/24                     172.16.1.3
1
```

Router 2

```
[admin@Router 2] > ip route print
Flags: X - disabled, A - active, D - dynamic,
C - connect, S - static, r - rip, b - bgp, o - ospf, m - mme,
B - blackhole, U - unreachable, P - prohibit
 #      DST-ADDRESS        PREF-SRC        GATEWAY
DISTANCE
```

```
 0 A S  0.0.0.0/0                       172.16.0.2
1
 1   S  0.0.0.0/0                       172.16.1.2
1
 2 A S  10.0.0.0/24                     172.16.0.2
1
 3   S  10.0.0.0/24                     172.16.1.2
1
 4 ADC  172.16.0.0/24      172.16.0.3      ether1
0
 5 ADC  172.16.1.0/24      172.16.1.3      ether2
0
 6 ADC  172.16.2.0/24      172.16.2.2      ether3
0
 7 A S  192.168.2.0/24                  172.16.2.3
1
```

Router 3

```
[admin@Router 3] > ip route print
Flags: X - disabled, A - active, D - dynamic,
C - connect, S - static, r - rip, b - bgp, o - ospf, m - mme,
B - blackhole, U - unreachable, P - prohibit
 #      DST-ADDRESS         PREF-SRC        GATEWAY
DISTANCE
 0 A S  0.0.0.0/0                       172.16.2.2
1
 1 A S  10.0.0.0/24                     172.16.2.2
1
 2 A S  172.16.0.0/24                   172.16.2.2
1
 3 A S  172.16.1.0/24                   172.16.2.2
1
 4 ADC  172.16.2.0/24      172.16.2.3      ether1
0
 5 ADC  192.168.2.0/24     192.168.2.2     ether2
0
```

Guest PC

```
[admin@Guest PC] > ip route print
Flags: X - disabled, A - active, D - dynamic,
C - connect, S - static, r - rip, b - bgp, o - ospf, m - mme,
B - blackhole, U - unreachable, P - prohibit
 #      DST-ADDRESS        PREF-SRC        GATEWAY
DISTANCE
 0 A S  0.0.0.0/0                          192.168.2.2
1
 1 ADC  192.168.2.0/24     192.168.2.5     ether1
0
```

Appendix 1.3 Route Tables After Inclusion of Floating Static Routes

Router 1

```
[admin@Router 1] > ip route print
Flags: X - disabled, A - active, D - dynamic,
C - connect, S - static, r - rip, b - bgp, o - ospf, m - mme,
B - blackhole, U - unreachable, P - prohibit
 #      DST-ADDRESS        PREF-SRC        GATEWAY
DISTANCE
 0 A S  0.0.0.0/0                          10.0.0.1
1
 1 ADC  10.0.0.0/24        10.0.0.254      ether1
0
 2 ADC  172.16.0.0/24      172.16.0.2      ether2
0
 3 ADC  172.16.1.0/24      172.16.1.2      ether3
0
 4 A S  172.16.2.0/24                      172.16.1.3
1
 5   S  172.16.2.0/24                      172.16.0.3
10
```

```
 6 A S  192.168.2.0/24                     172.16.0.3
1
 7   S  192.168.2.0/24                     172.16.1.3
10
```

Router 2

```
 [admin@Router 2] > ip route print
Flags: X - disabled, A - active, D - dynamic,
C - connect, S - static, r - rip, b - bgp, o - ospf, m - mme,
B - blackhole, U - unreachable, P - prohibit
 #      DST-ADDRESS        PREF-SRC        GATEWAY
DISTANCE
 0 A S  0.0.0.0/0                          172.16.0.2
1
 1   S  0.0.0.0/0                          172.16.1.2
10
 2 A S  10.0.0.0/24                        172.16.0.2
1
 3   S  10.0.0.0/24                        172.16.1.2
10
 4 ADC  172.16.0.0/24      172.16.0.3      ether1
0
 5 ADC  172.16.1.0/24      172.16.1.3      ether2
0
 6 ADC  172.16.2.0/24      172.16.2.2      ether3
0
 7 A S  192.168.2.0/24                     172.16.2.3
1
```

Appendix 1.4 Configuration of Load Balancing

Router 1

```
/ip route
add dst-address=172.16.2.0/24 gateway=172.16.1.3,172.16.0.3
add dst-address=192.168.2.0/24 gateway=172.16.1.3,172.16.0.3
/tool bandwidth-server
set authenticate=no
```

Router 2

```
/ip route
add gateway=172.16.1.2,172.16.0.2
add dst-address=10.0.0.0/24 gateway=172.16.1.2,172.16.0.2
```

Appendix 1.5 NAT Configuration

Router 1

```
/ip firewall nat
add action=masquerade chain=srcnat
```

Laboratory Exercise 2 - Configurations

Router 1

```
/interface bridge
add name=br1
/interface gre
add !keepalive local-address=172.16.1.10 name=gre-tunnel1 re-
mote-address=\
    172.16.2.13
/routing ospf instance
set [ find default=yes ] distribute-default=always-as-type-1
router-id=\
    172.16.0.1
/ip address
add address=192.168.10.254/24 interface=ether1 net-
work=192.168.10.0
add address=172.16.1.10/24 interface=ether2 network=172.16.1.0
add address=172.16.4.20/24 interface=ether3 network=172.16.4.0
add address=172.16.0.1 interface=br1 network=172.16.0.1
add address=11.11.11.11/24 interface=gre-tunnel1 net-
work=11.11.11.0
/ip route
add gateway=192.168.10.1
add dst-address=192.168.2.0/24 gateway=11.11.11.12
/routing ospf network
add area=backbone network=172.16.0.0/16
/system identity
set name="Router 1"
```

Router 2

```
/interface bridge
add name=br1
/routing ospf area
add area-id=0.0.0.2 name=area2
```

```
/routing ospf instance
set [ find default=yes ] router-id=172.16.0.2
/ip address
add address=172.16.1.11/24 interface=ether1 network=172.16.1.0
add address=192.168.2.2/24 interface=ether2 network=192.168.2.0
add address=172.16.2.12/24 interface=ether3 network=172.16.2.0
add address=172.16.0.2 interface=br1 network=172.16.0.2
/routing ospf interface
add interface=ether2 network-type=broadcast passive=yes
/routing ospf network
add area=backbone network=172.16.0.0/16
add area=area2 network=192.168.2.0/24
/system identity
set name=Router2
```

Router 3

```
/interface bridge
add name=br1
/routing ospf area
add area-id=0.0.0.3 name=area3
/routing ospf instance
set [ find default=yes ] router-id=172.16.0.3
/ip address
add address=172.16.4.21/24 interface=ether1 network=172.16.4.0
add address=172.16.3.22/24 interface=ether2 network=172.16.3.0
add address=192.168.3.2/24 interface=ether3 network=192.168.3.0
add address=172.16.0.3 interface=ether1 network=172.16.0.3
/routing ospf network
add area=backbone network=172.16.0.0/16
add area=area3 network=192.168.3.0/24
/system identity
set name="Router 3"
```

Router 4

```
/interface bridge
add name=br1
/interface gre
add !keepalive local-address=172.16.2.13 name=gre-tunnel1 remote-address=\
    172.16.1.10
/routing ospf area
add area-id=0.0.0.4 name=area4
/routing ospf instance
set [ find default=yes ] router-id=172.16.0.4
/ip address
add address=172.16.2.13/24 interface=ether1 network=172.16.2.0
add address=192.168.4.2/24 interface=ether2 network=192.168.4.0
add address=172.16.3.23/24 interface=ether3 network=172.16.3.0
add address=172.16.0.4 interface=br1 network=172.16.0.4
add address=11.11.11.12/24 interface=gre-tunnel1 network=11.11.11.0
/ip route
add dst-address=192.168.10.0/24 gateway=11.11.11.11
/routing ospf network
add area=backbone network=172.16.0.0/16
add area=area4 network=192.168.4.0/24
/system identity
set name="Router 4"
```

Guest PC

```
/ip address
add address=192.168.4.5/24 interface=ether1 network=192.168.4.0
/ip route
add gateway=192.168.4.2
/system identity
set name="Guest PC"
```