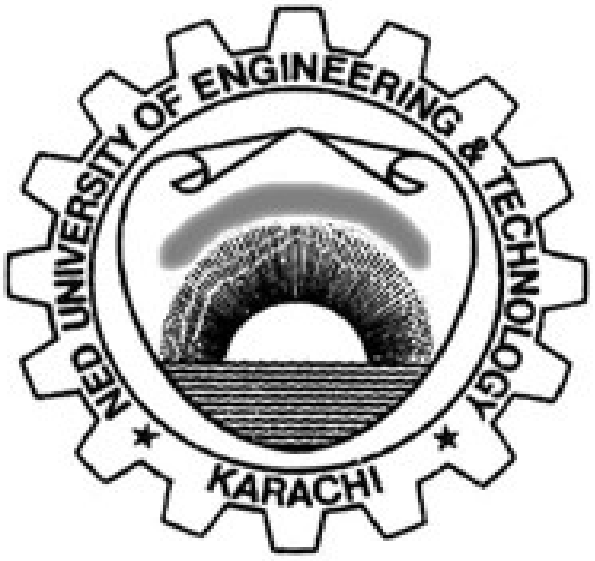
Practical Workbook

CS-351/CS-418
COMPUTER COMMUNICATION NETWORKS
(TCIT / SE / EE)



Name: _____

Year: _____

Batch: _____

Roll No.: _____

Department: _____

Department of Computer & Information Systems Engineering
NED University of Engineering & Technology

INTRODUCTION

The days of mainframe computing using dumb terminals are long gone. The present time is the era of very powerful personal computers, interconnecting with each other and even better equipped servers, sometimes connecting across continental boundaries.

Computer Communication Networks is a senior level undergraduate course in Computer and Information Systems Engineering, which covers various aspects of computer networks. It covers various classifications of computer networks and gives the students a good grasp on the various topics in computer networks. This laboratory manual aims to augment the classroom teaching of the course and to provide the students essential practical knowledge in the subject.

The first and second labs deal with learning IPv4 Addressing, Sub-netting & Variable Length Subnet Masking (VLSM).

The third lab deals with making crossover and straight-through UTP cables. This skill will come in very handy in various trades when the students go into practical life. It introduces some related standards and equipment used in this regard.

The fourth lab jumps into Cisco routers. It is a hands-on exercise using some commonly used Cisco IOS commands. In this lab, the students will learn how to connect to and interact with Cisco routers.

The fifth lab is about connecting different IP networks by defining static routes all around.

Sixth lab introduces dynamic routing using a simple routing protocol, namely RIP (Routing Information Protocol) and its later version called RIP version 2. In following two labs configuration and debugging of two more dynamic routing protocols are explored, that are OSPF and EIGRP.

The ninth lab teaches task of everyone's interest, i.e. connecting to internet using PPP

Labs through ten to fourteen are based on switching and cover basic LAN switch operation, loop avoidance using Spanning Tree Protocol and Virtual LANs and reducing administration overhead by using VLAN Trunking Protocol in switched network.

As careful as one might be, the disaster of lost, forgotten or stolen password will, nonetheless, strike sooner or later. Fifteenth, the last lab teaches how to do disaster recovery on a Cisco router in terms of recovering a forgotten password.

CONTENTS

Lab Session 01

OBJECT

Making the following kinds of UTP cables:

1. ***Straight through cable***
2. ***Cross cable***

THEORY

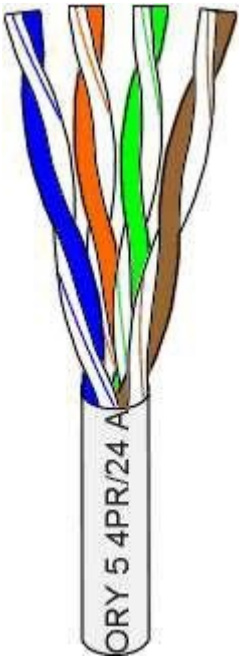


Figure 3.1: UTP cable

There are several classifications of twisted pair cable. Let's skip right over them and state that we'll use Category 5 (or CAT 5) cable for all new installations. Likewise, there are several fire code classifications for the outer insulation of CAT 5 cable. We'll use CMR cable, or "riser cable," for most of the wiring we do. You should also be aware of CMP or plenum cable (a plenum is used to distribute air in a building) you may be required by local or national codes to use the more expensive plenum-jacketed cable if it runs through suspended ceilings, ducts, or other areas, if they are used to circulate air or act as an air passage from one room to another. If in doubt, use plenum. CMR cable is generally acceptable for all applications not requiring plenum cable.

CAT 5 cable is available in reel-in-box packaging. This is very handy for pulling the wire without putting twists in it. Without this kind of package or a cable reel stand, pulling wire is a two-person job. Before the advent of the reel-in-box, we used to put a reel of wire on a broom handle to pull it. One person would hold the broom handle and the other would pull broom handle to pull it. You will produce a tangled mess, if your pull the wire off the end of the reel alone.

Standard wire patch cables are often specified for cable segments running form a wall jack to a PC and for patch panels. They are more flexible than solid core wire. However, the rationale for using it
is that the constant flexing of patch cables may wear-out solid core cable and break it. This is not a real concern in the average small network.

Most of the wiring we do simply connects computers directly to other computers or hubs. Solid core cable is quite suitable for this purpose and for many home and small business network. It is also quite acceptable for use as patch cables. You might consider a stranded wire patch cable if you have a notebook computer you are constantly moving around.

CAT 5 cable has four twisted pairs of wire for a total of eight individually insulated wires. Each pair is color coded with one wire having solid color (blue, orange, green, or brown) twisted around a second wire with a white background and a stripe of the same color. The solid color may have white stripe in some cables. Cable colors are commonly described using the background color followed by the color of the stripe; e.g; white-orange is a wire with a white background and an orange stripe.

Connectors

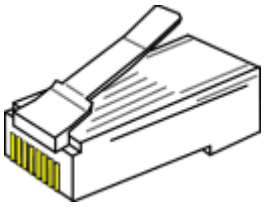


Figure 3.2: RJ-45 Connector

The straight through and cross-over patch cables are discussed in this article which are terminated with CAT 5 RJ-45 modular plugs. RJ-45 plugs are similar to those you'll see on the end of your telephone cable except they have eight as opposed to four or six contacts on the end of the plug and they are about twice as big. Make sure they are rated for CAT 5 wiring. (RJ stands for "Registered Jack"). Also, there are RJ-45 plugs designed for both solid core wire and stranded wire. Others are designed specifically for one kind of wire or the other. Be sure you buy plugs appropriate for the wire you are going to use. We normally use plugs designed to accommodate both kinds of wire.

Network cabling tools

1. Modular Plug Crimp Tool

Figure 3.3: Modular plug crimp tool

You will need a modular crimp tool. This is very similar to the ones which have been used for many years for all kinds of telephone cable work and it works just fine for Ethernet cables. You don't need a lot of bells and whistles, just a tool which will securely crimp RJ-45 connectors. Some crimpers have cutters which can be used to cut the cable and individual wires, and possibly stripping the outer jacket.

2. Universal UTP Stripping Tool (Eclipse)

It makes a much neater cut. It is highly recommending for anyone who will make a lot of cables.

Figure 3.4: Eclipse

3. Diagonal Cutters

It is easier to use diagonal cutters ("diags" or "dikes") to cut the cable off at the reel and to fine-tune the cable ends during assembly. Also, if you don't have a stripper, you can strip the cable by using a small knife to carefully slice the outer jacket longitudinally and use the diags to cut it off around the circumference.

Figure 3.5: Diagonal cutters

UTP basics

The 10BASE-T and 100BASE-TX Ethernet consist of two transmission lines. Each transmission line is a pair of twisted wires. One pair receives data signals and the other pair transmits data signals. A balanced line driver or transmitter is at one end of one of these lines

and a line receiver is at the other end. A (much) simplified schematic for one of these lines and its transmitter and receiver follows:

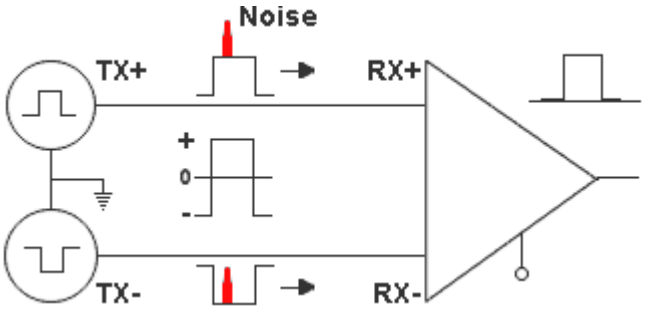


Figure 3.6: Schematic diagram of transmission line

Pulses of energy travel down the transmission line at about the speed of light (186,000 miles/second). The principal components of these pulses of energy are the potential difference between the wires and the current flowing near the surface of the wires. This energy can also be considered as residing in the magnetic field which surrounds the wires and the electric field between the wires. In other words, an electromagnetic wave which is guided by, and travels down the wires.

The main concern are the transient magnetic fields which surround the wires and the magnetic fields generated externally by the other transmission lines in the cable, other network cables, electric motors, fluorescent lights, telephone and electric lines, lightning, which may literally bury the Ethernet pulses, the conveyor of the information being sent down the line.

The twisted-pair Ethernet employs two principal means for combating noise. The first is the use of balanced transmitters and receivers. A signal pulse actually consists of two simultaneous pulses relative to ground: a negative pulse on one line and a positive pulse on the other. The receiver detects the total difference between these two pules. Since a pulse of noise usually produces pulses of the same polarity on both lines, it is essentially canceled out at the receiver. Also, the magnetic field surrounding one wire from a signal pulse is a mirror of the one on the other wire. At a very short distance from the two wires the magnetic fields are opposite and have a tendency to cancel the effect of each other out. This reduces the line's impact on the other pairs of wires and the rest of the world.

The second and the primary means of reducing cross-talk (the term cross-talk came from the ability to overhear conversations on other lines on your phone) between the pairs in the cable, is the double helix configuration produced by twisting the wires together. This configuration produces symmetrical (dentinal) noise signals in each wire. Ideally, their difference as detected at the receiver, is zero. In actuality it is much reduced.

Straight through and cross over cable

Again, the wire with colored backgrounds may have white stripes and may be denoted that way in diagrams found elsewhere. For example, the green wire may be labeled Green-White. The background color is always specified first.

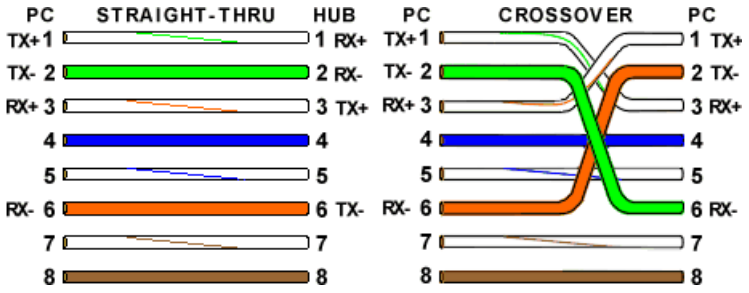


Figure 3.7: Straight through and crossover cable wire scheme

A Straight-through cable has identical ends, whereas a Crossover cable has different ends.

EIA/TIA 568A and 568B standards

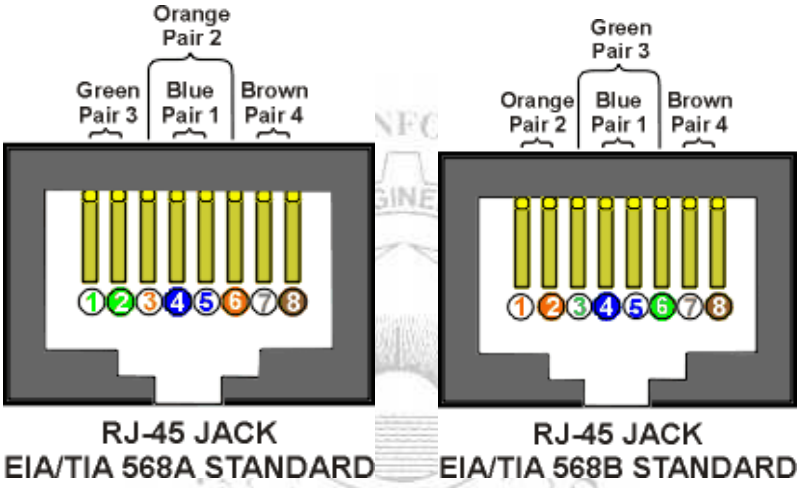


Figure 3.8: Cable connector standard ordering

It makes no functional difference which standard you use for a straight-through cable. Your can start a crossover cable with either standard as long as the other end is the other standard. It makes no functional difference which end is which. Despite what you may have read elsewhere, a 568A patch cable will work in a network with 568B wiring and 568B patch cable will work in a 568A network. The electrons couldn't care less.

Figure 3.9: EIA/TIA 568A and 568B

PROCEDURE

To Make Cable

1. Pull the cable off the reel to the desired length and cut the total length of wire segments between a PC and a hub or between two PC's cannot exceed 100 Meters (328 feet or about the length of a football field) for 100BASE-TX and 300 Meters for 100BASE-T.

2. Strip one end of the cable with the stripper or a knife and diags. If you are using the stripper, place the cable in the groove on the blade (left) side of the stripper and align the end of the cable with the right side of the stripper. This will strip about ½" of the jacket off the cable. Turn the stripper about 1 ¼ turn and pull. If you turn it more, you will probably nick the wires. If you are using knife and diags, carefully slit the cable for about an inch or so and neatly trim around the circumference of the cable with diags to remove the jacket.

3. Inspect the wires for nicks. Cut off the end and start over if you see any. You may have to adjust the blade with the screw at the front stripper. Cable diameters and jacket thicknesses vary.

4. Spread and arrange the pairs roughly in the order of the desired cable end.

5. Untwist the pairs and arrange the wires in the order of the desired cable end. Flatten the end between your thumb and forefinger. Trim the ends of the wires so they are even with one another.
   It is very important that the unstripped (untwisted) end be slightly less than ½" long. If it is longer than ½" it will be out-of-spec and susceptible to crosstalk. If it is less than ½" it will not be properly clinched when RJ-45 plug is crimped on. Flatten again. There should be little or no space between the wires.

6. Hold the RJ-45 plug with the clip facing down or away from you. Push the wire firmly into the plug. **Now, inspect before crimping and wasting the plug!** Looking through the bottom of the plug, the wire on the far-left side will have a white background. The wires should alternative light and dark from left to right. The furthest right wire is brown. The wires should all end evenly at the front of the plug. The jacket should end just about where you see it in the diagram-right on the line.

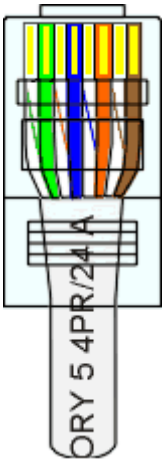


Figure 3.10:
Preparing the RJ-45 Connector

ALL ABOUT CRIMPING

7. Hold the wire near the RJ-45 plug with the clip down and firmly push it into the left side of the front of the Crimper (it will only go in one way). Hold the wire in place and squeeze the crimper handles quite firmly. This is what will happen:

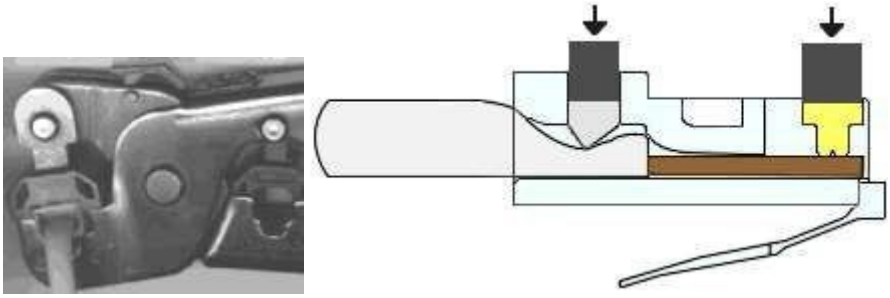


Figure 3.11: Crimping

(Crimp it once). The crimper pushes two plungers down on the RJ-45 plug. One forces, what amounts to, a cleverly designed plastic plug/wedge onto the cable jacket and very

firmly clinches it. The other seats the "pins", each with two teeth at its end, through the insulation and into the conductors of their respective wires.

8. Test the crimp… if done properly an average person will not be able to pull the plug off the cable with his or her bare hands. And that quite simply, besides lower cost, is the primary advantage of twisted-pair cables over the older thin wire, coaxial cables. In fact, the ease of installation and the modular RJ-45 plug is the main reason coaxial cable is no longer widely used for small Ethernet. But, don't pull that hard on the plug. It could stretch the cable and change its characteristics. Look at the side of the plug and see if it looks like the diagram and give it a fairly firm tug to make sure it is crimped well.

9. Prepare the other end of the cable so it has the desired end and crimp.

10. If both ends of the cable are within reach, hold them next to each other and with RJ-45 clips facing away. Look through the bottom of the plugs. If the plugs are wired correctly, and they are identical, it is a straight-through cable. If they are wired correctly and they are different, it is a crossover cable.

PRECAUTIONS

1. Try to avoid running cables parallel to power cables.

2. If you bundle a group of cables together with cable ties (zip ties), do not over-clinch them. It's okay to snug them together firmly; but don't tighten them so much that you deform the cables.

3. Keep cables away from devices which can introduce noise into them. Here's a short list: electric heaters, loud speakers, printers, TV sets, fluorescent light, copiers, welding machines, microwave ovens, telephones, fans, elevator motors, electric ovens, dryers, washing machines, and shop equipment.

4. Avoid stretching UTP cables (the force should not exceed 24 LBS).

5. Do not use a stapler to secure UTP cables. Use telephone wire hangers, which are available at most hardware stores.

EXERCISES

1. Give the reason why it is not advisable to bend UTP cables more than four times the diameter of the cable.

_____

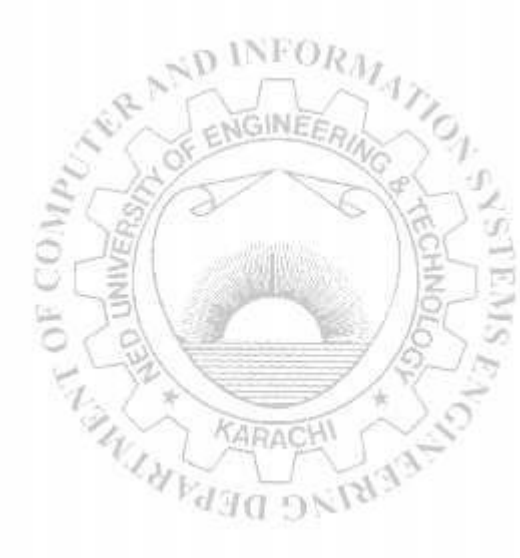
_____

_____

_____

_____

2. Why is it not advisable to run UTP cable outside of a building?



F/OBEM 01/05/00

NED University of Engineering & Technology
Department of Software Engineering
Course Code and Title: CS-351, Computer Communication Networks

Psychomotor Domain Assessment Rubric-Level P3					
Skill Sets	Extent of Achievement				
	1	2	3	4	5
Tool Utilization Effective **use** of software tools	Ineffective use of tools, significant misuse	Limited effective tool use, moderate misuse	Effective tool use with minor issues	Highly effective tool utilization	Mastery of tool utilization
Troubleshooting Issue **resolution** skills	Unable to troubleshoot effectively, significant issues persist	Limited troubleshooting abilities, moderate issues remain	Adequate troubleshooting skills, minor issues persist	Excellent troubleshooting skills, no issues remain	Mastery of troubleshooting
Task Accuracy ***Accuracy*** *in completing tasks*	Frequent errors and inaccuracies	Several errors, significant inaccuracies	Few errors, minor inaccuracies	No errors, highly accurate task completion	Mastery of accuracy
Task Completion Time Task completion **time management**	Tasks take significantly longer than expected	Tasks take longer than expected	Tasks completed within a reasonable timeframe	Tasks completed efficiently, ahead of schedule	Mastery of time management
Documentation Clarity and Organization ***Clarity. organization and structure*** *of documentation*	Poor, unclear, and highly disorganized documentation with no structure	Somewhat clear with a lack of organization and structure	Reasonably clear and organized with adequate structuring	Highly clear and highly organized and well-structured documentation	Mastery of documentation clarity, organization, and structure

Laboratory Session No. _____ Date: _____

Weighted CLO (Psychomotor Score)	
Remarks	
Instructor's Signature with Date:	

Lab Session 02

OBJECT

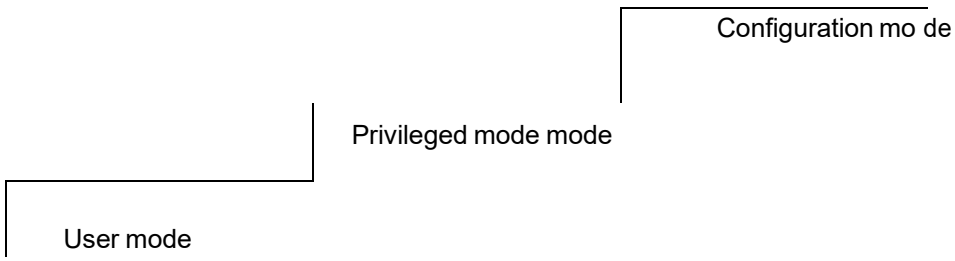
Practicing some basic commands to interact with the Cisco IOS (Internetwork Operating System) CLI Software

THEORY

Welcome to "hands on routing." The goal of this lab is to introduce you to Cisco routers and other equipment that you will be using throughout the semester. In order to do well in the labs, we need to understand the basic set-up of the lab.

- The lab has one rack, which is connected to a PC. You will be using the PC as a terminal to talk to the routers.
- The routers are labeled alphanumerically (Example R1, R2…)
- Each rack has two patch panels. One of them has RJ-45 connectors and the other has serial connectors. Ethernet ports are pre-connected to the RJ-45 patch panel. Serial ports are pre-connected to the serial patch panel. The ports are labeled on their left.
- To connect the PC to a specific router, connect the PC's console cable to the appropriate console port on the patch panel in the rack. You will find the console cable as a UTP cable with one of its ends connected through a small devices to a serial port on the PC.

Cisco routers support different modes of operation. When you access a router, it will typically be in the "user" mode. User mode gives a user access to simple "show commands." From user mode the next step is "Privileged mode." In the "Privileged mode" a user can have full access to all the databases maintained by the router. Cisco routers use many other modes, but let us keep it simple for now.



PROCEDURE

It is time to have fun:

1. Connect the PC to R1.

2. Press "`enter`" a few times and you should get a prompt that looks like: `router>`

3. You are now in the "user mode".

4. Type "`?`". Question mark lists commands that can be used in a certain context.

   First type "`help`"
   Try typing these commands:
   `p?`
   `pi?`

5. The IOS will complete commands for you with the help of the TAB key.

   Type `sh<TAB>`
   Finish the command with a "`?`" to see what commands you can use with show. (`show ?`)

6. You don't have to type a complete command for the IOS to execute it. You only need to type enough of a command to differentiate it from all other commands.

7. We have been operating in User Mode (identified by the prompt ending in >), now we want to go into the Privileged Mode:

   Type "`enable`" or "`en`"
   The prompt should end with a # (`Router#`)

   Type "`?`" to see all the commands possible from this mode

8. One of the most useful commands in the Cisco IOS is "`show.`" Try these variations:

   "`show configuration`" – shows saved router configuration
   "`show version`" - shows IOS statistics
   "`show startup-config`" – shows the configuration during startup
   "`show running-config`" – shows the dynamic configuration
   "`show flash`" – gives details of flash memory where IOS is stored
   "`show protocols`" – shows protocol and interface statistics
   "`show interface`" – gives detailed statistics on each interface
   "`show interface s0`" - Try this command with some other interfaces as well.

9. Now let's move to configuration mode. Type the following commands:

`configure terminal`
This will take you to configuration mode. The prompt ends with (`router-config)#?`
; to see the available commands

10. Next we will change the name of router to R1

11. Go into configuration mode and type the following commands:

```
hostname R1          ;this command will change name.
ctrl+Z               ;this is to come out of privilege mode
```

Now we want to set up an interface for a TCP/IP network.

Type these commands:

```
config t

interface Ethernet 0
```

This puts you in interface mode. Now you can configure interface Ethernet0.

```
ip address 130.10.20.5 255.255.255.0
```

This gives the interface an IP address and subnet mask.

```
no shutdown
```

By default all interface are administratively down. This command will bring them up.

```
ctrl+Z
```

This is to come out of privilege mode. Now type the following command:

```
sh interface e0
```

Observe and record carefully what you see.

Now connect a cable from router R1`s Ethernet 'e0' interface to a hub or switch. Again type this command:

```
sh interface e0
```

Again observe and record carefully what you see.

Note: Cisco commands are not case-sensitive.

EXERCISES

1. Determine which mode you operate in when you first access the router.

_____

_____

2. Start-up configuration is stored in NVRAM (true or false).
3. Running-configuration is stored in_____.
4. The command used to save changes made in the running configuration to start-up configuration is:

   _____

5. List the interfaces on three routers of your choice. Be sure to indicate the router number.

   _____

   _____

   _____

   _____

6. Elaborate on the information presented by the command "`show version`."

7. Which of the condition(s) are possible for an interface:
    a. administratively down, line protocol down
    b. administratively down, line protocol up
    c. administratively up, line protocol up
    d. administratively up, line protocol down

F/OBEM 01/05/00

NED University of Engineering & Technology
Department of Software Engineering
Course Code and Title: CS-351, Computer Communication Networks

Psychomotor Domain Assessment Rubric-Level P3					
Skill Sets	Extent of Achievement				
	1	2	3	4	5
Tool Utilization Effective **use** of software tools	Ineffective use of tools, significant misuse	Limited effective tool use, moderate misuse	Effective tool use with minor issues	Highly effective tool utilization	Mastery of tool utilization
Troubleshooting Issue **resolution** skills	Unable to troubleshoot effectively, significant issues persist	Limited troubleshooting abilities, moderate issues remain	Adequate troubleshooting skills, minor issues persist	Excellent troubleshooting skills, no issues remain	Mastery of troubleshooting
Task Accuracy ***Accuracy*** *in completing tasks*	Frequent errors and inaccuracies	Several errors, significant inaccuracies	Few errors, minor inaccuracies	No errors, highly accurate task completion	Mastery of accuracy
Task Completion Time Task completion **time management**	Tasks take significantly longer than expected	Tasks take longer than expected	Tasks completed within a reasonable timeframe	Tasks completed efficiently, ahead of schedule	Mastery of time management
Documentation Clarity and Organization ***Clarity. organization and structure*** *of documentation*	Poor, unclear, and highly disorganized documentation with no structure	Somewhat clear with a lack of organization and structure	Reasonably clear and organized with adequate structuring	Highly clear and highly organized and well-structured documentation	Mastery of documentation clarity, organization, and structure

Laboratory Session No. _____ Date: _____

Weighted CLO (Psychomotor Score)	
Remarks	
Instructor's Signature with Date:	

Lab Session 03

OBJECT

Configuring static routes on Cisco routers

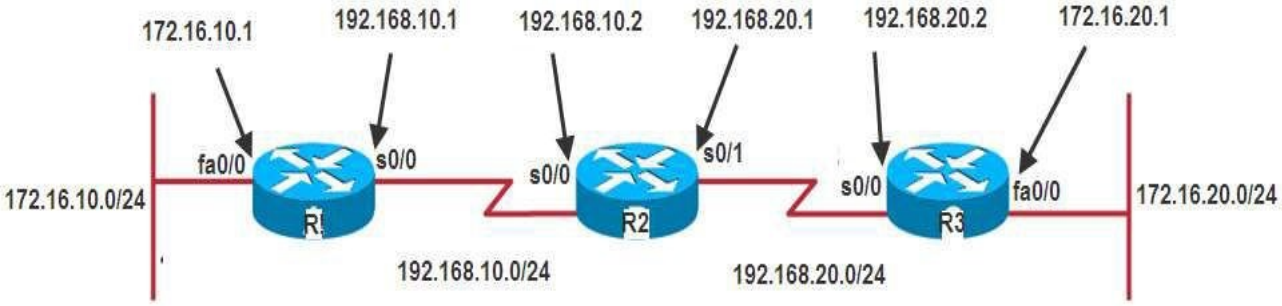


Figure 5.1: Scenario for static routes

THEORY

Routed & Routing Protocols

- A **Routed Protocol** is a protocol by which data can be routed. Routed protocols are IP, AppleTalk, and IPX. In this kind of protocols we require an addressing scheme and sub netting. Addressing scheme will be used to determine the network to which a host belongs and to identifying that host on that particular network. All hosts on an internetwork use the services of a routed protocol.
- A **Routing Protocol** is different and is only used between routers. It makes possible for routers to build and maintain routing tables. There are three classes of routing protocols-
  1) Distance Vector,
  2) Link State,
  3) Hybrid

Static & Dynamic Routing

The simplest method to route packets on a network is static routes. Although dynamic routing protocols are flexible and adjust to network changes, they do have associated network traffic which competes for network bandwidth with the user data traffic.

Configuring Static Routes

Static routes specify a fixed route for a certain destination network. They need to be configured on any router that needs to reach a network that it is not directly connected to. The IOS command used to configure static routes is `ip route`. The syntax is:

`ip route` *destination-address subnet-mask* {*ip-address* | *outgoing-interface*} [*distance*] [**tag** *tag*] [*permanent*]

where:

- *destination-address* is the destination address prefix for the network that we would like the router to reach
- *subnet-mask* is the subnet mask to be used on the address prefix to match for destination addresses. Multiple networks may be combined such that the destination-address and subnet-mask combination matches all hosts on those networks.
- *ip-address* specifies what ip address to forward a packet to if an IP packet arrives with a destination address that matches the destination-address subnet-mask pair specified in this command.
- Alternatively *outgoing-interface* specifies which interface the packet should be sent out of. Adding a static route to an Ethernet or other broadcast interface (for example, `ip route 0.0.0.0 0.0.0.0 Ethernet 1/2`) will cause the route to be inserted into the routing table only when the interface is up. This configuration is not generally recommended. When the next hop of a static route points to an interface, the router considers each of the hosts within the range of the route to be directly connected through that interface, and therefore it will send ARP requests to any destination addresses that route through the static route.
- *distance* is the optional administrative distance value for the route. If unspecified the default value is 1.
- *tag* value can be used as a "match" value for controlling redistribution via route maps.
- *permanenet* specifies that the route will not be removed even if the interface shuts down.

DTE/DCE

DCE and DTE are the interfaces. The DCE-DTE connection between routers is referred to as a null serial cable DCE(data communication equipment) and DTE (Data terminal equipment). DCE is located at the service provider end while the DTE is attached device.

The services that are given to the DTE is often accessed via modems or channel service unit/data service unit(CSU/DSU). DCE provides clocking and DTE receives the clock

PROCEDURE

1. Connect the network as shown in the network diagram.

2. Configure appropriate ip addresses and clock rates(if needed) on the router interfaces as specified in the network diagram.

3. For R1, enter the following static routes
   ip route 172.16.20.0 255.255.255.0 192.168.10.2
   ip route 192.168.20.0 255.255.255.0 192.168.10.2

4. On R2 enter:
   ip route 172.16.10.0 255.255.255.0 192.168.10.1
   ip route 172.16.20.0 255.255.255.0 192.168.20.2

5. On R3 enter:

ip route 172.16.10.0 255.255.255.0 192.168.20.1
ip route 192.168.10.0 255.255.255.0 192.168.20.1

6. After that verify the static routes by entering the following commands in the privilege mode:

router# sh ip route

EXERCISES

1. Run the command show IP route and write its output.

1. What is the default administrative distance of static route? Write the IP route command to modify the same.

3. Create a loop back interface on R3 and assign an IP address 10.1.0.1 /16 to it. Now add static routes to each of the other routers to reach this interface. Verify your work by pinging the newly created interface from routers R1 and R2 respectively.

Lab Session 04

OBJECT

Configuring RIP (Routing Information Protocol) and RIP version 2

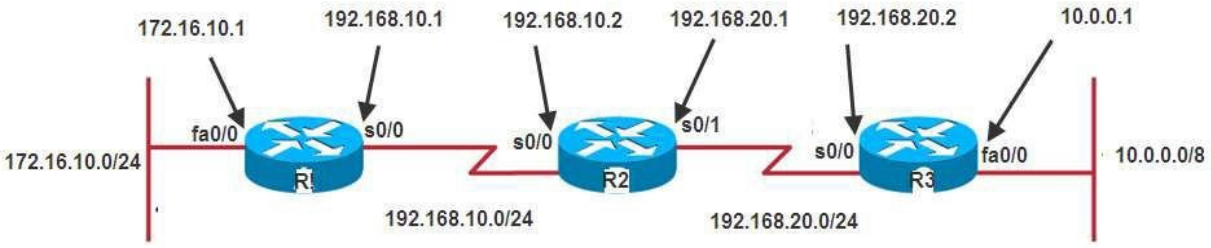


Figure 6.1: Scenario for RIP

THEORY

Distance Vector Routing Protocols

- Broadcast their entire routing table to each neighbor router at predetermined intervals
- The actual interval depends on the distance-vector routing protocol in use
- Varies between 30 and 90 seconds
- Sometimes referred to as *routing by rumor*
- Suffer from slow time to *convergence*
- *Convergence* is an state where all routers on the internetwork share a common view of the internetwork routes

Routing Information Protocol (RIP)

A distance-vector protocol, RIP was designed to work with small to medium-sized networks. RIP is an Interior Gateway Protocol (IGP), meaning it is used within an autonomous system. An autonomous system is a collection of networks under a single administration, sharing a common routing strategy.

RIP is easy to implement, compared to newer IGPs, and has been implemented in networks around the world. Advantage of using RIP, especially in small networks, is that there is very little overhead, in terms of bandwidth used and configuration and management time.

RIP Timers

RIP uses timers both to regulate its performance and to help prevent routing loops. All routers that use RIP send an update message to all of their neighbors approximately every 30 seconds; this process is termed *advertising*. The Cisco implementation sends updates every 30 seconds minus up to 15 percent, or 4.5 seconds.

If a neighbor has not responded in 180 seconds, it is assumed that the neighboring router is unavailable or the network connecting it to the router has become unusable. When the neighbor has not responded for 180 seconds, the route is marked invalid; 180 seconds is long enough that a route won't be invalidated by a single missed update message. The neighbor is shown to be unreachable by sending a normal update message with a metric of "infinity;" in the case of RIP, this number is 16. If an advertisement is received from a neighbor with a metric of infinity, then the route is placed into hold-down state, advertised with a distance of 16, and kept in the routing table. No updates from other neighbors for the same route are accepted while the route is in hold-down state. If other neighbors are still advertising the same route when the hold-down timer expires, then their updates will then be accepted. The route will be advertised with infinity metric for a period of time after the hold-down state if no alternate paths are found.

The actual timers used to accomplish the above tasks are a *routing-update timer*, a *route-invalid timer*, a *route-hold-down timer,* and a *route-flush* timer. The RIP routing-update timer is generally set to 30 seconds, ensuring that each router will send a complete copy of its routing table to all neighbors every 30 seconds. The route-invalid timer determines how much time must expire without a router having heard about a particular route before that route is considered invalid. When a route is marked invalid or put in hold-down state, neighbors are notified of this fact. This notification must occur prior to expiration of the route-flush timer. When the route flush-timer expires, the route is removed from the routing table. Typical initial values for these timers are 180 seconds for the route-invalid and route-holddown timers and 240 seconds for the route-flush timer. The values for each of these timers can be adjusted with the `timers basic` router configuration command.

Several Stability Features

To adjust for rapid network-topology changes, RIP specifies numerous stability features that are common to many routing protocols. RIP implements split horizon with poison-reverse and hold-down mechanisms to prevent incorrect routing information from being propagated. Split horizon prevents incorrect messages from being propagated by not advertising routes over an interface that the router is using to reach the route. Implementing split horizon helps avoid routing loops. Poison reverse operates by advertising routes that are unreachable with a metric of infinity back to the original source of the route. Hold-down is a method of marking routes invalid (expired). As discussed above, no updates from other neighbors for the same route are accepted while the route is in hold-down state.

Triggered updates are also an included convergence and stability feature. Updates are triggered whenever a metric for a route changes. Triggered updates may also contain only information regarding routes that have changed, unlike scheduled updates.

RIP version 2

RIPv2 is almost the same as the RIP version 1. RIPv2 also sends its complete routing table to its active interfaces at periodic time intervals. The timers, loop avoidance schemes and administrative distance are the same as Rip version 1.But RIPv2 is considered classless routing protocol because it also sends subnet information’s with each router. It also allows authentication using MD5 encryption scheme. And it also supports dis-contiguous networks. Configuring RIP version 2 on a router is very simple; it just requires one additional command.

PROCEDURE

Configuring RIP

1. Cable up the network as shown in the diagram.

2. Assign the IP address as shown in the diagram to the appropriate interfaces. For the serial links, has been used to indicate a DCE port.

3. Issue RIP routing commands on all the routers starting from the global config mode.

4. On R1:

```
router rip
network 172.16.10.0
network 192.168.10.0
```

On R2

```
router rip
network 192.168.10.0
network 192.168.20.0
```

On R 3

```
router rip
network 10.0.0.0
network 192.168.20.0
```

5. To verify the working of RIP ping one host, say H2, on LAN connected to R3 from the host, say H1, on LAN connected to R1. Also run some other debugging command to explore more.

Configuring RIP version 2

1. Issue the following commands on R1.

```
router rip
version 2
network 172.16.10.0
network 192.168.10.0
```

2. Repeat the same for R2 and R3.
3. Verify and debug, as you did earlier for RIP.

EXERCISES

1. Configure RIP on all three routers, note down routing table of router R1, and run command Debug ip rip to note the address on which updates are sent.

2. Write commands to modify the default update and hold-down timers.

3. Repeat exercise #1 for RIPv2 and note down the multicast address on which RIPv2 forwards the updates.

4. Write down the source IP address for the ping packets when you ping H1 from R1.

5. While working on R1, how could you check if H1 can reach the loopback interface? In other words, how can you verify if a ping from H1 to loopback of R1 is successful?

Lab Session 05

OBJECT

Configuring OSPF (Open Shortest Path First) Single Area

THEORY

Open Shortest Path First (OSPF) was developed by the Internet Engineering Task Force (IETF) as a replacement for the problematic RIP and is now the IETF-recommended Interior Gateway Protocol (IGP). OSPF is a link state protocol that, as the name implies, uses Dijkstra's Shortest Path First (SPF) algorithm. It is an open standards protocol—that is, it isn't proprietary to any vendor or organization. Link-state routing protocols perform the following functions:

- Respond quickly to network changes
- Send triggered updates only when a network change has occurred
- Send periodic updates known as *link-state refreshes*
- Use a *hello mechanism* to determine the reachability of neighbors
  - Each router keeps track of the state or condition of its directly connected neighbors by multicasting hello packets
- Each router also keeps track of all the routers in its network or area of the network by using *link-state advertisements (LSAs).*

Like all link state protocols, OSPF's major advantages over distance vector protocols are fast convergence, support for much larger internetworks, and less susceptibility to bad routing information. Other features of OSPF are:

- The use of areas, which reduces the protocol's impact on CPU and memory, contains the flow of routing protocol traffic, and makes possible the construction of hierarchical internetwork topologies
- Fully classless behavior, eliminating such class-full problems as dis-contiguous subnets. Support of classless route table lookups, VLSM, and super-netting for efficient address management
- A dimensionless, arbitrary metric
- Equal-cost load balancing for more efficient use of multiple paths
- Support of authentication for more secure routing
- The use of route tagging for the tracking of external routes

Characteristics of OSPF

Characteristic	OSPF
VLSM support	Yes
Manual summarization	Yes

Type of protocol	Link state
Classless support	Yes
Auto-summarization	No
Dis-contiguous support	Yes
Route propagation	Multicast on change
Hop count limit	None
Convergence	Fast
Peer authentication	Yes
Hierarchical network Updates/ Route computation	Event triggered/ Dijkstra

DR and BDR

DR (Designated Routers)

DR has the following duties:

- To represent the multi-access network and its attached routers to the rest of the internetwork
- To manage the flooding process on the multi-access network.
- The concept behind the DR is that the network itself is considered a "pseudo node," or a virtual router. Each router on the network forms an adjacency with the DR which represents the pseudo-node. Only the DR will send LSAs to the rest of the internetwork.

Note: router might be a DR on one of its attached multi-access networks, and it might not be the DR on another of its attached multi-access networks. In other words, the DR is a property of a router's interface, not the entire router.

BDR(Backup Designated Router):

A *Backup Designated Router (BDR)* is a hot standby for the DR on multi-access links. The BDR receives all routing updates from OSPF adjacent routers but doesn't flood LSA updates.

Note: if the router interface priority value is set to zero then that router won't participate in the DR or BDR elections on that interface.

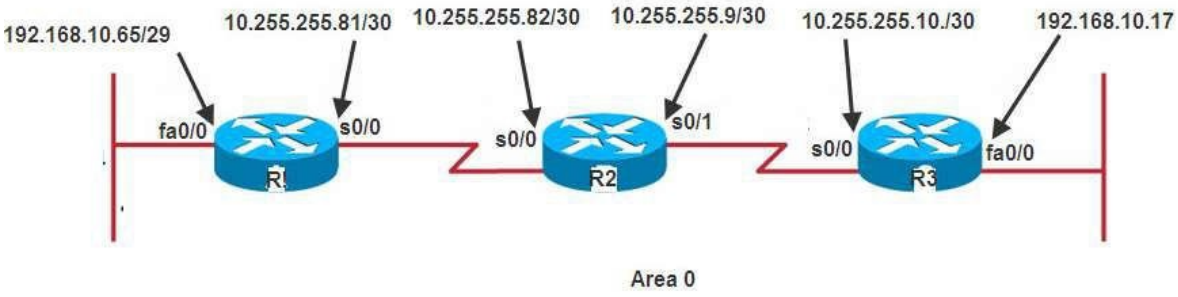


Fig 8.1: Scenario for OSPF implementation

After assigning ip addresses to interfaces of the routers the following IP Routing commands of OSPF on each other will be given as below.

```
Router A:

Router_A#config t
Router_A(config)#router ospf 1
Router_A(config-router)#network 192.168.10.64 0.0.0.7 area 0
Router_A(config-router)#network 10.255.255.80 0.0.0.3 area 0
```

The Router_A is using a /29 or 255.255.255.248 mask on the fa0/0 interface. This is a block size of 8, which is a wildcard of 7. The s0/0 interface is a mask of 255.255.255.252 block size of 4, with a wildcard of 3. Similarly the other subnet ,mask, and wildcard can be determined by looking at the IP address of an interface.

```
Router B:

Router_B#config t
Router_B(config)#router ospf 1
Router_B(config-router)#network 10.255.255.80 0.0.0.3 area 0
Router_B(config-router)#network 10.255.255.8 0.0.0.3 area 0

Router C:
Router_C#config t
Router_C(config)#router ospf 1
Router_C(config-router)#network 192.168.10.16 0.0.0.7 area 0
Router_C(config-router)#network 10.255.255.8 0.0.0.3 area 0
```

EXERCISES

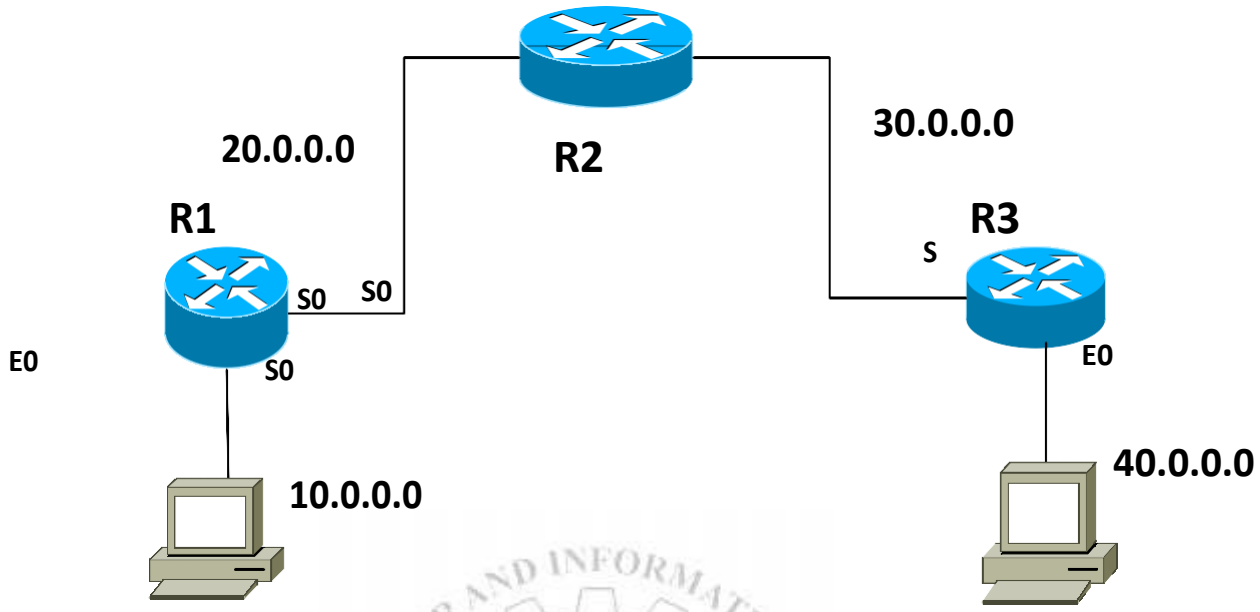


Fig 8.2: Scenario for exercise problems

Configure the network shown above on the routers in the lab. Assign appropriate IP addresses on the interfaces and configure OSPF on the routers. Write down the configuration commands entered on all three routers for configuration of OSPF.

1. Router 1:

2. Router 2:

3. Router 3:

Configure the network shown above on the routers in the lab. Assign appropriate IP addresses on the interfaces and configure EIGRP on the routers. Write down the configuration commands entered on all three routers for configuration of EIGRP.

1. Router 1:

2. Router 2:

3. Router 3:

F/OBEM 01/05/00

NED University of Engineering & Technology
Department of Software Engineering
Course Code and Title: CS-351, Computer Communication Networks

Psychomotor Domain Assessment Rubric-Level P3					
Skill Sets	Extent of Achievement				
	1	2	3	4	5
Tool Utilization Effective **use** of software tools	Ineffective use of tools, significant misuse	Limited effective tool use, moderate misuse	Effective tool use with minor issues	Highly effective tool utilization	Mastery of tool utilization
Troubleshooting Issue **resolution** skills	Unable to troubleshoot effectively, significant issues persist	Limited troubleshooting abilities, moderate issues remain	Adequate troubleshooting skills, minor issues persist	Excellent troubleshooting skills, no issues remain	Mastery of troubleshooting
Task Accuracy ***Accuracy*** *in completing tasks*	Frequent errors and inaccuracies	Several errors, significant inaccuracies	Few errors, minor inaccuracies	No errors, highly accurate task completion	Mastery of accuracy
Task Completion Time Task completion **time management**	Tasks take significantly longer than expected	Tasks take longer than expected	Tasks completed within a reasonable timeframe	Tasks completed efficiently, ahead of schedule	Mastery of time management
Documentation Clarity and Organization ***Clarity. organization and structure*** *of documentation*	Poor, unclear, and highly disorganized documentation with no structure	Somewhat clear with a lack of organization and structure	Reasonably clear and organized with adequate structuring	Highly clear and highly organized and well-structured documentation	Mastery of documentation clarity, organization, and structure

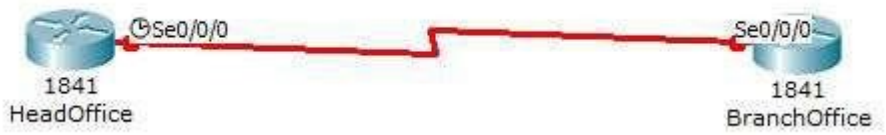
Laboratory Session No. _____ Date: _____

Weighted CLO (Psychomotor Score)	
Remarks	
Instructor's Signature with Date:	

Lab Session 06

OBJECT

Connecting two routers (Branch office and Head office) with the help of PPP



THEORY

PPP (Point-To-Point Protocol)

Short for **P***oint-to-***P***oint* **P***rotocol,* PPP is a method of connecting a computer to the Internet. PPP is more stable than the older SLIP protocol and provides error checking features. Working in the data link layer of the OSI model, PPP sends the computer's TCP/IP packets to a server that puts them onto the Internet.

NCP

A Network Control Protocol is a protocol that runs atop the Point-to-Point Protocol (PPP) and that is used to negotiate options for a network layer protocol running atop PPP. Network Control Protocols include the Internet Protocol Control Protocol for the Internet Protocol, the Internetwork Packet Exchange Control Protocol for the Internet Packet Exchange protocol, and the AppleTalk Control Protocol for AppleTalk. This protocol operates on the data link layer.

LCP

Short for Link Control Protocol, a protocol that is part of the PPP. In PPP communications, both the sending and receiving devices send out LCP packets to determine specific information that will be required for the data transmission. The LCP checks the identity of the linked device and either accepts or rejects the peer device, determines the acceptable packet size for transmission, searches for errors in configuration and can terminate the link if the parameters are not satisfied. Data cannot be transmitted over the network until the LCP packet determines that the link is acceptable.

Authentications methods of PPP

PAP and CHAP are two methods that PPP uses for authentication.

PAP (Password Authentication Protocol)

Short for Password Authentication Protocol, the most basic form of authentication, in which a user's name and password are transmitted over a network and compared to a table of name-password pairs. Typically, the passwords stored in the table are encrypted. The Basic Authentication feature built into the HTTP protocol uses PAP. The main weakness of PAP is that both the username and password are transmitted "in the clear" -- that is, in an unencrypted form. It's a two way hand shake method.

CHAP (Challenge Handshake Authentication Protocol)

Short for Challenge Handshake Authentication Protocol, CHAP is a type of authentication in which the authentication agent (typically a network server) sends the client program a random value that is used only once and an ID value. Both the sender and peer share a predefined secret. The peer concatenates the random value (or nonce), the ID and the secret and calculates a one-way hash using MD5. The hash value is sent to the authenticator, which in turn builds that same string on its side, calculates the MD5 sum itself and compares the result with the value received from the peer. If the values match, the peer is authenticated .By transmitting only the hash, the secret can't be reverse-engineered. The ID value is increased with each CHAP dialogue to protect against replay attacks. It's a three way hand shake method.

PROCEDURE

1. Change Hostname and assign Username and password on both routers. Assign the username of Branch Office Router in Head Office Router and Head Office Router username initialized on Branch Office Router but Password must be same on both Routers.

   HeadOffice Router:

```
Router(config)#hostname FasiRehman
FasiRehman(config)#usernam
FasiRehman(config)#username FasiRehmanCisco pas
FasiRehman(config)#username FasiRehmanCisco password 123
FasiRehman(config)#
```

   BranchOffice Router:

```
Router(config)#hostname FasiRehmanCisco
FasiRehmanCisco(config)#use
FasiRehmanCisco(config)#username FasiRehman pas
FasiRehmanCisco(config)#username FasiRehman password 123
```

2. Issue PPP debugging commands in privileged mode of both routers.

   - **debug ppp authentication:** The most common reasons for failed dial backup calls are incorrect dial strings and PPP authentication problems. You can easily diagnose both of these problems with this command.

- **debug ppp negotiation:** Displays PPP packets related to the negotiation of the PPP link.

```
FasiRehman#deb
FasiRehman#debug p
FasiRehman#debug ppp a
FasiRehman#debug ppp authentication
PPP authentication debugging is on
FasiRehman#de
FasiRehman#deb
FasiRehman#debug p[
FasiRehman#debug pp
FasiRehman#debug ppp n
FasiRehman#debug ppp negotiation
PPP protocol negotiation debugging is on
```

3. Assingning IP addresses on serial interfaces and enabling PPP on both routers

- **encapsulation ppp:** Change encapsulation from default HDLC to PPP
- **ppp authentication chap pap:** Define that the Link will use PAP authentication, but will try CHAP if PAP fails or is rejected by other side.

HeadOffice Router:

```
FasiRehman(config)#interface se
FasiRehman(config)#interface serial 0/0/0
FasiRehman(config-if)#ip ad
FasiRehman(config-if)#ip address 10.0.0.1 255.0.0.0
FasiRehman(config-if)#no shu
FasiRehman(config-if)#no shutdown

%LINK-5-CHANGED: Interface Serial0/0/0, changed state to down
FasiRehman(config-if)#c;lo
FasiRehman(config-if)#clo
FasiRehman(config-if)#clock ra
FasiRehman(config-if)#clock rate 64000
FasiRehman(config-if)#en
FasiRehman(config-if)#encapsulation pp
FasiRehman(config-if)#encapsulation ppp
FasiRehman(config-if)#
Serial0/0/0 PPP: Using default call direction
Serial0/0/0 PPP: Treating connection as a dedicated line
Serial0/0/0 PPP: Phase is ESTABLISHING, Active Open

FasiRehman(config-if)#ppp
FasiRehman(config-if)#ppp a
FasiRehman(config-if)#ppp authentication c
FasiRehman(config-if)#ppp authentication chap p
FasiRehman(config-if)#ppp authentication chap pap
```

BranchOffice Router:

```
FasiRehmanCisco(config)#interface serial 0/0/0
FasiRehmanCisco(config-if)#ip ad
FasiRehmanCisco(config-if)#ip address 10.0.0.2 255.0.0.0
FasiRehmanCisco(config-if)#no shu
FasiRehmanCisco(config-if)#no shutdown

FasiRehmanCisco(config-if)#
%LINK-5-CHANGED: Interface Serial0/0/0, changed state to up

FasiRehmanCisco(config-if)#en
FasiRehmanCisco(config-if)#encapsulation p
FasiRehmanCisco(config-if)#encapsulation ppp
FasiRehmanCisco(config-if)#
Serial0/0/0 PPP: Using default call direction
Serial0/0/0 PPP: Treating connection as a dedicated line
Serial0/0/0 PPP: Phase is ESTABLISHING, Active Open

Serial0/0/0 LCP: State is Open

Serial0/0/0 PPP: Phase is AUTHENTICATING

Serial0/0/0 IPCP: O CONFREQ [Closed] id 1 len 10

Serial0/0/0 IPCP: I CONFACK [Closed] id 1 len 10

Serial0/0/0 IPCP: O CONFREQ [Closed] id 1 len 10

Serial0/0/0 IPCP: I CONFACK [REQsent] id 1 len 10

Serial0/0/0 PPP: Phase is FORWARDING, Attempting Forward
Serial0/0/0 Phase is ESTABLISHING, Finish LCP
```

```
FasiRehmanCisco(config-if)#ppp authentication chap p
FasiRehmanCisco(config-if)#ppp authentication chap pap
FasiRehmanCisco(config-if)#
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0/0, changed state to do
wn

Serial0/0/0 LCP: State is Open

Serial0/0/0 PPP: Phase is AUTHENTICATING

Serial0/0/0 IPCP: O CONFREQ [Closed] id 1 len 10

Serial0/0/0 IPCP: I CONFREQ [Closed] id 1 len 10

Serial0/0/0 IPCP: O CONFACK [Closed] id 1 len 10

Serial0/0/0 IPCP: I CONFACK [Closed] id 1 len 10

Serial0/0/0 IPCP: O CONFREQ [Closed] id 1 len 10

Serial0/0/0 IPCP: I CONFREQ [REQsent] id 1 len 10

Serial0/0/0 IPCP: O CONFACK [REQsent] id 1 len 10

Serial0/0/0 IPCP: I CONFACK [REQsent] id 1 len 10

Serial0/0/0 PPP: Phase is FORWARDING, Attempting Forward
Serial0/0/0 Phase is ESTABLISHING, Finish LCP
Serial0/0/0 Phase is UP
```

EXERCISES

1. Run the command show ppp authentication and write its output.

2. Write down the difference between chap and pap

3. Write down the command to use chap and pap as a fall back method to one another

Lab Session 07

OBJECT

Studying and configuring Access Lists

THEORY

An access list is essentially a list of conditions that categorize packets. One of the most common and easiest to understand uses of access lists is filtering unwanted packets when implementing security policies. Access lists can even be used in situations that don't necessarily involve blocking packets.

There are a few important rules that a packet follows when it's being compared with an access list:

Rule#1
It's always compared with each line of the access list in sequential order—that is, it'll always start with the first line of the access list, then go to line 2, then line 3, and so on.

Rule#2
It's compared with lines of the access list only until a match is made. Once the packet matches the condition on a line of the access list, the packet is acted upon and no further comparisons take place.

Rule#3
There is an implicit "deny" at the end of each access list—this means that if a packet doesn't match the condition on any of the lines in the access list, the packet will be discarded. Each of these rules has some powerful implications when filtering IP packets with access lists, so keep in mind that creating effective access lists truly takes some practice.

There are two main types of access lists:

1. Standard access lists
2. Extended access lists

Standard access lists

These use only the source IP address in an IP packet as the condition test. All decisions are made based on the source IP address. This means that standard access lists basically permit or deny an entire suite of protocols. They don't distinguish between any of the many types of IP traffic such as web, Telnet, UDP, and so on.

Its command syntax is

```
access-list <number> {permit| deny} <destination> [log]
```

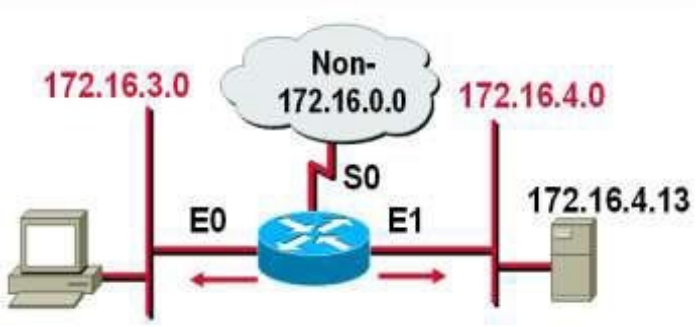


Fig 12.1: Standard Access list to allow my network

Commands on router will be

```
R1(config)#aaccess-list 1 permit 172.16.0.0 0.0.255.255
R1(config)#interface ethernet 0
R1(config)#ip access-group 1 out
R1(config)#interface ethernet 1
R1(config)#ip access-group 1 out
```

The above commands will permit the network 172.16.0.0 only and will block other network through the router on its ethernet interfaces in its out side directions

Extended access lists

Extended access lists can evaluate many of the other fields in the layer 3 and layer 4 headers of an IP packet. They can evaluate source and destination IP addresses, the protocol field in the Network layer header, and the port number at the Transport layer header. This gives extended access lists the ability to make much more granular decisions when controlling traffic.

Its command syntax is

```
access-list <number> {permit| deny}
<protocol><source>[<ports>]<destination>[ports][<options>]
```

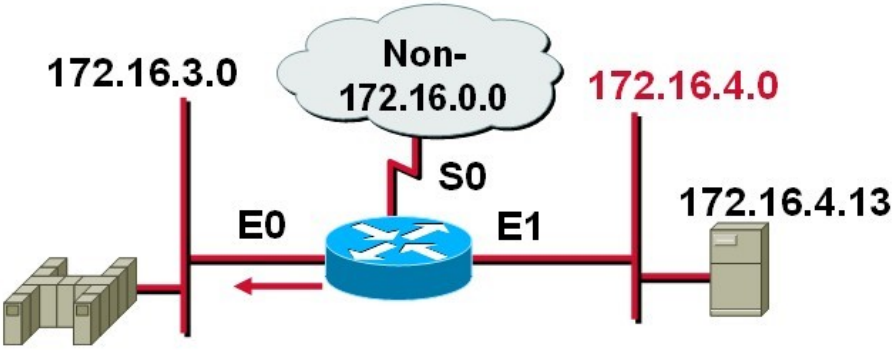


Fig 12.2: Extended access list

Commands on the router will be:

```
access-list 101 deny tcp 172.16.4.0 0.0.0.255 any eq 23
access-list 101 permit ip any any
interface ehternet 0
ip access-group 101 out
```

The above commands will Deny only the Telent from subnet 172.16.40.0 out of E0 and will permit all other traffic.

EXERCISE

Give commands to enable logging for the given access list and to show the entries that have been blocked

Lab Session 08

OBJECT

Studying basic LAN switch operation.

THEORY

LAN switch performs 3 operations

- Address learning
- Forward filter decision
- Loop avoidance

In this session, we will explore how an Ethernet switch learns addresses of the attached hosts.

Address learning

A new switch has empty MAC address table. As each frame transits switch, it learns source MAC address against the source port. As the switch does not know to which port the destination is attached, it initially transmits the frame to all ports. This process is called flooding. As the responses are received, the MAC address table is further populated.

PROCEDURE

Consider the following scenario

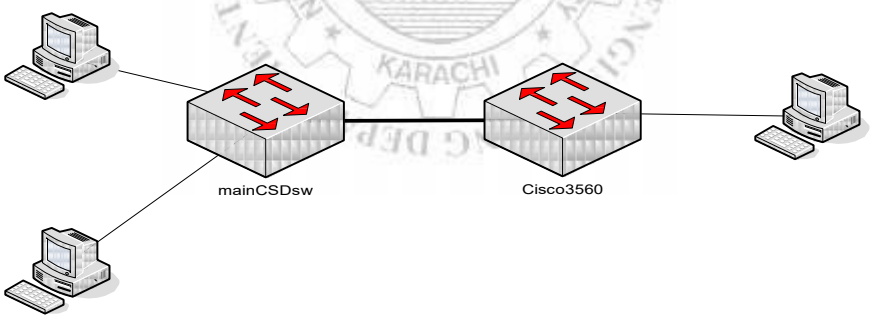


Fig 13.1: Scenario for LAN switch operation

Initially the MAC database of Cisco3560 will be

```
Switch#sh mac-address-table

          Mac Address Table
        -------------------------------------------

        Vlan    Mac Address       Type        Ports
        ----    -----------       --------    -----

           1    0006.2a75.100c    DYNAMIC     Fa0/1
Switch#
```

And that of mainCISDsw is;

```
mainCISDsw#sh mac-address-table

        Mac Address Table
      ------------------------------------------

      Vlan    Mac Address       Type        Ports
      ----    -----------       --------    -----

         1    0060.471b.ae01    DYNAMIC     Eth0/1

mainCISDsw#
```

Now as any of the computers generates ping for any of the remaining computers, the MAC address table will grow

```
Switch#sh mac-address-table

      Mac Address Table
      ------------------------------------------

      Vlan    Mac Address       Type        Ports
      ----    -----------       --------    -----

         1    0006.2a75.100c    DYNAMIC     Fa0/1
         1    0040.0ba5.183a    DYNAMIC     Fa0/1
         1    00e0.f7a4.475c    DYNAMIC     Fa0/2
Switch#
```

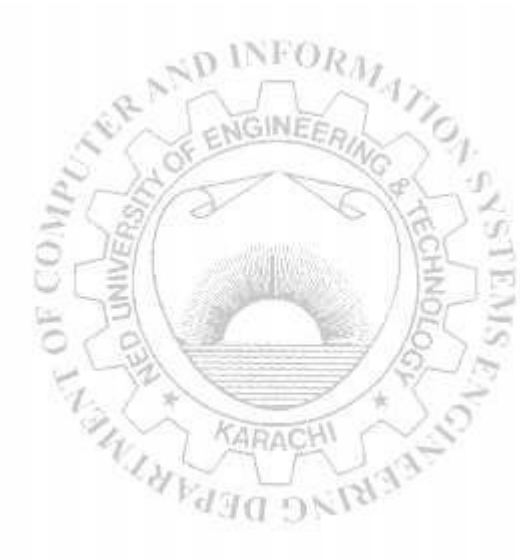
Also for mainCSDsw

```
mainCISDsw#sh mac-address-table

       Mac Address Table
      ------------------------------------------

      Vlan    Mac Address       Type        Ports
      ----    -----------       --------    -----

         1    0040.0ba5.183a    DYNAMIC     Eth1/1
         1    0060.471b.ae01    DYNAMIC     Eth0/1
         1    00e0.f7a4.475c    DYNAMIC     Eth0/1
mainCISDsw#
```

EXERCISES

1. If a destination MAC address is not in the forward/filter table, what will the switch do with the frame?

2. If a frame is received on a switch port and the source MAC address is not in the forward/filter table, what will the switch do?



F/OBEM 01/05/00

NED University of Engineering & Technology
Department of Software Engineering
Course Code and Title: CS-351, Computer Communication Networks

Psychomotor Domain Assessment Rubric-Level P3					
Skill Sets	Extent of Achievement				
	1	2	3	4	5
Tool Utilization Effective **use** of software tools	Ineffective use of tools, significant misuse	Limited effective tool use, moderate misuse	Effective tool use with minor issues	Highly effective tool utilization	Mastery of tool utilization
Troubleshooting Issue **resolution** skills	Unable to troubleshoot effectively, significant issues persist	Limited troubleshooting abilities, moderate issues remain	Adequate troubleshooting skills, minor issues persist	Excellent troubleshooting skills, no issues remain	Mastery of troubleshooting
Task Accuracy ***Accuracy*** *in completing tasks*	Frequent errors and inaccuracies	Several errors, significant inaccuracies	Few errors, minor inaccuracies	No errors, highly accurate task completion	Mastery of accuracy
Task Completion Time Task completion **time management**	Tasks take significantly longer than expected	Tasks take longer than expected	Tasks completed within a reasonable timeframe	Tasks completed efficiently, ahead of schedule	Mastery of time management
Documentation Clarity and Organization ***Clarity. organization and structure*** *of documentation*	Poor, unclear, and highly disorganized documentation with no structure	Somewhat clear with a lack of organization and structure	Reasonably clear and organized with adequate structuring	Highly clear and highly organized and well-structured documentation	Mastery of documentation clarity, organization, and structure

Laboratory Session No. _____ Date: _____

Weighted CLO (Psychomotor Score)	
Remarks	
Instructor's Signature with Date:	

Lab Session 09

OBJECT

Learning Loop Avoidance with Spanning Tree.

THEORY

The **Spanning Tree Protocol** (**STP**) is a link layer network protocol that ensures a loop-free topology for any switched LAN. Thus, the basic function of STP is to prevent switching loops and ensuing broadcast radiation.

In the OSI model for computer networking, STP falls under the OSI layer-2. It is standardized as 802.1D. As the name suggests, it creates a spanning tree within a mesh network of connected layer-2 switches (typically Ethernet switches), and disables those links that are not part of the spanning tree, leaving a single active path between any two network nodes.

Spanning tree allows a network design to include spare (redundant) links to provide automatic backup paths if an active link fails, without the danger of switch loops, or the need for manual enabling/disabling of these backup links. Switch loops must be avoided because they result in flooding the local network.

STP is based on an algorithm invented by Radia Perlman while working for Digital Equipment Corporation.

Protocol Operation

The collection of switches in a LAN can be considered a graph whose nodes are the bridges and the LAN segments (or cables), and whose edges are the interfaces connecting the bridges to the segments. To break loops in the LAN while maintaining access to all LAN segments, the bridges collectively compute a spanning tree. The spanning tree is not necessarily a minimum cost spanning tree. A network administrator can reduce the cost of a spanning tree, if necessary, by altering some of the configuration parameters in such a way as to affect the choice of the root of the spanning tree.

The spanning tree that the bridges compute using the Spanning Tree Protocol can be determined using the following rules.

Select a root bridge. The *root bridge* of the spanning tree is the bridge with the smallest (lowest) bridge ID. Each bridge has a unique identifier (ID) and a configurable priority number; the bridge ID contains both numbers. To compare two bridge IDs, the priority is compared first. If two bridges have equal priority, then the MAC addresses are compared. For example, if switches A (MAC=0200.0000.1111) and B (MAC=0200.0000.2222) both have a priority of 10, then switch A will be selected as the root bridge. If the network administrators would like switch B to become the root bridge, they must set its priority to be less than 10.

Determine the least cost paths to the root bridge. The computed spanning tree has the property that messages from any connected device to the root bridge traverse a least cost path, i.e., a path from the device to the root that has minimum cost among all paths from the device to the root. The cost of traversing a path is the sum of the costs of the segments on the path. Different technologies have different default costs for network segments. An administrator can configure the cost of traversing a particular network segment.

The property that messages always traverse least-cost paths to the root is guaranteed by the following two rules.

Least cost path from each bridge. After the root bridge has been chosen, each bridge determines the cost of each possible path from itself to the root. From these, it picks one with the smallest cost (a least-cost path). The port connecting to that path becomes the *root port* (RP) of the bridge.

Least cost path from each network segment. The bridges on a network segment collectively determine which bridge has the least-cost path from the network segment to the root. The port connecting this bridge to the network segment is then the *designated port* (DP) for the segment.

Disable all other root paths. Any active port that is not a root port or a designated port is a *blocked port* (BP).

Bridge Protocol Data Units (BPDUs)

The above rules describe one way of determining what spanning tree will be computed by the algorithm, but the rules as written require knowledge of the entire network. The bridges have to determine the root bridge and compute the port roles (root, designated, or blocked) with only the information that they have. To ensure that each bridge has enough information, the bridges use special data frames called **Bridge Protocol Data Units** (BPDUs) to exchange information about bridge IDs and root path costs.

A bridge sends a BPDU frame using the unique MAC address of the port itself as a source address, and a destination address of the STP multicast address 01:80:C2:00:00:00.

There are three types of BPDUs:

- Configuration BPDU (CBPDU), used for Spanning Tree computation
- Topology Change Notification (TCN) BPDU, used to announce changes in the network topology
- Topology Change Notification Acknowledgment (TCA)

BPDUs are exchanged regularly (every 2 seconds by default) and enable switches to keep track of network changes and to start and stop forwarding at ports as required.

When a device is first attached to a switch port, it will not immediately start to forward data. It will instead go through a number of states while it processes BPDUs and determines the topology of the network. When a host is attached such as a computer, printer or server the port

will always go into the forwarding state, albeit after a delay of about 30 seconds while it goes through the listening and learning states (see below). The time spent in the listening and learning states is determined by a value known as the forward delay (default 15 seconds and set by the root bridge). However, if instead another *switch* is connected, the port may remain in blocking mode if it is determined that it would cause a loop in the network. Topology Change Notification (TCN) BPDUs are used to inform other switches of port changes. TCNs are injected into the network by a non-root switch and propagated to the root. Upon receipt of the TCN, the root switch will set a Topology Change flag in its normal BPDUs. This flag is propagated to all other switches to instruct them to rapidly age out their forwarding table entries.

Spanning Tree port states:

- **Blocking** - A port that would cause a switching loop, no user data is sent or received but it may go into forwarding mode if the other links in use were to fail and the spanning tree algorithm determines the port may transition to the forwarding state. BPDU data is still received in blocking state.
- **Listening** - The switch processes BPDUs and awaits possible new information that would cause it to return to the blocking state.
- **Learning** - While the port does not yet forward frames (packets) it does learn source addresses from frames received and adds them to the filtering database (switching database)
- **Forwarding** - A port receiving and sending data, normal operation. STP still monitors incoming BPDUs that would indicate it should return to the blocking state to prevent a loop.
- **Disabled** - Not strictly part of STP, a network administrator can manually disable a port

Now consider the following topology

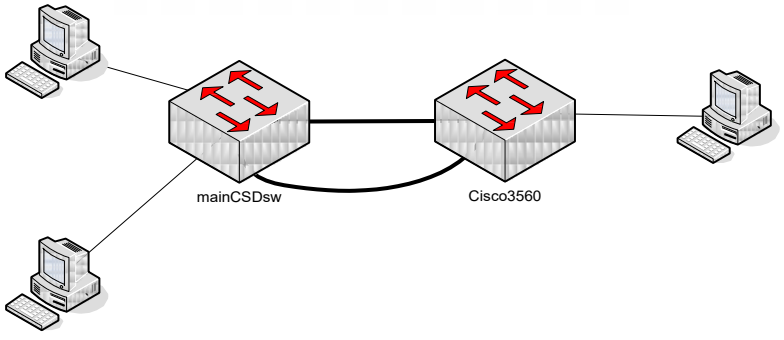


Fig 14.1: Scenario for implementing spanning tree

Here a physical loop can be observed

Now observe the spanning tree calculations for **mainCSDsw** first

```
mainCISDsw#sh spanning-tree
VLAN0001
```

```
  Spanning tree enabled protocol ieee
  Root ID     Priority    32769
              Address     0010.1100.58CE
              This bridge is the root
              Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID   Priority    32769  (priority 32768 sys-id-ext 1)
              Address     0010.1100.58CE
              Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec
              Aging Time  20

Interface         Role Sts Cost      Prio.Nbr Type
----------------- ---- --- --------- -------- --------------------------------
--
Et3/1             Desg FWD 100       128.4    P2p
Et2/1             Desg FWD 100       128.3    P2p
Et0/1             Desg FWD 100       128.1    P2p
Et1/1             Desg FWD 100       128.2    P2p

For cisco3560 the calculations will be

Switch#sh spanning-tree
VLAN0001
  Spanning tree enabled protocol ieee
  Root ID     Priority    32769
              Address     0010.1100.58CE
              Cost        100
              Port        1(FastEthernet0/1)
              Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID   Priority    32769  (priority 32768 sys-id-ext 1)
              Address     00E0.B02B.5EA0
              Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec
              Aging Time  20

Interface         Role Sts Cost      Prio.Nbr Type
----------------- ---- --- --------- -------- --------------------------------
--
Fa0/1             Root FWD 100       128.1    P2p
Fa0/3             Altn BLK 100       128.3    P2p
Fa0/2             Desg FWD 19        128.2    P2p
```

Modifying priorities and other parameters

To change default priority one can use the following command.

```
mainCISDsw(config)#spanning-tree vlan 1 priority 36864
```

Now see what happens to the root bridge.

```
mainCISDsw#sh spanning-tree
VLAN0001
  Spanning tree enabled protocol ieee
  Root ID     Priority    32769
              Address     00E0.B02B.5EA0
              Cost        100
              Port        1(Ethernet0/1)
              Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec
```

```
  Bridge ID  Priority    36865  (priority 36864 sys-id-ext 1)
             Address     0010.1100.58CE
             Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec
             Aging Time  20

Interface        Role Sts Cost      Prio.Nbr Type
---------------- ---- --- --------- -------- --------------------------------
--
Et3/1            Altn BLK 100       128.4    P2p
Et2/1            Desg FWD 100       128.3    P2p
Et0/1            Root FWD 100       128.1    P2p
Et1/1            Desg FWD 100       128.2    P2p
```

Other details on STP can be observed through the following set of commands under spanning tree.

```
Switch#sh spanning-tree ?
  active      Report on active interfaces only
  detail      Detailed information
  interface   Spanning Tree interface status and configuration
  summary     Summary of port states
  vlan        VLAN Switch Spanning Trees
  <cr>
```

EXERCISES

1. What is used to prevent switching loops in a network with redundant switched paths?

2. When is STP considered said to be converged?

Lab Session 10

OBJECT

Configuring Virtual LANs

THEORY

A **virtual LAN**, commonly known as a **VLAN**, is a group of hosts with a common set of requirements that communicate as if they were attached to the same broadcast domain, regardless of their physical location. A VLAN has the same attributes as a physical LAN, but it allows for end stations to be grouped together even if they are not located on the same network switch. Network reconfiguration can be done through software instead of physically relocating devices.

To physically replicate the functions of a VLAN, it would be necessary to install a separate, parallel collection of network cables and switches/hubs which are kept separate from the primary network. However unlike a physically separate network, VLANs must share bandwidth; two separate one-gigabit VLANs using a single one-gigabit interconnection can both suffer reduced throughput and congestion. It virtualizes VLAN behaviors (configuring switch ports, tagging frames when entering VLAN, lookup MAC table to switch/flood frames to trunk links, and untagging when exit from VLAN.)

Implementation

A basic switch not configured for VLANs will either have VLAN functionality disabled, or will have it permanently enabled with what is known as a *default VLAN* which simply contains all ports on the device as members.

Configuration of the first custom VLAN port group usually involves subtracting ports from the default VLAN, such that the first custom group of VLAN ports is actually the second VLAN on the device, apart from the default VLAN. The default VLAN typically has an ID of 1.

If a VLAN port group were to only exist on the one device, all ports that are members of the VLAN group only need to be "untagged". It is only when the port group is to extend to another device that tagging is used. For communications to occur from switch to switch, an uplink port needs to be a tagged member of every VLAN on the switch that uses that uplink port, including the default VLAN.

Some switches either allow or require a name be created for the VLAN, but it is only the VLAN group number that is important from one switch to the next.

Where a VLAN group is to simply pass through an intermediate switch via two pass-through ports, only the two ports need to be a member of the VLAN, and are tagged to pass both the required VLAN and the default VLAN on the intermediate switch.

Management of the switch requires that the management functions be associated with one of the configured VLANs. If the default VLAN were deleted or renumbered without moving the

management to a different VLAN first, it is possible to be locked out of the switch configuration, requiring a forced clearing of the device configuration to regain control.

Switches typically have no built-in method to indicate VLAN port members to someone working in a wiring closet. It is necessary for a technician to either have management access to the device to view its configuration, or for VLAN port assignment charts or diagrams to be kept next to the switches in each wiring closet. These charts must be manually updated by the technical staff whenever port membership changes are made to the VLANs.

Remote configuration of VLANs presents several opportunities for a technician to accidentally cut off communications and lock themselves out of the devices they are attempting to configure. Actions such as subdividing the default VLAN by splitting off the switch uplink ports into a separate new VLAN can suddenly cut off all remote communication, requiring the technician to physically visit the device in the distant location to continue the configuration process.

When inside the world of VLANs there are two types of links. These links allow us to connect multiple switches together or just simple network devices e.g PC, that will access the VLAN network. Depending on their configuration, they are called Access Links, or Trunk Links.

Access Links
Access Links are the most common type of links on any VLAN switch. All network hosts connect to the switch's Access Links in order to gain access to the local network. These links are your ordinary ports found on every switch, but configured in a special way, so you are able to plug a computer into them and access your network.

Trunk Link
A Trunk Link, or 'Trunk' is a port configured to carry packets for any VLAN. These type of ports are usually found in connections between switches. These links require the ability to carry packets from all available VLANs because VLANs span over multiple switches.

PROCEDURE

VLAN 1 is the default

```
Switch #sh int vlan 1

Vlan1 is administratively down, line protocol is down
  Hardware is CPU Interface, address is 00e0.b02b.5ea0 (bia 00e0.b02b.5ea0)
  MTU 1500 bytes, BW 100000 Kbit, DLY 1000000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
```

Configuring IP on default VLAN

```
Switch(config)#int vlan 1
Switch(config-if)#ip address 172.16.68.2 255.255.255.0
```

Creating VLANs

```
Switch(config)#int vlan 2
```

Assigning ports to vlans

```
Switch(config)#int fa 0/2
Switch(config-if)#switchport mode access
Switch(config-if)#switchport access vlan 2
```

Configuring trunk link

Consider the following topology

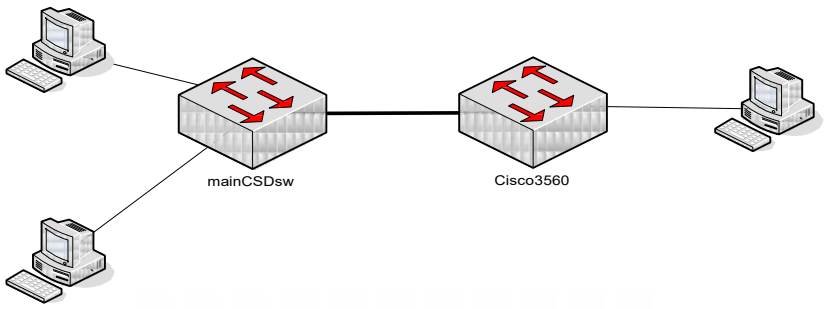


Fig 10.1: Scenario for implementing VLANs

Suppose mainCSDsw has two VLANs configured VLAN1 and VLAN2, whereas cisco3560 has only VLAN1. Now both switched must have at least one common trunk link connecting the two switches, so that the PCs which are in VLAN1 may communicate. Here we have interface fa 0/1 on each switch connected to the other. Hence the configuration would be

```
Switch(config)#int fa 0/1
Switch(config-if)#switchport mode trunk
```

Verification of configurations

```
Switch#show interface switchport
Name: Fa0/1
Switchport: Enabled
Administrative Mode: trunk
Operational Mode: trunk
Administrative Trunking Encapsulation: dot1q
Operational Trunking Encapsulation: dot1q
Negotiation of Trunking: On
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 1 (default)
Voice VLAN: none
Administrative private-vlan host-association: none
Administrative private-vlan mapping: none
Administrative private-vlan trunk native VLAN: none
Administrative private-vlan trunk encapsulation: dot1q
Administrative private-vlan trunk normal VLANs: none
Administrative private-vlan trunk private VLANs: none
Operational private-vlan: none
Trunking VLANs Enabled: All
Pruning VLANs Enabled: 2-1001
Capture Mode Disabled
Capture VLANs Allowed: ALL
Protected: false
Unknown unicast blocked: disabled
Unknown multicast blocked: disabled
```

```
Appliance trust: none

Name: Fa0/2
Switchport: Enabled
Administrative Mode: static access
Operational Mode: static access
Administrative Trunking Encapsulation: dot1q
Operational Trunking Encapsulation: native
Negotiation of Trunking: Off
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 1 (default)
Voice VLAN: none
Administrative private-vlan host-association: none
Administrative private-vlan mapping: none
Administrative private-vlan trunk native VLAN: none
Administrative private-vlan trunk encapsulation: dot1q
Administrative private-vlan trunk normal VLANs: none
Administrative private-vlan trunk private VLANs: none
Operational private-vlan: none
Trunking VLANs Enabled: All
```

A more handy way of verifying VLAN memberships would be

```
mainCISDsw#sh vlan brief

VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------
1    default                          active    Eth2/1, Eth3/1, Eth4/1
2    VLAN0002                         active    Eth1/1
1002 fddi-default                     active
1003 token-ring-default               active
1004 fddinet-default                  active
1005 trnet-default                    active
```

EXERCISES

1. What does trunking provide?

_____

_____

_____

2. What type of link is only part of one VLAN and is referred to as the “native VLAN” of the port?

_____

_____

_____

_____

F/OBEM 01/05/00

NED University of Engineering & Technology
Department of Software Engineering
Course Code and Title: CS-351, Computer Communication Networks

Psychomotor Domain Assessment Rubric-Level P3					
Skill Sets	Extent of Achievement				
	1	2	3	4	5
Tool Utilization Effective **use** of software tools	Ineffective use of tools, significant misuse	Limited effective tool use, moderate misuse	Effective tool use with minor issues	Highly effective tool utilization	Mastery of tool utilization
Troubleshooting Issue **resolution** skills	Unable to troubleshoot effectively, significant issues persist	Limited troubleshooting abilities, moderate issues remain	Adequate troubleshooting skills, minor issues persist	Excellent troubleshooting skills, no issues remain	Mastery of troubleshooting
Task Accuracy ***Accuracy*** *in completing tasks*	Frequent errors and inaccuracies	Several errors, significant inaccuracies	Few errors, minor inaccuracies	No errors, highly accurate task completion	Mastery of accuracy
Task Completion Time Task completion **time management**	Tasks take significantly longer than expected	Tasks take longer than expected	Tasks completed within a reasonable timeframe	Tasks completed efficiently, ahead of schedule	Mastery of time management
Documentation Clarity and Organization ***Clarity. organization and structure*** *of documentation*	Poor, unclear, and highly disorganized documentation with no structure	Somewhat clear with a lack of organization and structure	Reasonably clear and organized with adequate structuring	Highly clear and highly organized and well-structured documentation	Mastery of documentation clarity, organization, and structure

Laboratory Session No. _____ Date: _____

Weighted CLO (Psychomotor Score)	
Remarks	
Instructor's Signature with Date:	

Lab Session 11

OBJECT

To Configure VTP (VLAN Trunking Protocol) on Cisco Switches

THEORY

VTP

VLAN Trunk Protocol (VTP) reduces administration in a switched network. When you configure a new VLAN on one VTP server, the VLAN is distributed through all switches in the domain. This reduces the need to configure the same VLAN everywhere. VTP is a Cisco-proprietary protocol that is available on most of the Cisco Catalyst series products.

A switch using VTP can be configured in one of three modes: server, client, or transparent.

VTP Server Mode

- By default the switch is in this mode.
- VTP servers advertise the VTP domain VLAN information to other VTP-enabled switches in the same VTP domain
- VTP servers store the VLAN information for the entire domain in NVRAM
- The server is where VLAN is created, deleted, or renamed for the domain
- Synchronized VLAN information

VTP Transparent Mode

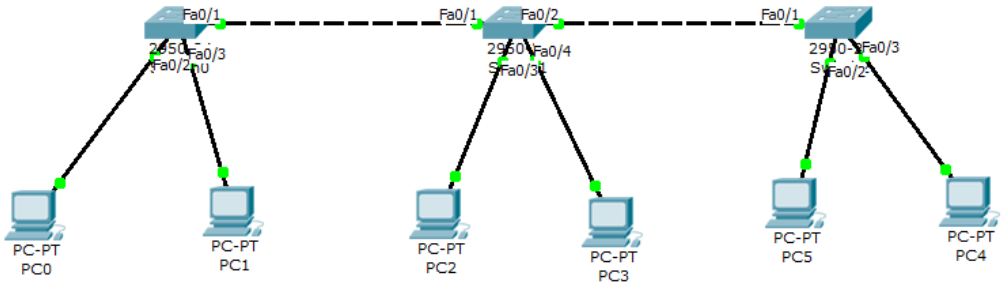
- VTP transparent advertise the VTP domain VLAN information to other VTP-enabled switches in the same VTP domain.
- VTP transparent store the VLAN information for the entire domain in NVRAM
- The transparent is where VLAN is created, deleted, or renamed for the domain
- VLANs that are created, renamed, or deleted on transparent switches are local to that switch only, hence cannot synchronize VLAN information

VTP Client Mode

- VTP clients function the same way as VTP servers.
- VTP clients advertise the VTP domain VLAN information to other VTP-enabled switches in the same VTP domain.
- VTP clients cannot store the VLAN information for the entire domain in NVRAM
- The client is where VLAN cannot be created, deleted, or renamed for the domain
- Synchronized VLAN information

PROCEDURE

1. Design a topology having 3 switches and 2 end devices in each switch. The figure shows the architecture of the topology.



2. Create trunk port in every switch's ports. Then create a VTP domain in the first switch named 'ahsandm', using command: '**vtp domain ahsandm**'. Now allocate a password to the switch by using command: '**vtp password abc**'. The below commands describes all the steps.

```
Switch(config)#hostname ahsan
ahsan(config)#inter
ahsan(config)#interface fa0/1
ahsan(config-if)#sw
ahsan(config-if)#switchport mode
ahsan(config-if)#switchport mode tr
ahsan(config-if)#switchport mode trunk
ahsan(config-if)#exit
ahsan(config)#vtp domain ahsandm
Changing VTP domain name from NULL to ahsandm
ahsan(config)#vtp password abc
Setting device VLAN database password to abc
```

3. Apply the same password on other two switches as well.

4. Now check the VTP status using command: '**show vtp status**' in all the switches

```
ahsan#shw
ahsan#sho
ahsan#show vt
ahsan#show vtp st
ahsan#show vtp status
VTP Version                     : 2
Configuration Revision          : 0
Maximum VLANs supported locally : 255
Number of existing VLANs        : 5
VTP Operating Mode              : Server
VTP Domain Name                 : ahsandm
VTP Pruning Mode                : Disabled
VTP V2 Mode                     : Disabled
VTP Traps Generation            : Disabled
MD5 digest                      : 0xD2 0x64 0xE7 0xCB 0xBE 0x3A 0xF6 0x
Configuration last modified by 0.0.0.0 at 0-0-00 00:00:00
Local updater ID is 0.0.0.0 (no valid interface found)
```

5. Now create 3 new VLAN in the first switch

```
ahsan(config)#vlan 3
ahsan(config-vlan)#exit
ahsan(config)#vlan 4
ahsan(config-vlan)#exit
ahsan(config)#vlan 2
ahsan(config-vlan)#exit
ahsan(config)#
ahsan(config)#do show vlan brief

VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
1    default                          active    Fa0/2, Fa0/3, Fa0/4, Fa0/5
                                                Fa0/6, Fa0/7, Fa0/8, Fa0/9
                                                Fa0/10, Fa0/11, Fa0/12, Fa0/13
                                                Fa0/14, Fa0/15, Fa0/16, Fa0/17
                                                Fa0/18, Fa0/19, Fa0/20, Fa0/21
                                                Fa0/22, Fa0/23, Fa0/24
2    VLAN0002                         active
3    VLAN0003                         active
5    VLAN0005                         active
10   VLAN0010                         active
1002 fddi-default                     active
1003 token-ring-default               active
1004 fddinet-default                  active
1005 trnet-default                    active
```

6. Now change the mode of 2nd switch to transparent and the 3rd to client mode.

```
ahsan(config)#vtp mode transparent
Setting device to VTP TRANSPARENT mode.
```

```
ahsan(config)#vtp mode client
Setting device to VTP CLIENT mode.
```

7. Create a VLAN 10 in the 1st switch (server mode switch) and then check it in the switch with transparent mode, the VLAN 10 does not exist satisfying that switch with this mode cannot synchronize VLAN information.

```
ahsan(config)#do show vlan brief

VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
1    default                          active    Fa0/3, Fa0/4, Fa0/5, Fa0/6
                                                Fa0/7, Fa0/8, Fa0/9, Fa0/10
                                                Fa0/11, Fa0/12, Fa0/13, Fa0/14
                                                Fa0/15, Fa0/16, Fa0/17, Fa0/18
                                                Fa0/19, Fa0/20, Fa0/21, Fa0/22
                                                Fa0/23, Fa0/24
2    VLAN0002                         active
3    VLAN0003                         active
5    VLAN0005                         active
1002 fddi-default                     active
1003 token-ring-default               active
1004 fddinet-default                  active
1005 trnet-default                    active
```

8. Now to study the property of switch in client mode, create a VLAN in this switch using command: '**vlan 11**', you can see the VLAN will not be created.

```
ahsan(config)#valn 11
              ^
% Invalid input detected at '^' marker.
```

EXERCISES

1. Why passwords for VTP should be same on all switches in a same VTP domain.

2. Write down the observations of a switch in VTP Transparent mode.

3. Write down a command which convert server mode switch into client mode

F/OBEM 01/05/00

NED University of Engineering & Technology
Department of Software Engineering
Course Code and Title: CS-351, Computer Communication Networks

Psychomotor Domain Assessment Rubric-Level P3					
Skill Sets	Extent of Achievement				
	1	2	3	4	5
Tool Utilization Effective **use** of software tools	Ineffective use of tools, significant misuse	Limited effective tool use, moderate misuse	Effective tool use with minor issues	Highly effective tool utilization	Mastery of tool utilization
Troubleshooting Issue **resolution** skills	Unable to troubleshoot effectively, significant issues persist	Limited troubleshooting abilities, moderate issues remain	Adequate troubleshooting skills, minor issues persist	Excellent troubleshooting skills, no issues remain	Mastery of troubleshooting
Task Accuracy ***Accuracy*** *in completing tasks*	Frequent errors and inaccuracies	Several errors, significant inaccuracies	Few errors, minor inaccuracies	No errors, highly accurate task completion	Mastery of accuracy
Task Completion Time Task completion **time management**	Tasks take significantly longer than expected	Tasks take longer than expected	Tasks completed within a reasonable timeframe	Tasks completed efficiently, ahead of schedule	Mastery of time management
Documentation Clarity and Organization ***Clarity. organization and structure*** *of documentation*	Poor, unclear, and highly disorganized documentation with no structure	Somewhat clear with a lack of organization and structure	Reasonably clear and organized with adequate structuring	Highly clear and highly organized and well-structured documentation	Mastery of documentation clarity, organization, and structure

Laboratory Session No. _____ Date: _____

Weighted CLO (Psychomotor Score)	
Remarks	
Instructor's Signature with Date:	

Lab Session 12

OBJECT

Recovering lost router password.

THEORY

In this lab you will learn the procedures required to recover a lost login or enable password. The procedures differ depending on the platform and the software used, but in all cases, password recovery requires that the router be taken out of operation and powered down. Note:

1. Please use cisco as the password where necessary.
2. Please be prepared to do password recovery right away. The group before you might have set a password other than cisco.
3. Use `show version` command to determine the platform before you try the password recovery.

You will be working with the configuration register as part of this lab. The config-register is a 16 bit register. Look up information about the config-register on documentation CD, CISCO web site, or any other resources available to you.

Software Configuration Register Bits (What do they mean)

Bit Number	Value	Meaning
0 to 3	0x000**0** to 0x000**F**	Boot field
6	0x0040 (setting bit 6 to 1)	Causes system software to ignore NVRAM contents
8	0x0100	Break disabled
13	0x2000	Boot default Flash software if network boot fails

Explanation of Boot Field

Boot Field	Meaning
0x0000	Stays at the system bootstrap prompt
0xXXX1	Boots the first system image in onboard Flash memory
0xXXX2 0xXXXF	If you set the boot field value to 0x2 through 0xF and there is a valid **boot system** command stored in the configuration file, the router boots the system software as directed by that value. If there is no boot system command, the router forms a default boot filename for booting from a network server. If there is no network server configured, as is the case in our lab, the standard setup dialogue is started.

PROCEDURE

Assume you have been locked out of the router. You have access only to the user mode. Follow the instructions below from the user mode. Do not get into privileged mode.

1. Type `show version` and record the value of the configuration register.

2. Using the power switch, turn off the router and then turn it on.

3. Press CTRL+Break on the terminal keyboard within 60 seconds of the powerup to put the router into ROMMON mode.

4. This is where the procedure differs depending on the platform.

   For 25XX and 4000:

- Type `o/r 0x2142` or `0x42` at the > prompt to boot from flash without loading the configuration.
- Type `i` or `reset` at the > prompt. The router reboots but ignores its saved configuration.

For 2600, 3600, 4500, 4700:

- Type `confreg 0x2142` at the rommon 1> prompt to boot from Flash without loading the configuration.
- Type `reset` at the rommon 2> prompt. The router reboots but ignores its saved configuration.

5. Type no after each setup question or press Ctrl-C to skip the initial setup procedure.

6. Type `enable` at the Router> prompt. You'll be in enable mode and see the Router# prompt.

7. Type `config mem` or `copy start running` to copy the nonvolatile RAM (NVRAM) into memory. **Do not type config term.**

8. Type `config term` and make the changes. The prompt is now hostname(config)#.

9. Type `enable password <password>` to set the password to the new value or issue the command `no enable password`.

10. Type `config-register 0x2102`, or the value you recorded in step 1.

11. Type `write mem` or `copy running startup` to commit the changes.

12. Type `show version` and observe the configuration register setting carefully.

EXERCISES

1. Explain the setting when the configuration-register is set to 0x2542.

2. There are many different ways to access a router. Write down these ways.

3. Explain the need for step 7 in password recovery procedure.

4. Write down the difference between "`enable password`" and "`enable secret password`."

5. What happens if "`enable password`" and "`enable secret password`" are the same?

6. When you configure enable password and issue the command show running, you can see the password set for the privileged mode. Is there a method to prevent it from being visible?

7. Set the configuration-register to 0x2542. Reload the router. Does the break sequence work? Crosscheck with configuration-register settings and see if it matches with the settings. Is there any difference? Explain

Lab Session 13

OBJECT

Introduction to Mininet

Overview

This lab provides an introduction to Mininet, a virtual testbed used for testing network tools and protocols. It demonstrates how to invoke Mininet from the command-line interface (CLI) utility and how to build and emulate topologies using a graphical user interface (GUI) application. In this lab we will use Containernet, a Mininet network emulator fork that allows the use of Docker containers as hosts in emulated network topologies. However, all the concepts covered are bounded to Mininet.

Objectives

By the end of this lab, you should be able to:

1. Understand what Mininet is and why it is useful for testing network topologies.
2. Invoke Mininet from the CLI.
3. Construct network topologies using the GUI.
4. Save/load Mininet topologies using the GUI.
5. Configure the interfaces of a router using the CLI.

Lab settings

The information in Table 1 provides the credentials of the machine containing Mininet.

Table 1**.** Credentials to access the Client machine.

Device	Account	Password
Client	admin	password

Lab roadmap

This lab is organized as follows:

1. Section 1: Introduction to Mininet.
2. Section 2: Invoke Mininet using the CLI.
3. Section 3: Build and emulate a network in Mininet using the GUI.
4. Section 4: Configure router r1.

Introduction to Mininet

Mininet is a virtual testbed enabling the development and testing of network tools and protocols. With a single command, Mininet can create a realistic virtual network on any type of machine (Virtual Machine (VM), cloud-hosted, or native). Therefore, it provides

an inexpensive solution and streamlined development running in line with production networks . Mininet offers the following features:

- Fast prototyping for new networking protocols.
- Simplified testing for complex topologies without the need of buying expensive hardware.
- Realistic execution as it runs real code on the Unix and Linux kernels.
- Open-source environment backed by a large community contributing extensive documentation.

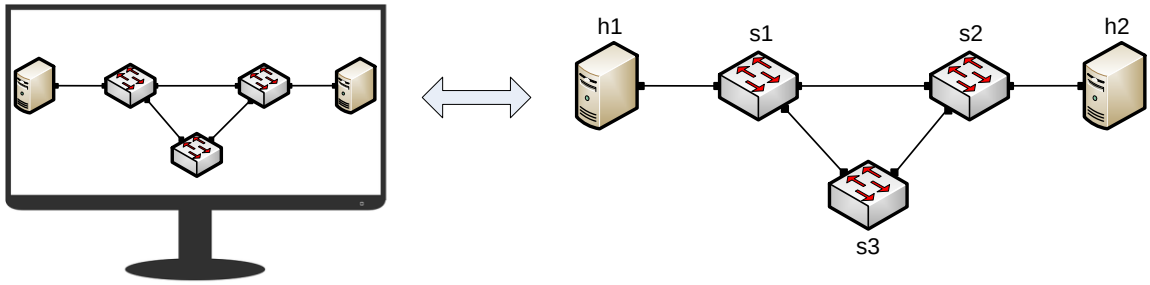


Figure 1. Hardware network vs. Mininet emulated network.

Mininet is useful for development, teaching, and research as it is easy to customize and interact with it through the CLI or the GUI. Mininet was originally designed to experiment with *OpenFlow* and *Software-Defined Networking (SDN)* . This lab, however, only focuses on emulating a simple network environment without SDN-based devices.

Mininet's logical nodes can be connected into networks. These nodes are sometimes called containers, or more accurately, *network namespaces*. Containers consume sufficiently fewer resources that networks of over a thousand nodes have created, running on a single laptop. A Mininet container is a process (or group of processes) that no longer has access to all the host system's native network interfaces. Containers are then assigned virtual Ethernet interfaces, which are connected to other containers through a virtual switch . Mininet connects a host and a switch using a virtual Ethernet (veth) link. The veth link is analogous to a wire connecting two virtual interfaces, as illustrated below.

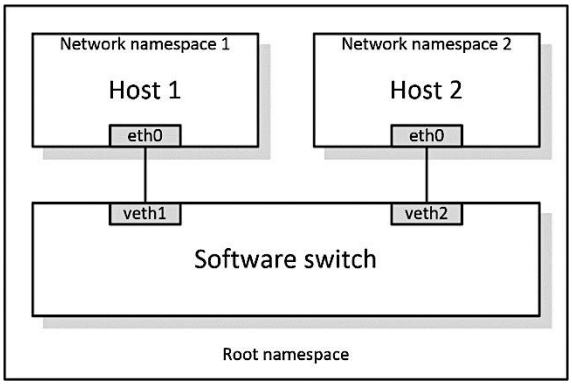


Figure 2. Network namespaces and virtual Ethernet links.

Each container is an independent network namespace, a lightweight virtualization feature that provides individual processes with separate network interfaces, routing tables, and Address Resolution Protocol (ARP) tables.

Mininet provides network emulation opposed to simulation, allowing all network software at any layer to be simply run *as is*; i.e. nodes run the native network software of the physical machine. On the other hand, in a simulated environment applications and protocol implementations need to be ported to run within the simulator before they can be used.

Invoke Mininet using the CLI

The first step to start Mininet using the CLI is to start a Linux terminal.

2.1 Invoke Mininet using the default topology

Step 1. Launch a Linux terminal by holding the `Ctrl+Alt+T` keys or by clicking on the Linux terminal icon.

Figure 3. Shortcut to open a Linux terminal.

The Linux terminal is a program that opens a window and permits you to interact with a command-line interface (CLI). A CLI is a program that takes commands from the keyboard and sends them to the operating system for execution.

Step 2. To start a minimal topology, enter the command shown below. When prompted for a password, type `password` and hit enter. Note that the password will not be visible as you type it.

```
sudo mn
```

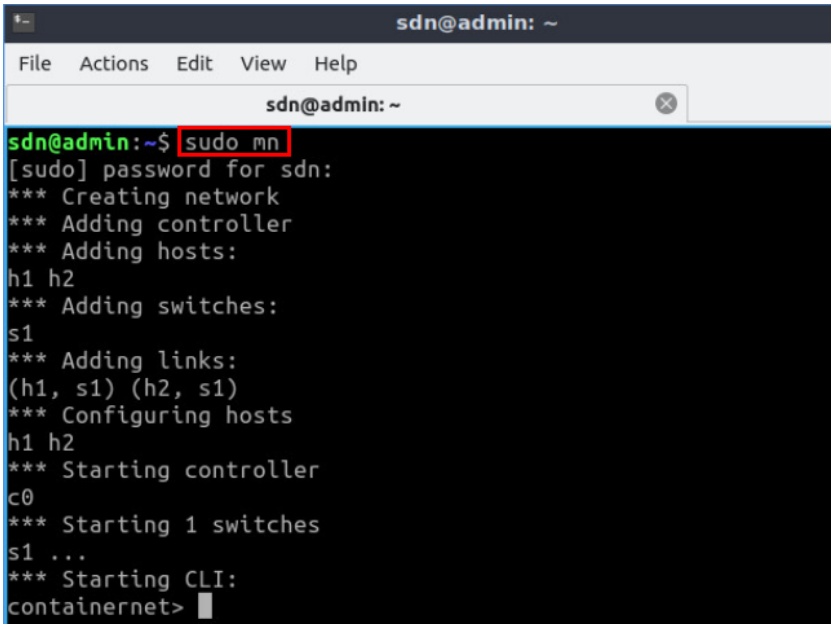


Figure 4. Starting Mininet using the CLI.

The above command starts Mininet with a minimal topology, which consists of a switch connected to two hosts as shown below. The loaded topology matches the figure below.

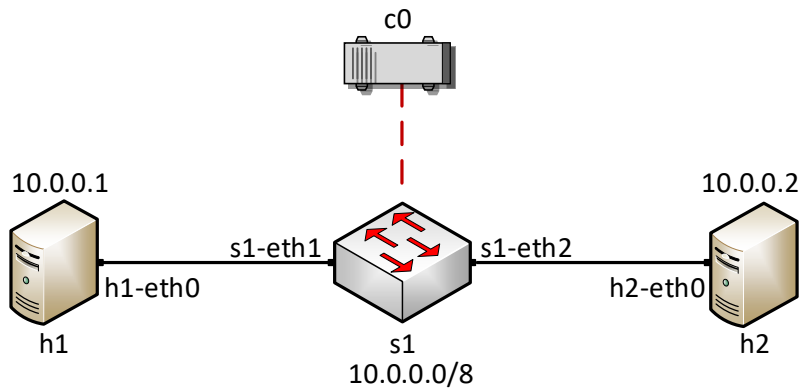


Figure 5. Mininet's default minimal topology.

When issuing the `sudo mn` command, Mininet initializes the topology and launches the containernet command line interface which looks like this:

```
containernet>
```

Step 3. To display the list of Mininet CLI commands and examples on their usage, type the following command.

```
help
```

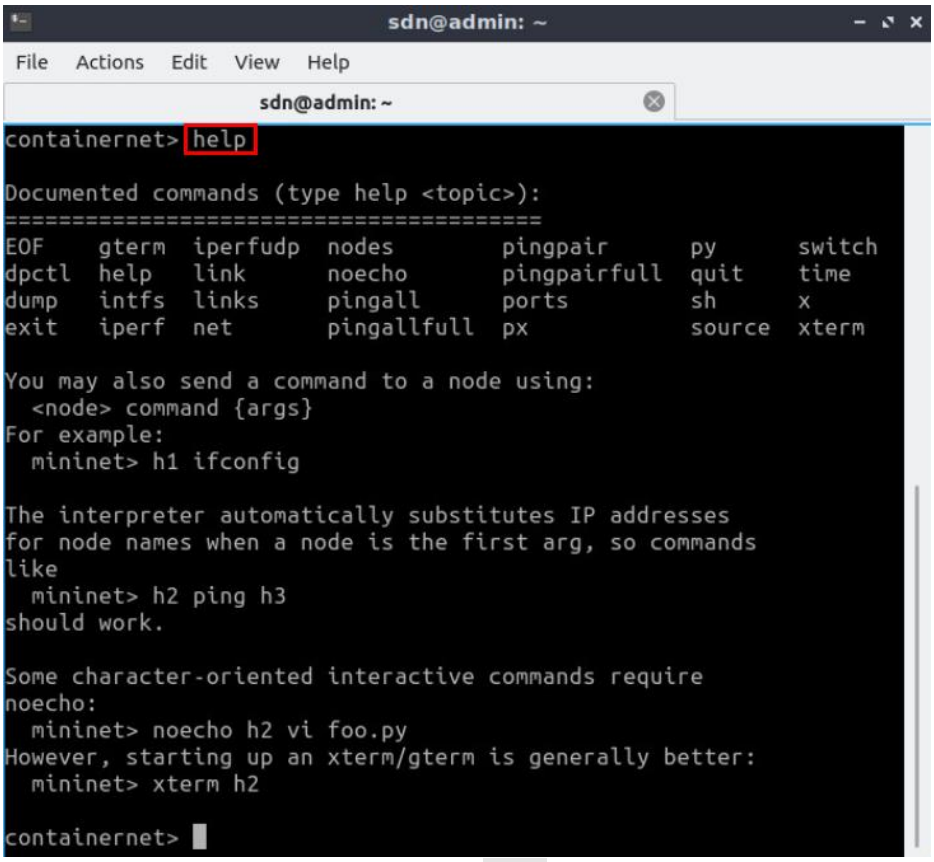


Figure 6. Mininet's `help` command.

Step 4. To display the available nodes, type the following command.

```
nodes
```

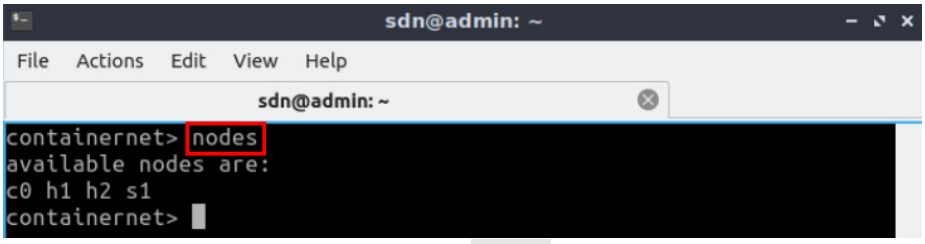


Figure 7. Mininet's `nodes` command.

The output of this command shows that there is a controller, two hosts (host h1 and host h2), and a switch (s1).

Step 5. It is useful sometimes to display the links between the devices in Mininet to understand the topology. Issue the command shown below to see the available links.

```
net
```

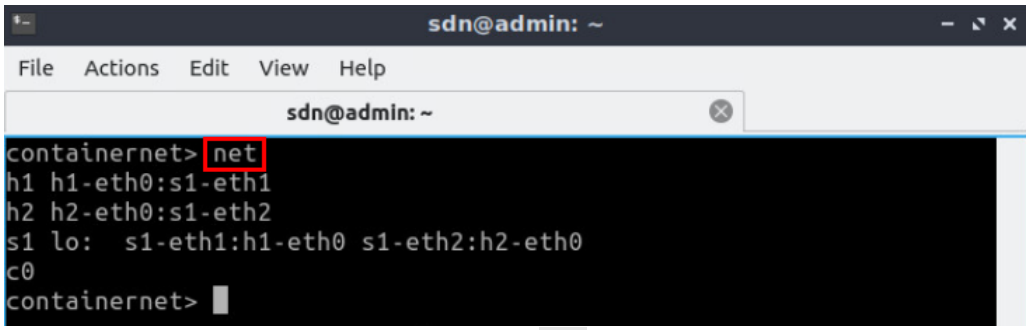


Figure 8. Mininet's `net` command.

The output of this command shows that:

1. Host h1 is connected using its network interface *h1-eth0* to the switch on interface *s1-eth1*.
2. Host h2 is connected using its network interface *h2-eth0* to the switch on interface *s1-eth2*.
3. Switch s1:
    a. has a loopback interface *lo*.
    b. connects to *h1-eth0* through interface *s1-eth1*.
    c. connects to *h2-eth0* through interface *s1-eth2*.
4. Controller c0 is the brain of the network, where it has a global knowledge about the network. A controller instructs the switches on how to forward/drop packets in the network.

Mininet allows you to execute commands on a specific device. To issue a command for a specific node, you must specify the device first, followed by the command.

Step 6. To proceed, issue the following command.

```
h1 ifconfig
```

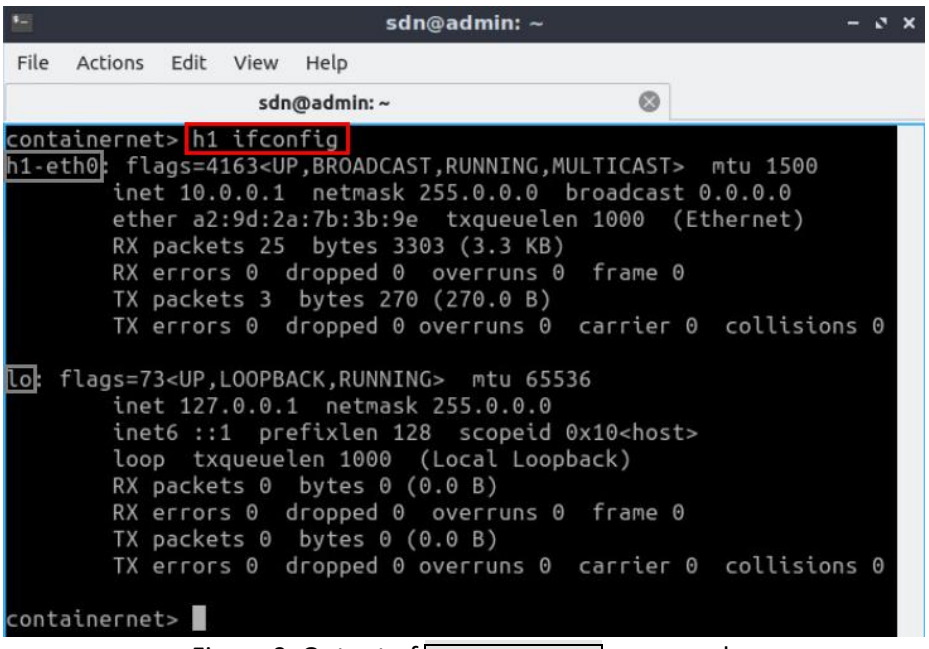


Figure 9. Output of `h1 ifconfig` command.

This command executes the `ifconfig` Linux command on host h1. The command shows host h1's interfaces. The display indicates that host h1 has an interface *h1-eth0* configured with IP address 10.0.0.1, and another interface lo, configured with IP address 127.0.0.1 (loopback interface).

Test connectivity

Mininet's default topology assigns the IP addresses 10.0.0.1/8 and 10.0.0.2/8 to host h1 and host h2 respectively. To test connectivity between them, you can use the command `ping`. The `ping` command operates by sending Internet Control Message Protocol (ICMP) Echo Request messages to the remote computer and waiting for a response or reply. Information available includes how many responses are returned and how long it takes for them to return.

Step 1. On the CLI, type the command shown below. This command tests the connectivity between host h1 and host h2.

```
h1 ping 10.0.0.2
```

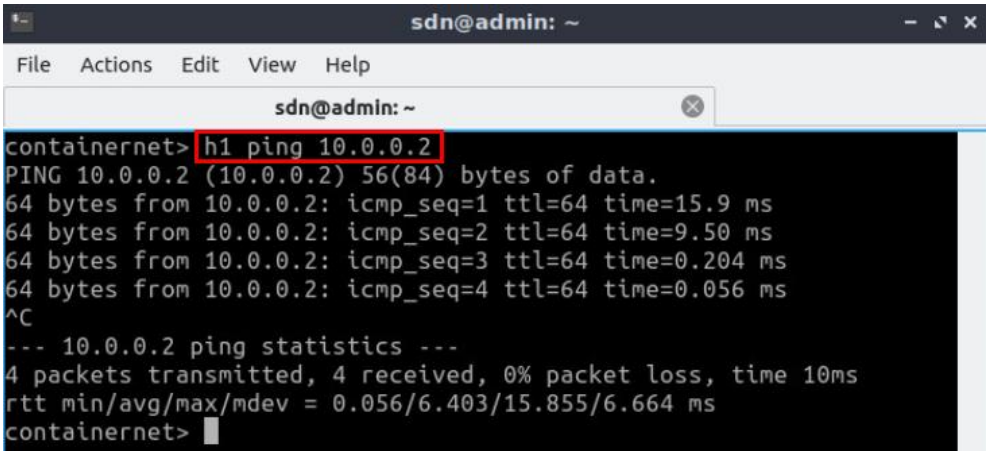


Figure 10. Connectivity test between host h1 and host h2.

To stop the test, press `Ctrl+c`. The figure above shows a successful connectivity test. Host h1 (10.0.0.1) sent four packets to host h2 (10.0.0.2) and successfully received the expected responses.

Step 2. Stop the emulation by typing the following command.

```
exit
```

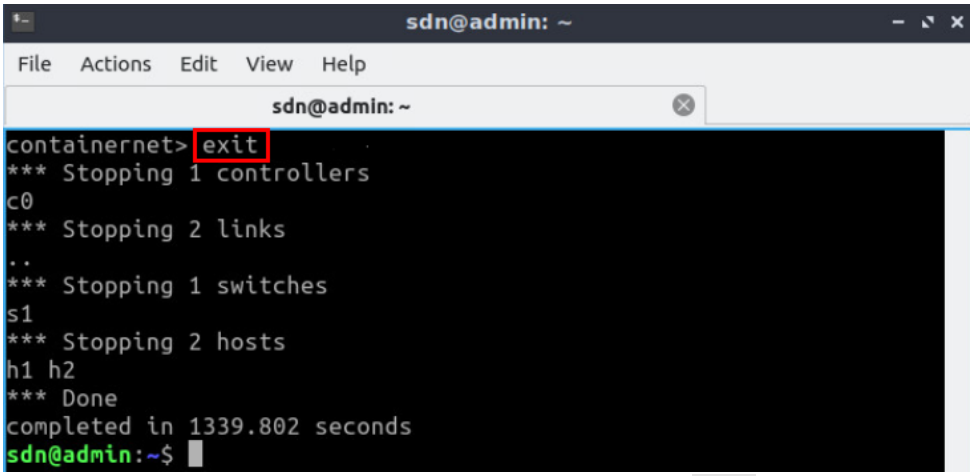


Figure 11. Stopping the emulation using `exit`.

The command `sudo mn -c` is often used on the Linux terminal (not on the Mininet CLI) to clean a previous instance of Mininet (e.g., after a crash).

Build and emulate a network in Mininet using the GUI

In this section, you will use the application MiniEdit[5] to deploy the topology illustrated below. MiniEdit is a simple GUI network editor for Mininet.

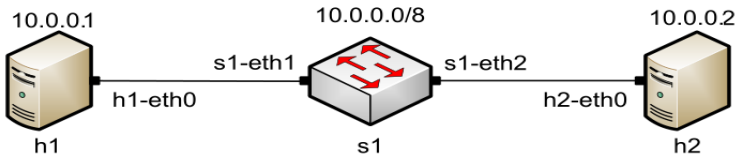


Figure 12. Lab topology.

Build the network topology

Step 1. A shortcut to MiniEdit is located on the machine's Desktop. Start MiniEdit by clicking on MiniEdit's shortcut. When prompted for a password, type `password`.

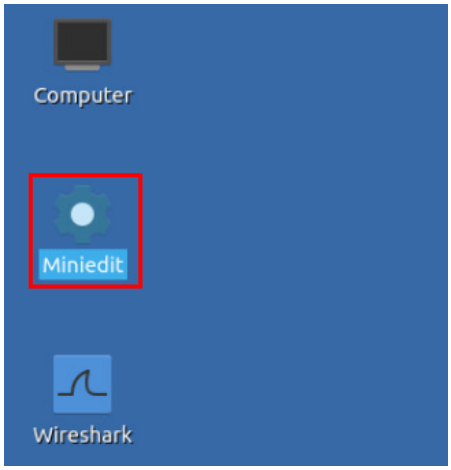


Figure 13. MiniEdit Desktop shortcut.

MiniEdit will start, as illustrated below.

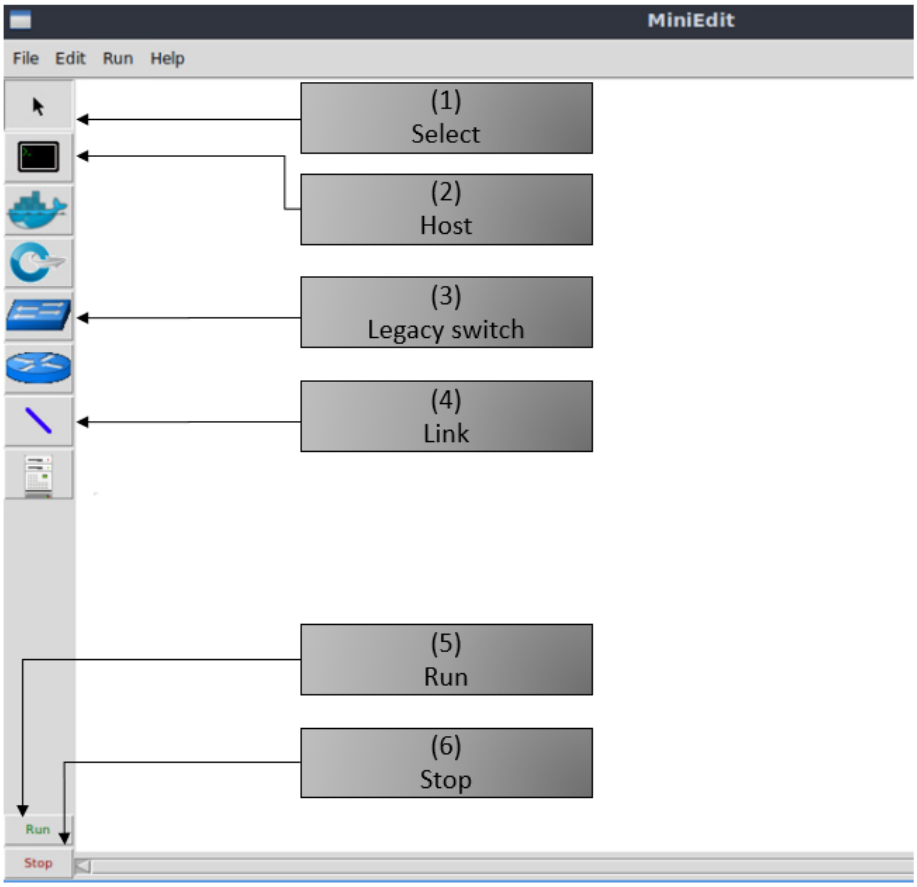


Figure 14. MiniEdit Graphical User Interface (GUI).

The main buttons in this lab are:

1. *Select*: allows selection/movement of the devices. Pressing *Del* on the keyboard after selecting the device removes it from the topology.
2. *Host*: allows addition of a new host to the topology. After clicking this button, click anywhere in the blank canvas to insert a new host.
3. *Legacy switch*: allows addition of a new legacy switch to the topology. After clicking this button, click anywhere in the blank canvas to insert the switch.
4. *Link*: connects devices in the topology (mainly switches and hosts). After clicking this button, click on a device and drag to the second device to which the link is to be established.
5. *Run*: starts the emulation. After designing and configuring the topology, click the run button.
6. *Stop*: stops the emulation.

Step 2. To build the topology illustrated in Figure 12, two hosts and one switch must be deployed. Deploy these devices in MiniEdit, as shown below.

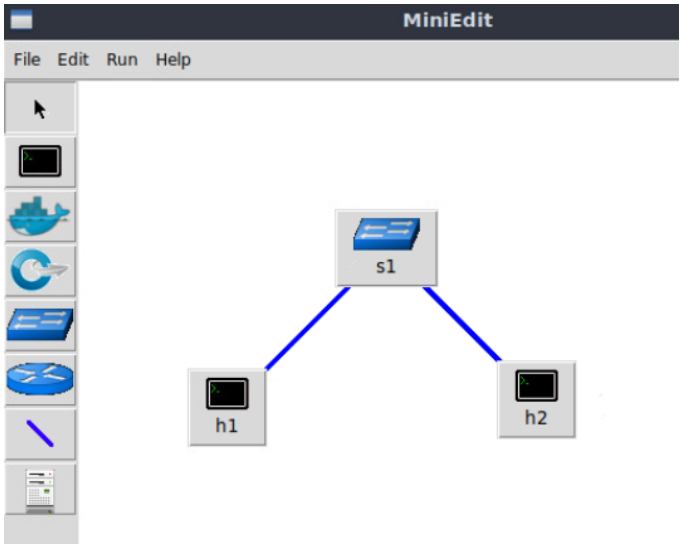


Figure 15. MiniEdit's topology.

Use the buttons described in the previous step to add and connect devices. The configuration of IP addresses is described in Step 3.

Step 3. Configure the IP addresses of host h1 and host h2. Host h1's IP address is 10.0.0.1/8 and host h2's IP address is 10.0.0.2/8. A host can be configured by holding the right click and selecting properties on the device. For example, host h2 is assigned the IP address 10.0.0.2/8 in the figure below. Click *OK* for the settings to be applied.

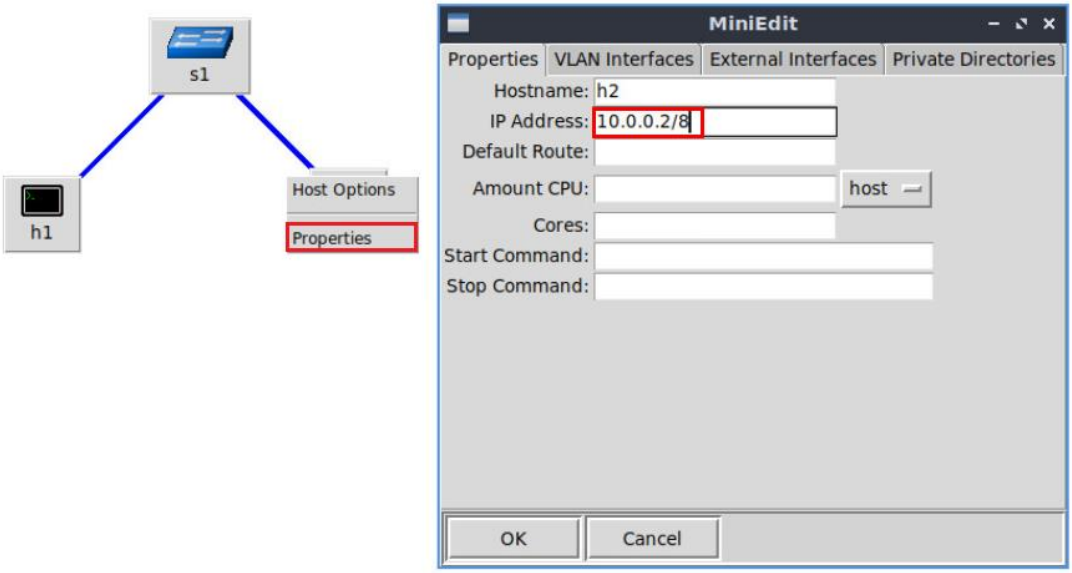


Figure 16. Configuration of a host's properties.

Test connectivity

Before testing the connection between host h1 and host h2, the emulation must be started.

Step 1. Click on the *Run* button to start the emulation. The emulation will start and the buttons of the MiniEdit panel will gray out, indicating that they are currently disabled.

Figure 17. Starting the emulation.

Step 2. Open a terminal on host h1 by holding the right click on host h1 and selecting *Terminal*. This opens a terminal on host h1 and allows the execution of commands on the host h1. Repeat the procedure on host h2.

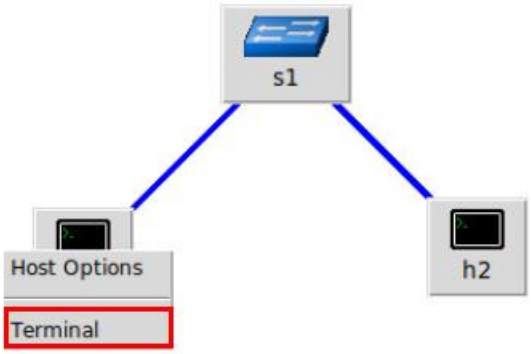


Figure 18. Opening a terminal on host h1.

The network and terminals at host h1 and host h2 will be available for testing.

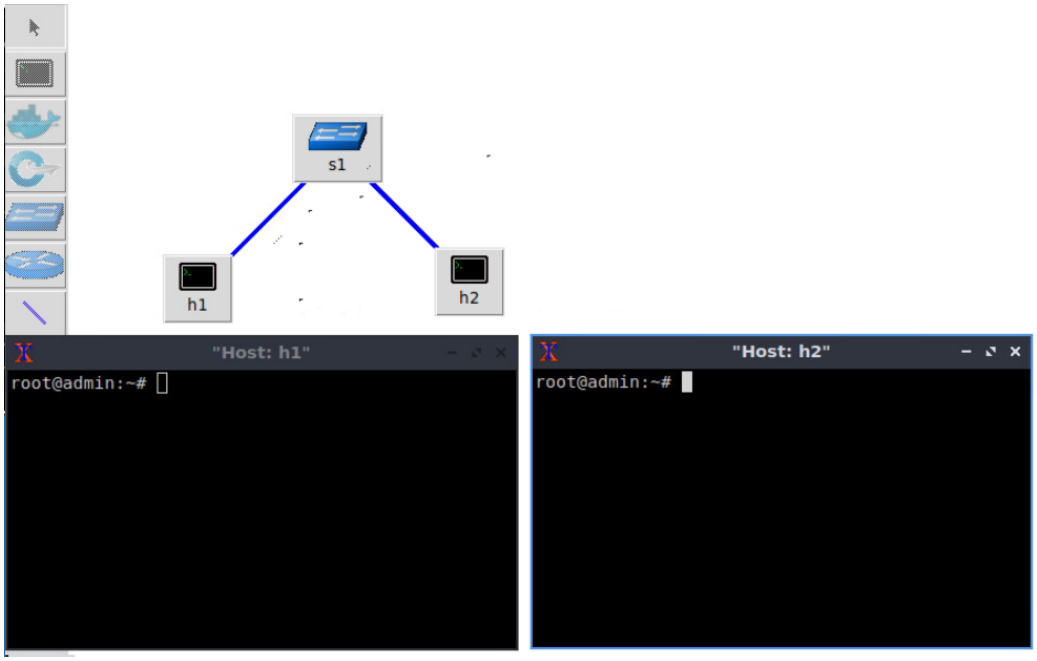


Figure 19. Terminals at host h1 and host h2.

Step 3. On host h1's terminal, type the command shown below to display its assigned IP addresses. The interface *h1-eth0* at host h1 should be configured with the IP address 10.0.0.1 and subnet mask 255.0.0.0.

```
ifconfig
```

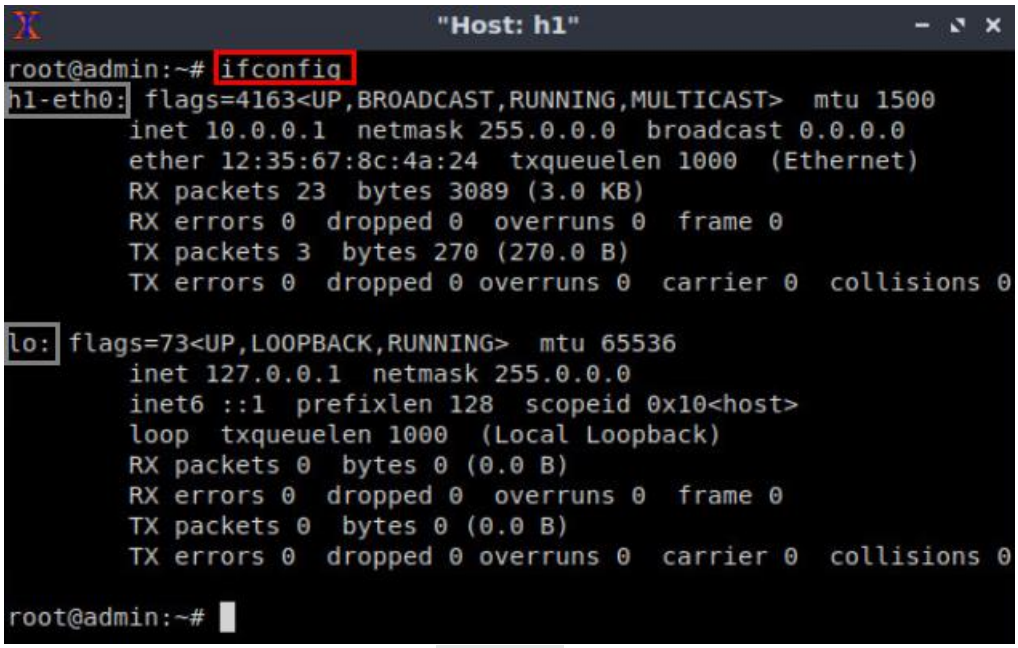


Figure 20. Output of `ifconfig` command on host h1.

Repeat step 3 on host h2. Its interface *h2-eth0* should be configured with IP address 10.0.0.2 and subnet mask 255.0.0.0.

Step 4. On host h1's terminal, type the command shown below. This command tests the connectivity between host h1 and host h2.

```
ping 10.0.0.2
```

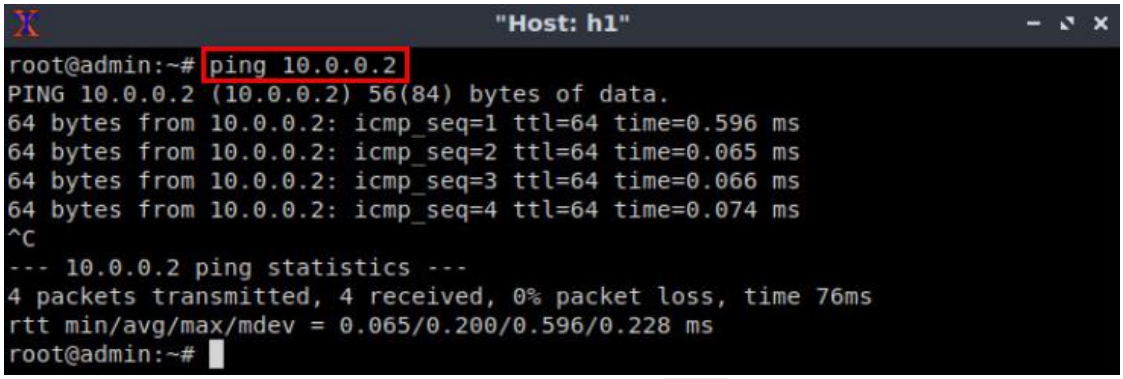


Figure 21. Connectivity test using `ping` command.

To stop the test, press `Ctrl+c`. The figure above shows a successful connectivity test. Host h1 (10.0.0.1) sent four packets to host h2 (10.0.0.2) and successfully received the expected responses.

Step 5. Stop the emulation by clicking on the *Stop* button.

Figure 22. Stopping the emulation.

Automatic assignment of IP addresses

In the previous section, you manually assigned IP addresses to host h1 and host h2. An alternative is to rely on Mininet for an automatic assignment of IP addresses (by default, Mininet uses automatic assignment), which is described in this section.

Step 1. Remove the manually assigned IP address from host h1. Hold right-click on host h1, *Properties*. Delete the IP address, leaving it unassigned, and press the *OK* button as shown below. Repeat the procedure on host h2.

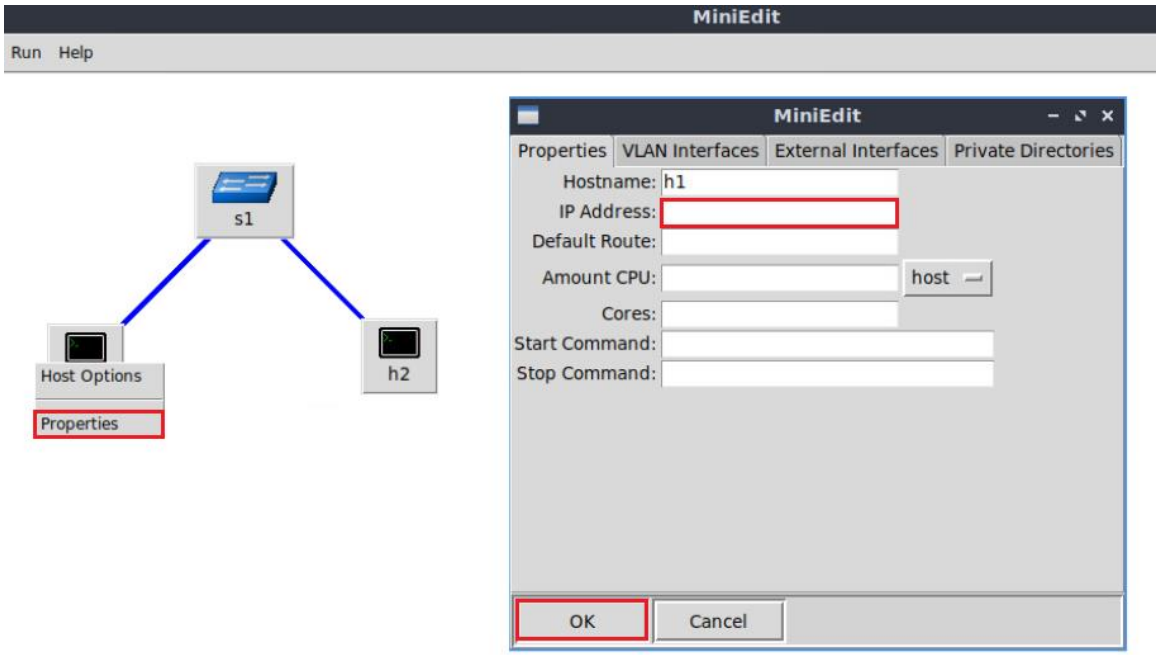


Figure 23. Host h1 properties.

Step 2. Click on *Edit*, *Preferences* button. The default IP base is 10.0.0.0/8. Modify this value to 15.0.0.0/8, and then press the *OK* button.

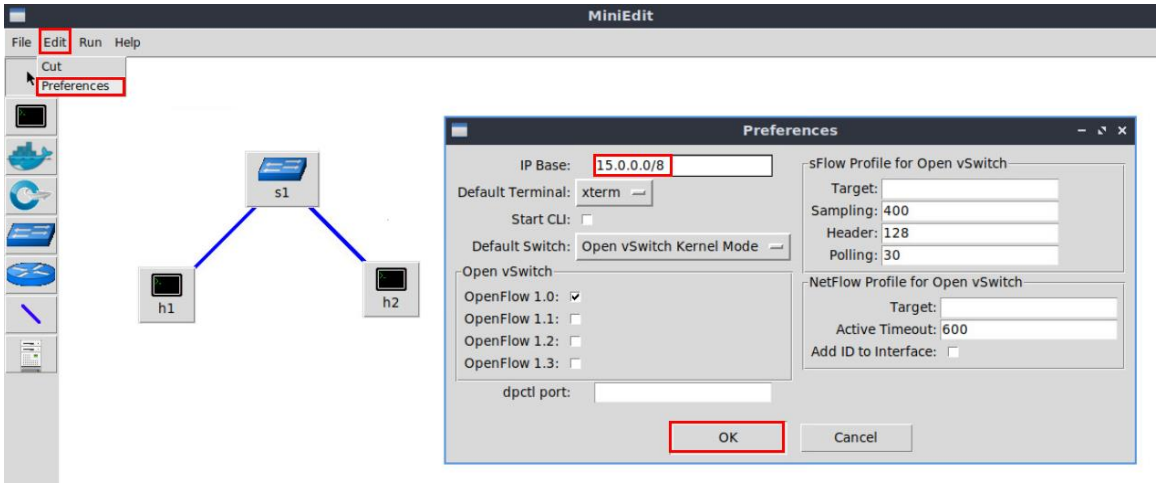


Figure 24. Modification of the IP Base (network address and prefix length).

Step 3. Run the emulation again by clicking on the *Run* button. The emulation will start and the buttons of the MiniEdit panel will be disabled.

Step 4. Open a terminal on host h1 by holding the right click on host h1 and selecting Terminal.

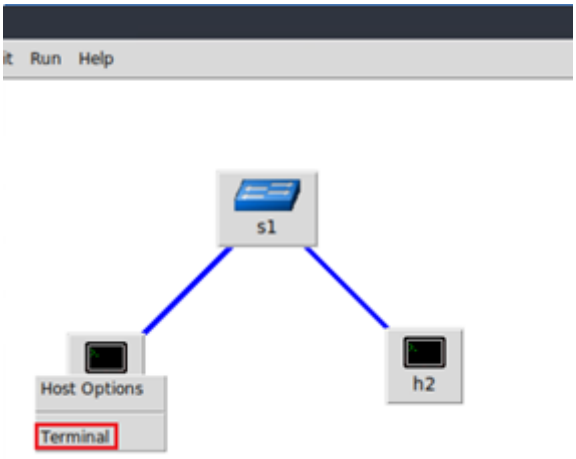


Figure 25. Opening a terminal on host h1.

Step 5. Type the command shown below to display the IP addresses assigned to host h1. The interface *h1-eth0* at host h1 now has the IP address 15.0.0.1 and subnet mask 255.0.0.0.

```
ifconfig
```

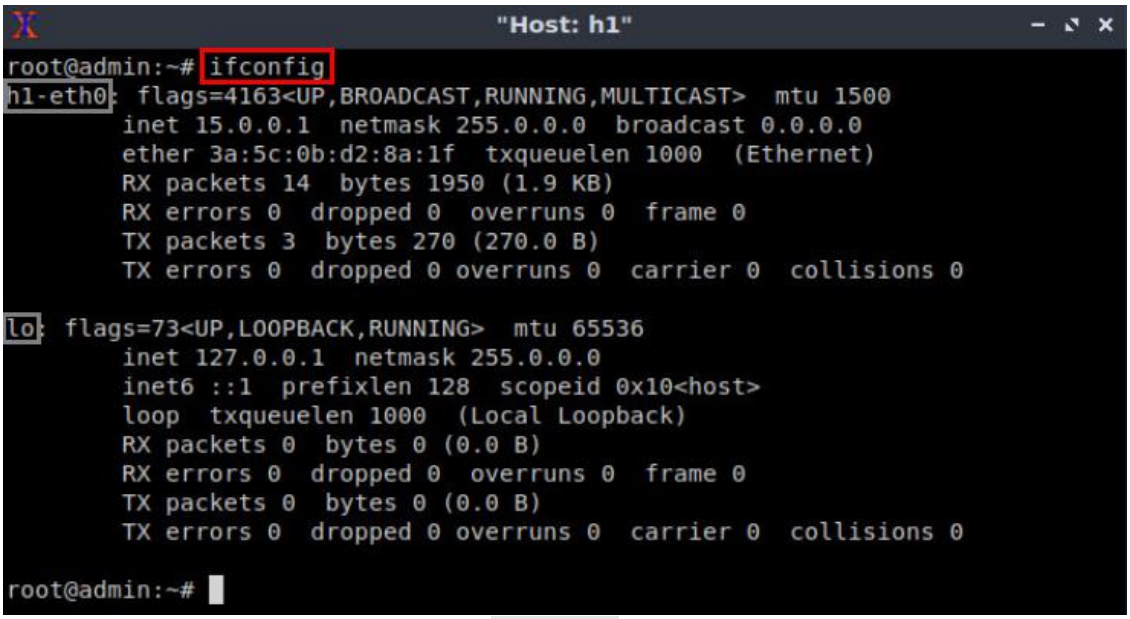


Figure 26. Output of `ifconfig` command on host h1.

You can also verify the IP address assigned to host h2 by repeating Steps 4 and 5 on host h2's terminal. The corresponding interface *h2-eth0* at host h2 has now the IP address 15.0.0.2 and subnet mask 255.0.0.0.

Step 6. Stop the emulation by clicking on *Stop* button.

Figure 27. Stopping the emulation.

Save and load a Mininet topology

In this section you will save and load a Mininet topology. It is often useful to save the network topology, particularly when its complexity increases. MiniEdit enables you to save the topology to a file.

Step 1. Save the current topology by clicking on *File* then *Save*. Provide a name for the topology and save it in the local folder. In this case, we used *myTopology* as the topology name.

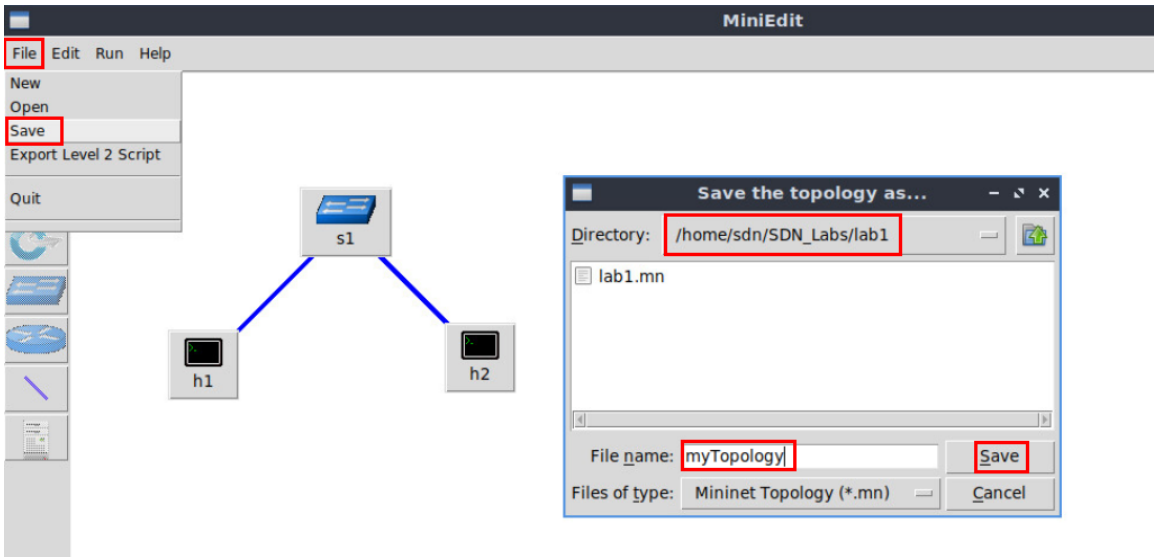


Figure 28. Saving the topology.

Step 2. Load the topology by clicking on *File* then *Open*. Search for the topology file called *lab1.mn* and click on *Open*. A new topology will be loaded to MiniEdit.

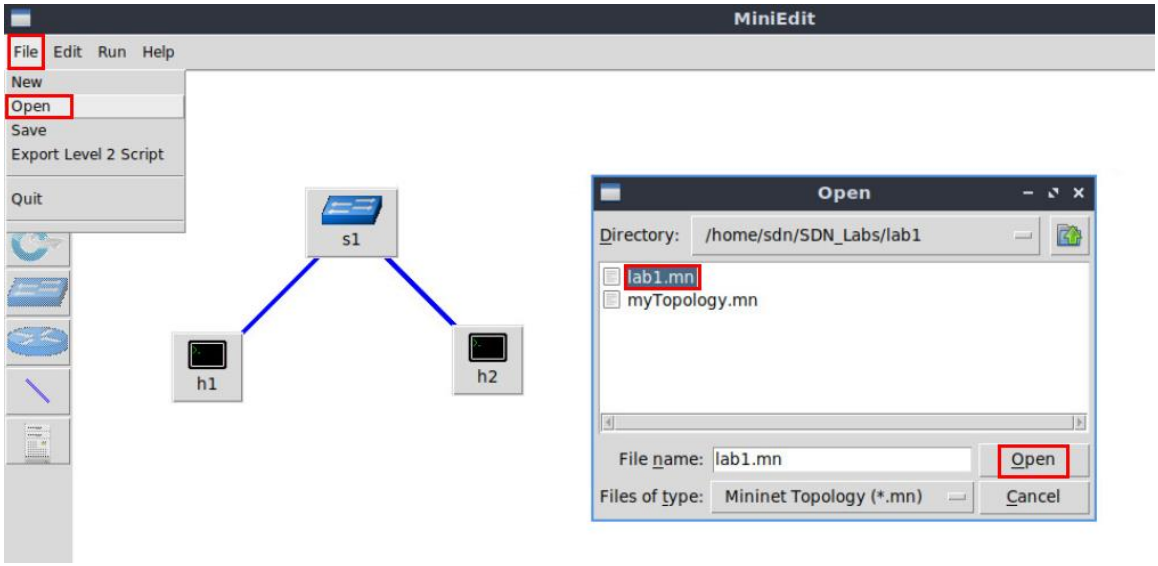


Figure 29. Opening a topology.

Configure router r1

In the previous step, you loaded a topology that consists of two networks directly connected to router r1. Consider Figure 30. In this topology two LANs, defined by switch s1 and switch s2 are connected to router r1. Initially, host h1 and host h2 do not have connectivity thus, you will configure router r1's interfaces in order to establish connectivity between the two networks.

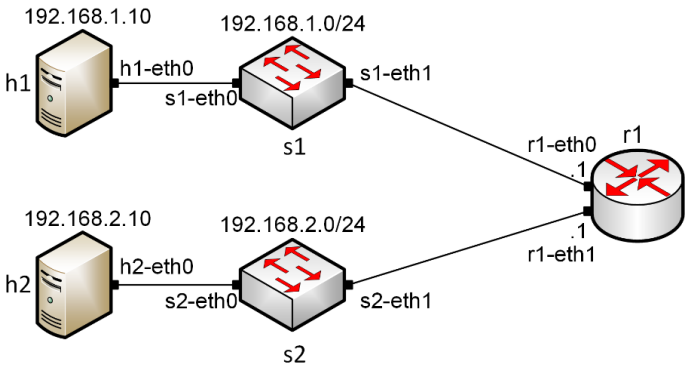


Figure 30. Topology.

Table 2 summarized the IP addresses used to configure router r1 and the end-hosts.

Table 2. Topology information.

Device	Interface	IP Address	Subnet	Default gateway
r1	r1-eth0	192.168.1.1	/24	N/A
	r1-eth1	192.168.2.1	/24	N/A
h1	h1-eth0	192.168.1.10	/24	192.168.1.1
h2	h2-eth0	192.168.2.10	/24	192.168.2.1

Step 1. Click on the *Run* button to start the emulation. The emulation will start and the buttons of the MiniEdit panel will gray out, indicating that they are currently disabled.

Figure 31. Starting the emulation.

Verify end-hosts configuration

In this section, you will verify that the IP addresses are assigned according to Table 2. Additionally, you will check routing information.

Step 1. Hold right-click on host h1 and select *Terminal*. This opens the terminal of host h1 and allows the execution of commands on that host.

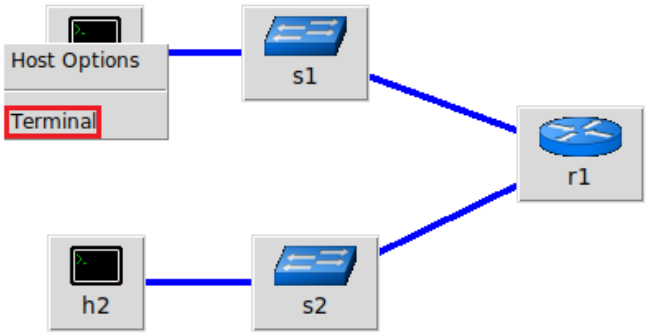


Figure 32. Opening a terminal on host h1.

Step 2. On host h1 terminal, type the command shown below to verify that the IP address was assigned successfully. You will verify that host h1 has two interfaces, *h1-eth0* configured with the IP address 192.168.1.10 and the subnet mask 255.255.255.0 and, the loopback interface *lo* configured with the IP address 127.0.0.1.

```
ifconfig
```

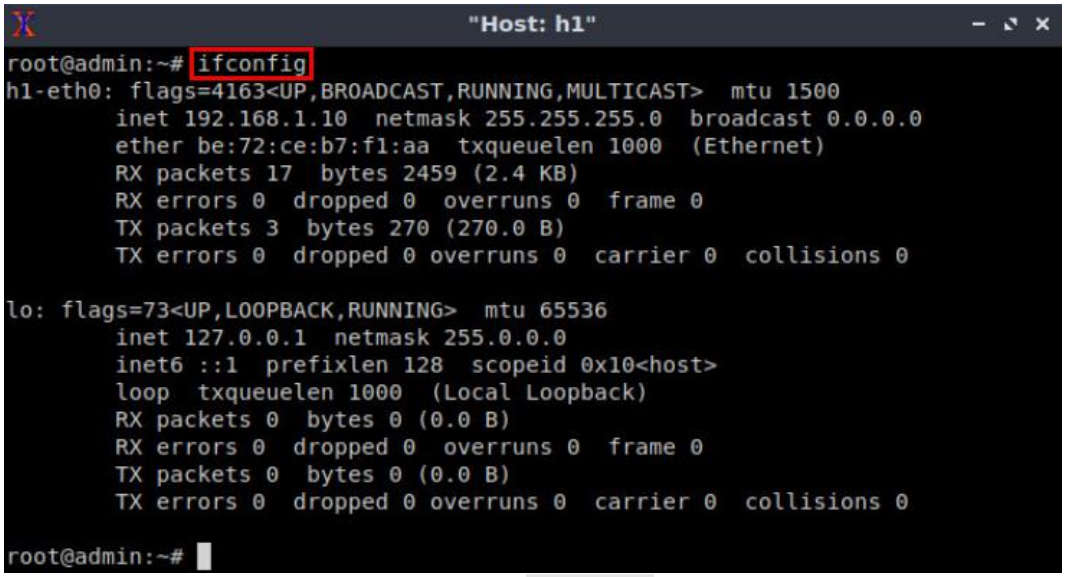


Figure 33. Output of `ifconfig` command.

Step 3. On host h1 terminal, type the command shown below to verify that the default gateway IP address is 192.168.1.1.

```
route
```

Figure 34. Output of `route` command.

Step 4. In order to verify host 2 default route, proceed similarly by repeating from step 1 to step 3 on host h2 terminal. Similar results should be observed.

Configure router's interface

Step 1. In order to configure router r1, hold right-click on router r1 and select *Terminal*.

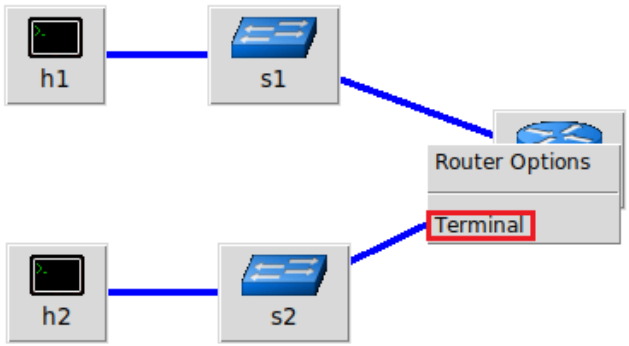


Figure 35. Opening a terminal on router r1.

Step 2. In this step, you will start the zebra daemon, a multi-server routing software that provides TCP/IP based routing protocols. The configuration will not be working if you do not enable the zebra daemon initially. To start zebra, type the following command.

```
zebra
```

Figure 36. Starting zebra daemon.

Step 3. After initializing zebra, vtysh should be started in order to provide all the CLI commands defined by the daemons. To proceed, issue the following command.

```
vtysh
```

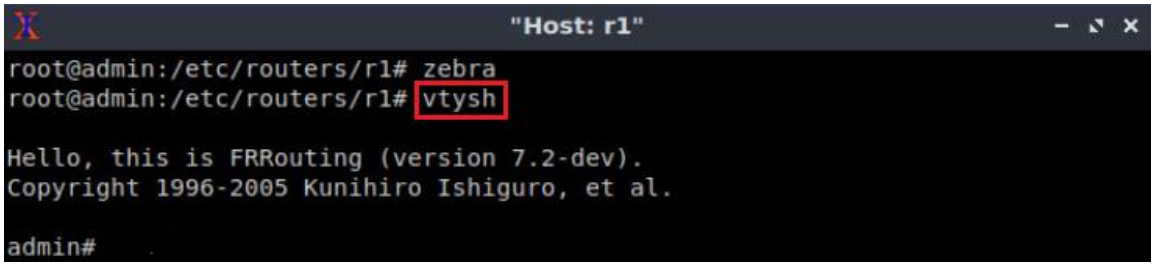


Figure 37. Starting vtysh on router r1.

Step 4. Type the following command in the router r1 terminal to enter in configuration mode.

```
configure terminal
```

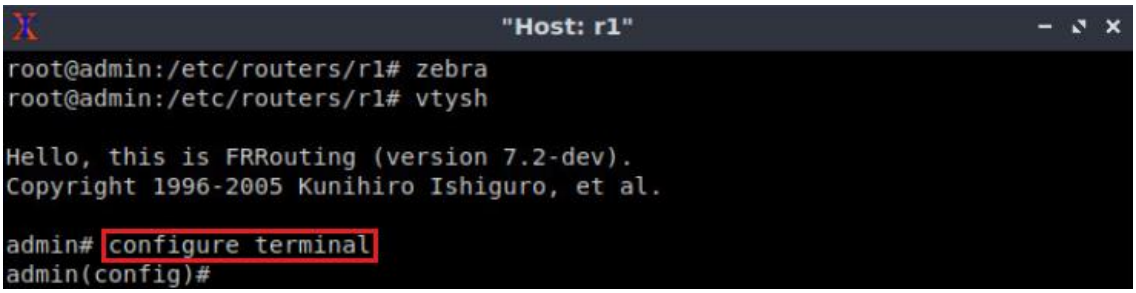


Figure 38. Entering in configuration mode.

Step 5. Type the following command in the router r1 terminal to configure interface *r1-eth0*.

```
interface r1-eth0
```

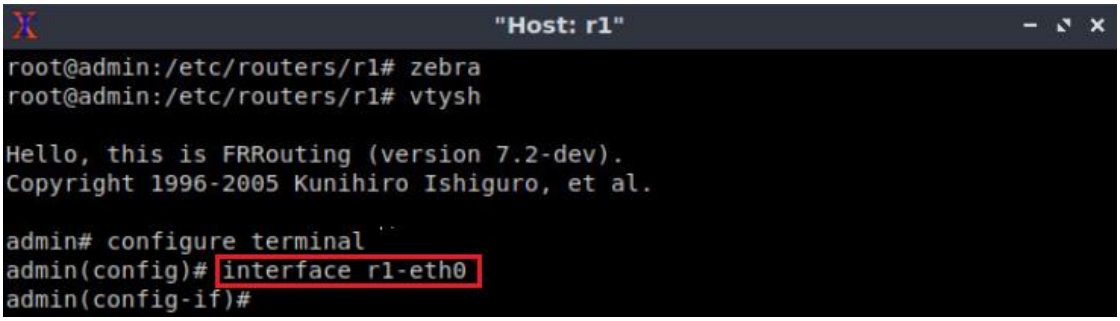


Figure 39. Configuring interface *r1-eth0*.

Step 6. Type the following command on router r1 terminal to configure the IP address of the interface *r1-eth0*.

```
ip address 192.168.1.1/24
```

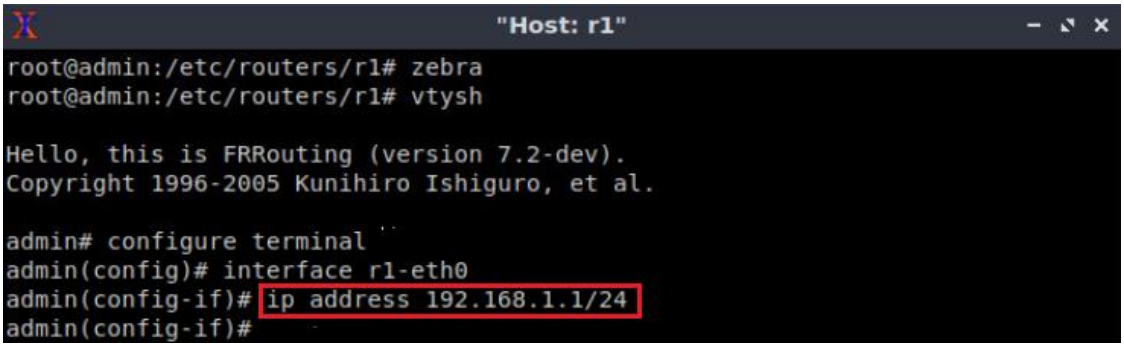


Figure 40. Configuring an IP address to interface *r1-eth0*.

Step 7. Type the following command exit from interface *r1-eth0* configuration.

```
exit
```

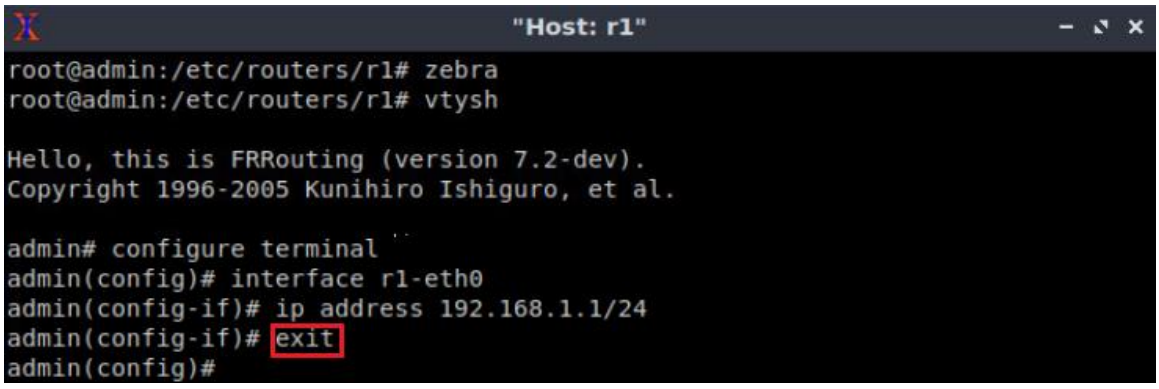


Figure 41. Exiting from configuring interface *r1-eth0*.

Step 8. Type the following command on router r1 terminal to configure the interface *r1-eth1*.

```
interface r1-eth1
```

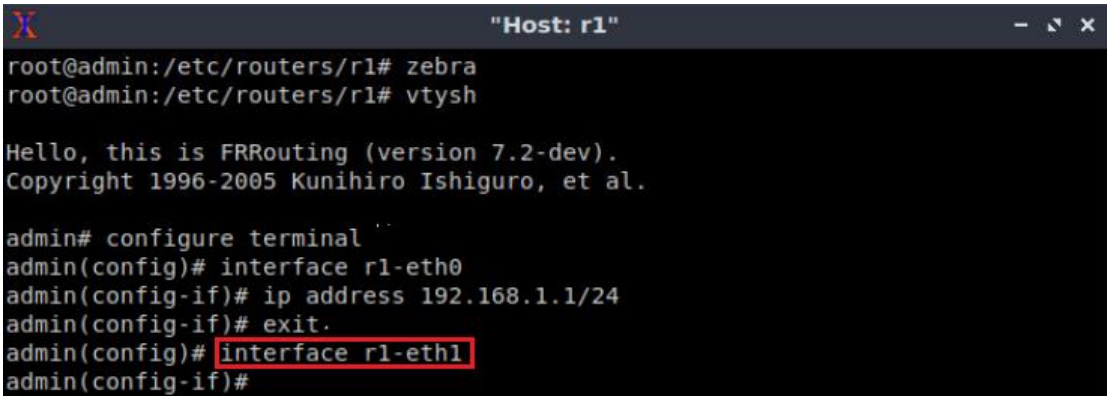


Figure 42. Configuring interface *r1-eth1*.

Step 9. Type the following command on router r1 terminal to configure the IP address of the interface *r1-eth1*.

```
ip address 192.168.2.1/24
```

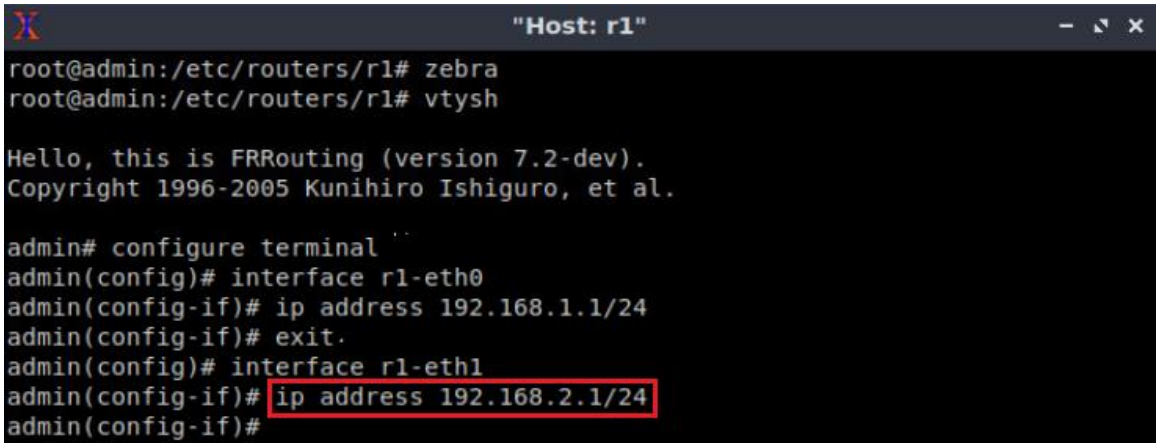


Figure 43. Configuring an IP address to interface *r1-eth1*.

Step 10. Type the following command to exit from *r1-eth1* interface configuration.

```
exit
```

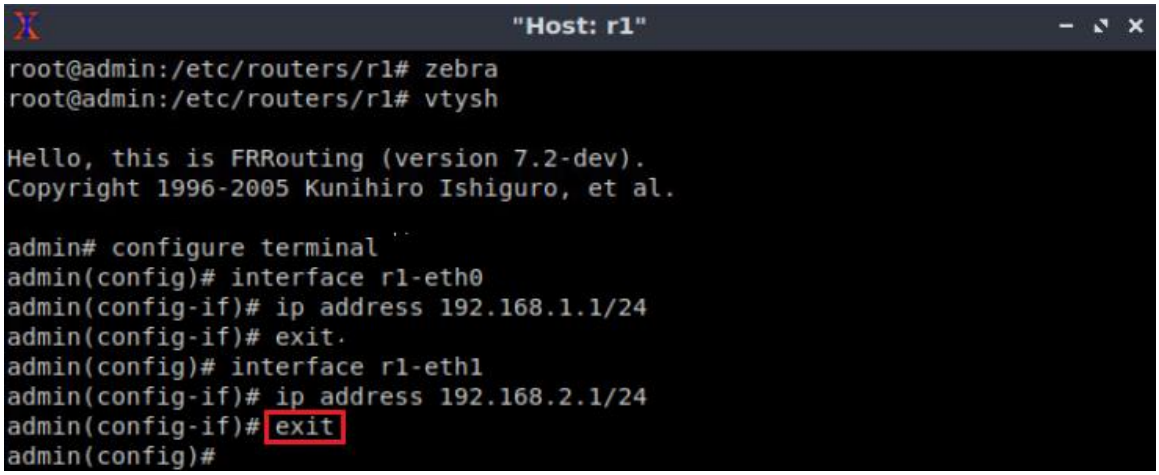


Figure 44. Exiting from configuring interface *r1-eth1*.

Verify router r1 configuration

Step 1. Exit from router r1 configuration mode issuing the following command.

```
exit
```

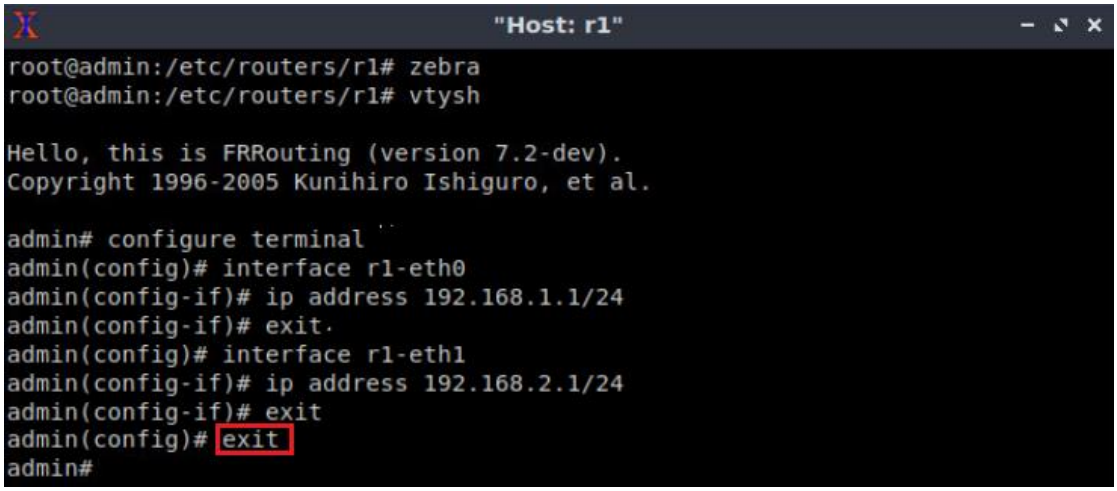


Figure 45. Exiting from configuration mode.

Step 2. Type the following command on router r1 terminal to verify the routing information of router r1. It will be showing all the directly connected networks.

```
show ip route
```

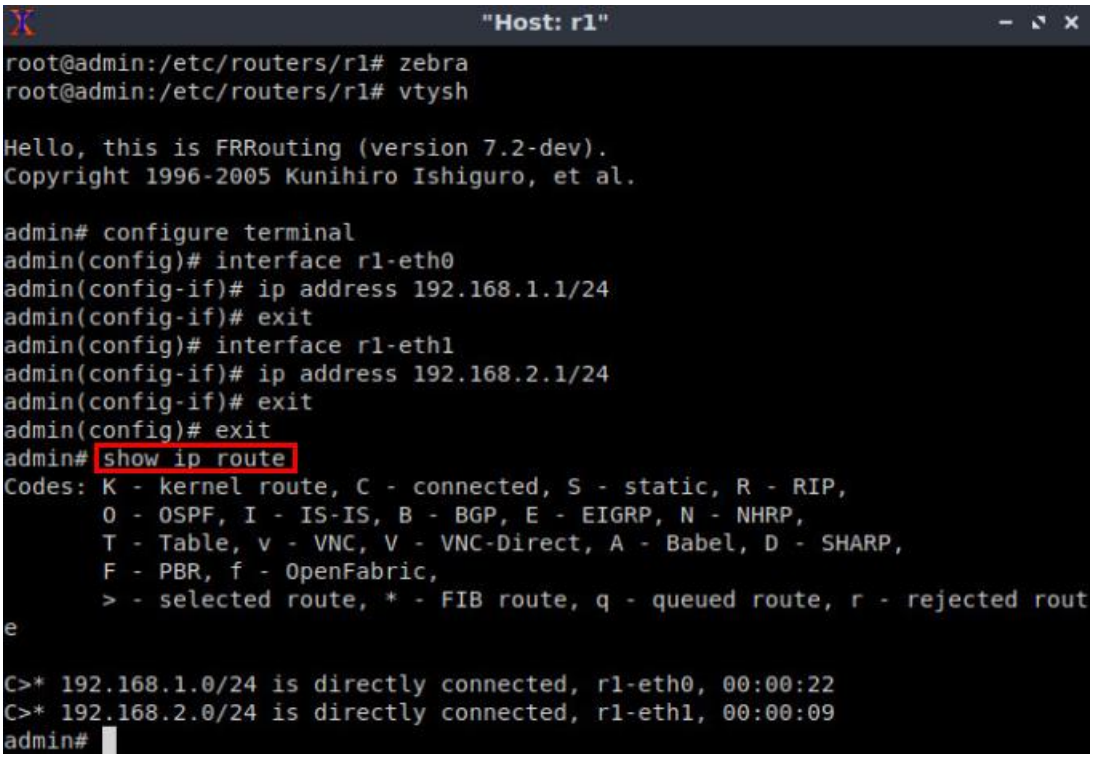


Figure 46. Displaying routing information of router r1.

Test connectivity between end-hosts

In this section you will run a connectivity test between host h1 and host h2.

Step 1. On host h1 terminal type the command shown below. Notice that according to Table 2, the IP address 192.168.2.10 is assigned to host h2. To stop the test press `ctrl+c`

```
ping 192.168.2.10
```

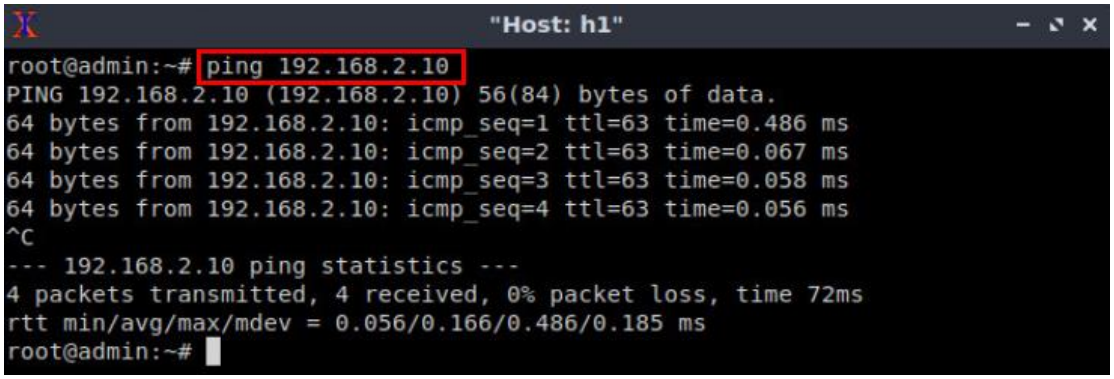


Figure 47. Connectivity test between host h1 and host h2.

This concludes Lab 1. Stop the emulation and then exit out of MiniEdit and Linux terminal.

EXERCISES

1. Construct simple network using mininet and its native SDN controller. All nodes should be able to communicate with each other. Use ICMP protocol for the verification of network connectivity. Also attach the snapshots of network and results of ping command.

Lab Session 14

Object

Introduction to ONOS SDN Controller

Pre-requisites

You will need a computer with at least 8GB of RAM and at least 20GB of free hard disk space. A faster processor or solid-state drive will speed up the virtual machine boot time, and a larger screen will help to manage multiple terminal windows.

The computer can run Windows, Mac OS X, or Linux – all work fine with VirtualBox, the only software requirement.
To install VirtualBox, you will need administrative access to the machine.

Set up your environment

Download and install required software

You will need to acquire two files: a VirtualBox installer and the ONOS tutorial OVA (for version 1.15.0).

*(Here are some **slides** that can be used to accompany the tutorial: **PDF** , **HTML**)*

After you have downloaded VirtualBox, install it, then go to the next section to verify that the VM is working on your system.

Create Virtual Machine

Double-click on the downloaded ONOS tutorial OVA file. This will open virtual box with an import dialog. Allocate 2-3 CPUs and 4-8GB of RAM for the VM.

47

Click on import. When the import is finished start the VM and log in as ***SDN User*** (***sdn***) with password ***rocks***

Important Command Prompt Notes

In this tutorial, commands are shown along with a command prompt to indicate the subsystem for which they are intended.

For example,

```
onos>
```

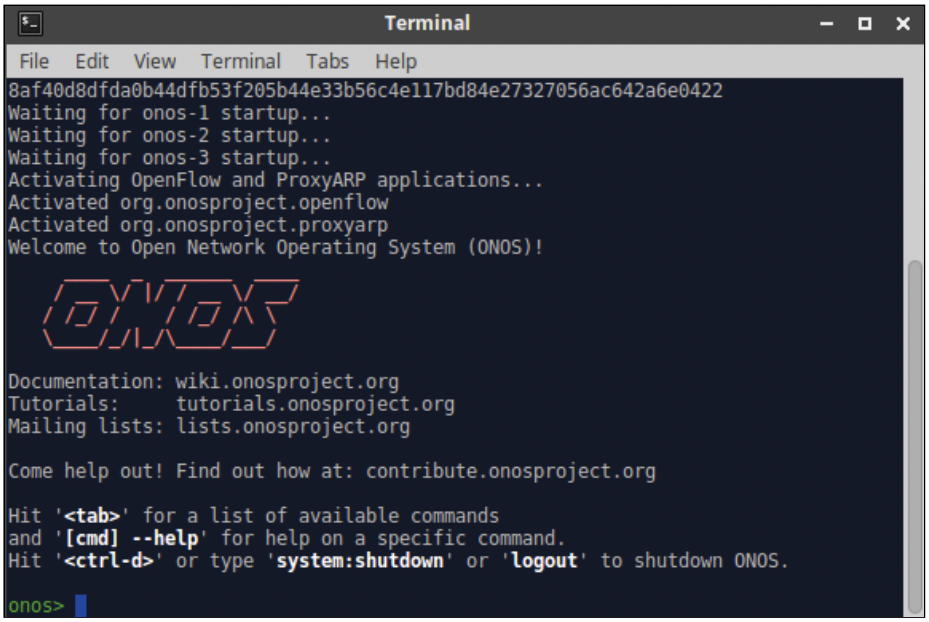
indicates that you are in the ONOS command line, whereas

```
mininet>
```

indicates that you are in mininet.

Setup ONOS Cluster

We have provided a simple mechanism which allows you to setup (or reset) the tutorial from scratch. Simply, click on the *Setup ONOS Cluster* icon on your desktop and this will reset ONOS cluster to its initial state. It'll take a few seconds for ONOS cluster be formed. During that time you may not be able to launch the ONOS CLI. Double click the Setup ONOS Cluster icon now and wait for ONOS to start-up. When ready, you should see the following:



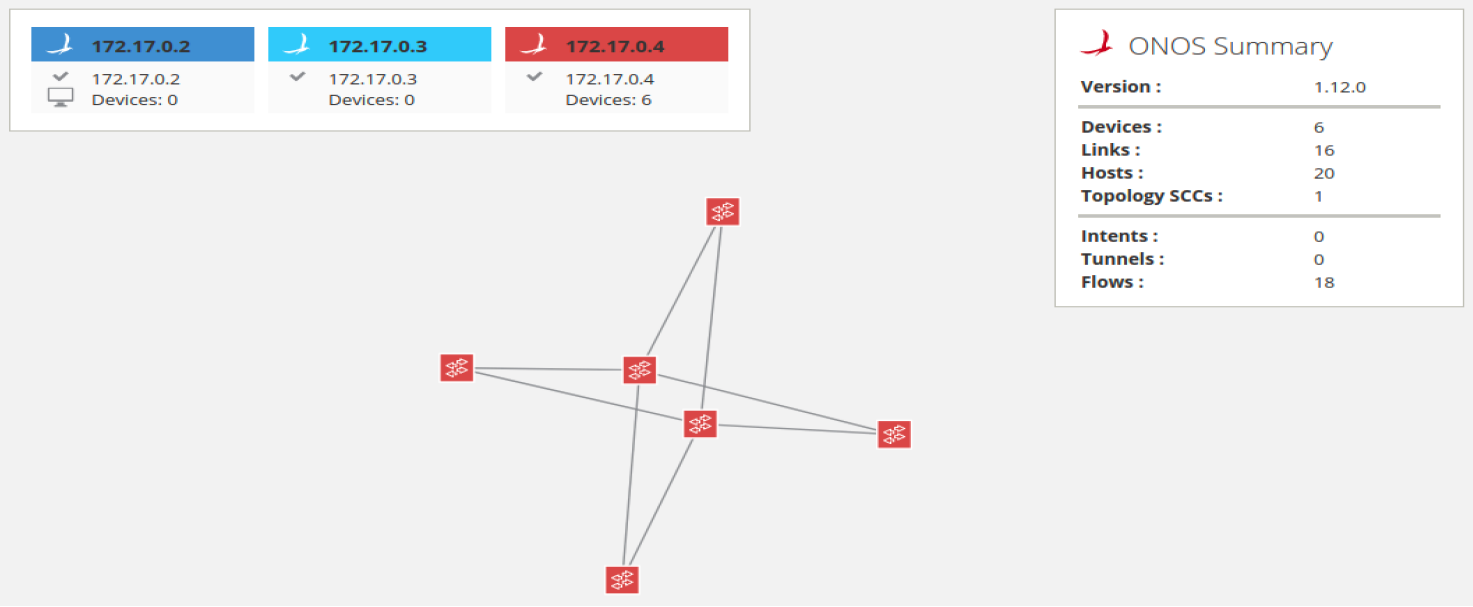
Launch ONOS GUI

ONOS has a web-based GUI which you can launch by clicking the provided *ONOS GUI* icon. Login as user ***onos*** with password ***rocks***

Start Mininet

Though the tutorial VM provides a few sample topologies, we'll be using the same spine-leaf physical topology for all exercises. The network is comprised of two spine switches and four leaf switches with five hosts connected to each leaf. To start mininet with this topology, simply double click on the *Spine Leaf Topology* icon on your desktop. When ready to exit mininet, not now however, type *Ctrl-D* or *exit* in the mininet prompt.

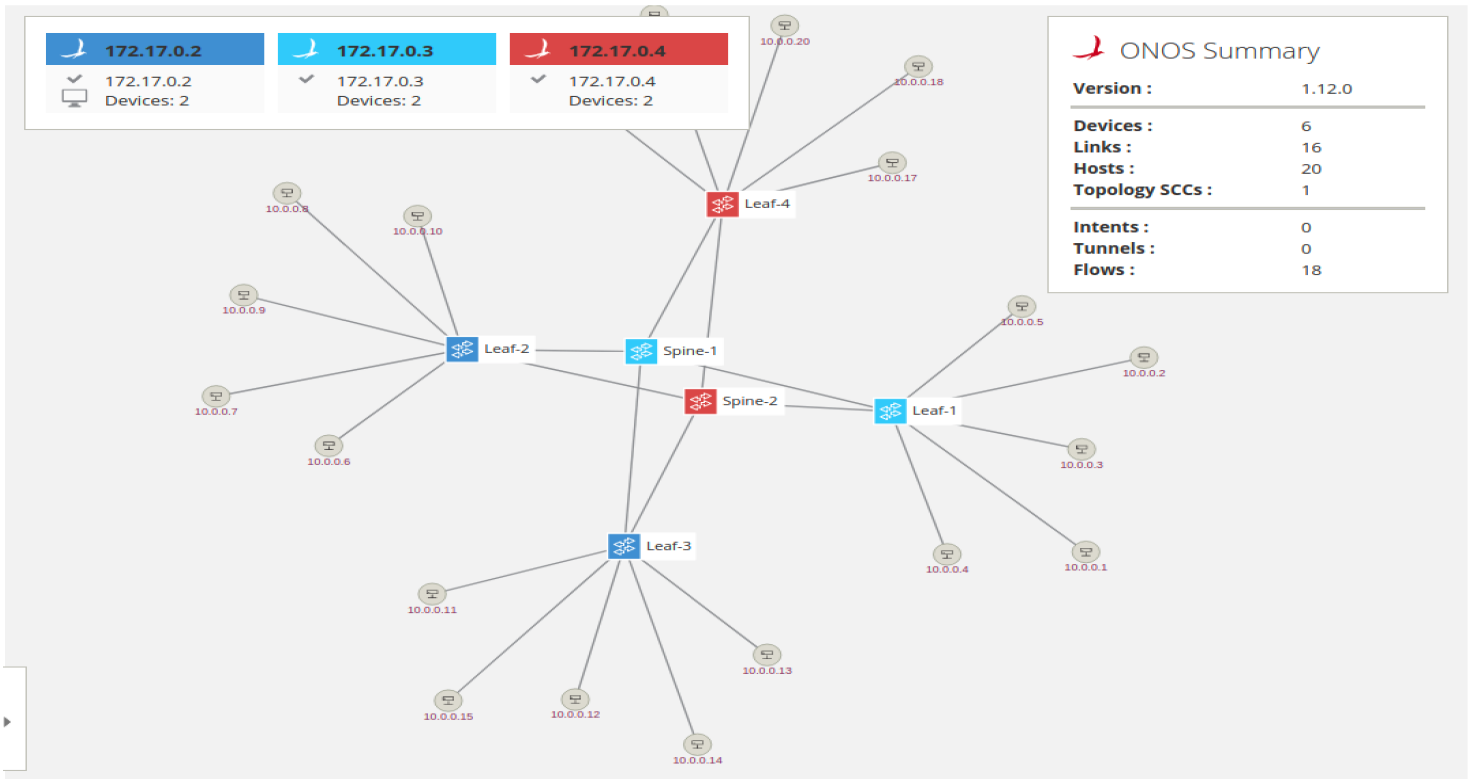
The ONOS GUI should now show the switches with the display looking similar to the following:



In order to display the node labels, press the ***L*** key to cycle between friendly lables, device ids and no labels. To toggle between showing and hiding hosts, you can press the ***H*** key. Press both ***L*** and ***H*** now.

Note that in the image above, all switches are now assigned mastership to the 3rd ONOS node (*172.17.0.4*), while the 1st and 2dn ONOS nodes have no devices for which they are the master. The ONOS GUI (as well as CLI) allow the user to force mastership re-balancing where the network devices will be roughly equally divided between all nodes in the ONOS cluster. To do this from the GUI, press the ***E*** key.

After toggling on host display, friendly labels and re-balancing, the display will look similar to the following:



Reactive Forwarding

In this exercise, we are going to use a sample app called *Reactive Forwarding.* It is shipped with ONOS and is a simple application that installs flows in response to every *miss* packet in that arrives at the controller.

Start by opening the ONOS CLI console by double clicking on the *ONOS CLI* icon.

No pings? Why?

First, let's see whether two hosts can reach each other via ICMP ping. Go to your mininet prompt and type the following:

```
mininet> h11 ping -c3 h41
```

You will notice that the ping fails as shown below.

```
mininet> h11 ping -c3 h41
PING 10.0.0.19 (10.0.0.19) 56(84) bytes of data.
From 10.0.0.1 icmp_seq=1 Destination Host Unreachable
From 10.0.0.1 icmp_seq=2 Destination Host Unreachable
From 10.0.0.1 icmp_seq=3 Destination Host Unreachable
--- 10.0.0.19 ping statistics ---
3 packets transmitted, 0 received, +3 errors, 100% packet loss, time 2009ms
```

So why did the ping fail? Well, there are no flows installed on the data-plane, which forward the traffic appropriately. ONOS comes with a simple *Reactive Forwarding* app that installs forwarding flows on demand, but this application is not activated by default. To see apps that are presently active, type the *apps -a -s* command and you will see the following output:

```
onos> apps -a -s
*  36 org.onosproject.optical-model        1.12.0   Optical Network Model
*  40 org.onosproject.openflow-base        1.12.0   OpenFlow Base Provider
*  41 org.onosproject.lldpprovider         1.12.0   LLDP Link Provider
*  44 org.onosproject.hostprovider         1.12.0   Host Location Provider
*  47 org.onosproject.drivers              1.12.0   Default Drivers
* 104 org.onosproject.openflow             1.12.0   OpenFlow Provider Suite
* 288 org.onosproject.proxyarp             1.12.0   Proxy ARP/NDP
```

Note: To see all installed apps, regardless whether they are active or not, use the same command, but without the *-a* flag. You should see over 140 different apps listed when you do that.

As you can see above, there is no reactive forwarding application currently active. Let's activate it.

Make it so, Number one

In the same ONOS CLI window, type the following to active the *Reactive Forwarding* app:

```
onos> app activate org.onosproject.fwd
Activated org.onosproject.fwd
```

Then, in a mininet window run the ping again, just this time don't limit the number of pings.

```
mininet> h11 ping h41
```

This time the ping is flowing:

```
mininet> h11 ping h41
PING 10.0.0.16 (10.0.0.16) 56(84) bytes of data.
64 bytes from 10.0.0.16: icmp_seq=1 ttl=64 time=39.6 ms
64 bytes from 10.0.0.16: icmp_seq=2 ttl=64 time=0.263 ms
64 bytes from 10.0.0.16: icmp_seq=3 ttl=64 time=0.058 ms
64 bytes from 10.0.0.16: icmp_seq=4 ttl=64 time=0.061 ms
64 bytes from 10.0.0.16: icmp_seq=5 ttl=64 time=0.065 ms
```

Start stop start stop....

You have now seen that you can activate applications into ONOS dynamically. Actually you can also deactivate applications while they are running so, for example, let's do this. Note that you can ommit the *org.onosproject* prefix and use *fwd* for short.

```
onos> app deactivate fwd
```

Observe that the ping has now stopped. This is because when the reactive forwarding application has withdrawn any flows it has installed before it stopped. We'll talk more about this in the next section. For now, let's activate the app again.

```
onos> app activate fwd
```

...and the ping restarts 😉

ONOS CLI commands

ONOS has many CLI commands. In this section, we will go through some of the most useful commands. This section may also serve as a CLI reference for you during this tutorial. While we will explain some of the ONOS CLI commands here you can find an exhaustive list by running:

```
onos> help onos
```

or more information about an individual command adding *--help* to any command. Also most commands have autocompletion to help you find the parameters quickly and easily.

Devices command

An SDN Controller would be nothing without devices to control. Luckily, ONOS has a convenient command to list the device currently known in the system. Running

```
onos> devices
```

will return the following information,

```
onos> devices
id=of:0000000000000001, available=true, local-status=connected 41m31s ago, role=STANDBY, type=SWITCH, m
id=of:0000000000000002, available=true, local-status=connected 41m31s ago, role=STANDBY, type=SWITCH, m
id=of:000000000000000b, available=true, local-status=connected 41m31s ago, role=STANDBY, type=SWITCH, m
id=of:000000000000000c, available=true, local-status=connected 41m31s ago, role=MASTER, type=SWITCH, mf
id=of:000000000000000d, available=true, local-status=connected 41m31s ago, role=MASTER, type=SWITCH, mf
id=of:000000000000000e, available=true, local-status=connected 41m31s ago, role=STANDBY, type=SWITCH, m
```

which consists of a device id, and a boolean value which indicates whether this devices is currently up. You also get the type of device and well as it's role relationship with this ONOS instance and other various attributes attached to each device.

Links command

Similarly, the *links* command is used to list the links detected by ONOS. At the ONOS prompt run

```
onos> links
```

and you should get the following output:

```
onos> links
src=of:0000000000000001/1, dst=of:000000000000000b/1, type=DIRECT, state=ACTIVE, expected=false
src=of:0000000000000001/2, dst=of:000000000000000c/1, type=DIRECT, state=ACTIVE, expected=false
src=of:0000000000000001/3, dst=of:000000000000000d/1, type=DIRECT, state=ACTIVE, expected=false
src=of:0000000000000001/4, dst=of:000000000000000e/1, type=DIRECT, state=ACTIVE, expected=false
src=of:0000000000000002/1, dst=of:000000000000000b/2, type=DIRECT, state=ACTIVE, expected=false
src=of:0000000000000002/2, dst=of:000000000000000c/2, type=DIRECT, state=ACTIVE, expected=false
...
```

The output shows you the list of discovered links. Reported links are formatted by source device-port pair to destination device-port pair. The *type* field indicates whether the link is a direct connection between two devices or not.

Hosts command

A network without hosts is a little like a city without bars, it would be a ridiculously boring place. Fortunately, ONOS has the ability to list the hosts (as opposed to bars, although that would be a great feature) currently in the system.

```
onos> hosts
```

with this output:

```
onos> hosts
id=00:00:00:00:00:01/None, mac=00:00:00:00:00:01, locations=[of:000000000000000b/3], vlan=None, ip(s)=[
id=00:00:00:00:00:02/None, mac=00:00:00:00:00:02, locations=[of:000000000000000b/4], vlan=None, ip(s)=[
id=00:00:00:00:00:03/None, mac=00:00:00:00:00:03, locations=[of:000000000000000b/5], vlan=None, ip(s)=[
id=00:00:00:00:00:04/None, mac=00:00:00:00:00:04, locations=[of:000000000000000b/6], vlan=None, ip(s)=[
id=00:00:00:00:00:05/None, mac=00:00:00:00:00:05, locations=[of:000000000000000b/7], vlan=None, ip(s)=[
id=00:00:00:00:00:06/None, mac=00:00:00:00:00:06, locations=[of:000000000000000c/3], vlan=None, ip(s)=[
id=00:00:00:00:00:07/None, mac=00:00:00:00:00:07, locations=[of:000000000000000c/4], vlan=None, ip(s)=[
...
```

Which displays the hosts' id as well as its mac address and where in the network it is connected.

Flows command

The *flows* command allows you to observe which flow entries are currently registered in the system. Flow entries may be in several states:

- **PENDING_ADD -** The flow has been submitted and forwarded to the switch.
- **ADDED -** The flow has been added to the switch.
- **PENDING_REMOVE -** The request to remove the flow has been submitted and forwarded to the switch.
- **REMOVED -** The rule has been removed.

So let's start some traffic by going to the mininet window and running

```
mininet> h11 ping h41
```

then in the ONOS window let's run the flows command

```
onos> flows
```

you should see the following output

```
onos> flows
deviceId=of:0000000000000001, flowRuleCount=6
    id=100007a585b6f, state=ADDED, bytes=330966, packets=4086, duration=3159, liveType=UNKNOWN, priorit
    id=100009465555a, state=ADDED, bytes=330966, packets=4086, duration=3159, liveType=UNKNOWN, priorit
    id=10000ea6f4b8e, state=ADDED, bytes=0, packets=0, duration=3159, liveType=UNKNOWN, priority=40000,
    id=125000008756fbe, state=PENDING_ADD, bytes=0, packets=0, duration=0, liveType=UNKNOWN, priority=1
    id=1000000ea1bfb, state=ADDED, bytes=0, packets=0, duration=1097, liveType=UNKNOWN, priority=5, tab
    id=10000021b41dc, state=ADDED, bytes=98, packets=1, duration=1097, liveType=UNKNOWN, priority=5, ta
deviceId=of:0000000000000002, flowRuleCount=6
    id=1000002bbd8d4, state=ADDED, bytes=330804, packets=4084, duration=3160, liveType=UNKNOWN, priorit
...
```

As you can see from the above output, ONOS provides many details about he the flows at the switches. For example each flow entry defines a selector and treatment which is the set of traffic matched by the the flow entry and how this traffic should be handled. Notice as well that each flow entry it tagged by an *appId* (application id), this *appId* identifies which application installed this flow entry. This is a useful feature because it can help an admin identify which application may be misbehaving or consuming many resources.

Paths command

Given a network topology, ONOS computes all the shortest paths between any two nodes. This is especially useful for your applications to obtain path information for either flow installation or some other use. The paths command takes two arguments, both of them are devices. To make things easy for you ONOS provides CLI autocompletion by simply hitting the <TAB> key.

```
onos> paths <TAB>
of:0000000000000001   of:0000000000000002   of:000000000000000b
of:000000000000000c   of:000000000000000d   of:000000000000000e
```

ONOS lists device options for you, thereby making it easier to find the devices you would like. For example, the output of the command below shows two paths of equal costs.

```
onos> paths of:000000000000000e of:000000000000000b
of:000000000000000e/2-of:0000000000000002/4==>of:0000000000000002/1-of:000000000000000b/2; cost=2.0
of:000000000000000e/1-of:0000000000000001/4==>of:0000000000000001/1-of:000000000000000b/1; cost=2.0
```

Intent Command

The intent command allows one to see what intents are stored in the system. Intents can be in several states:

- **SUBMITTED -** The intent has been submitted and will be processed soon.
- **COMPILING -** The intent is being compiled. This is a transient state.
- **INSTALLING -** The intent is in the process of being installed.
- **INSTALLED -** The intent has been installed.
- **RECOMPILING -** The intent is being recompiled after a failure.
- **WITHDRAWING -** The intent is being withdrawn.
- **WITHDRAWN -** The intent has been removed.
- **FAILED -** The intent is in a failed state because it cannot be satisfied.

For more information about Intents go here.

```
onos> intents
id=0x0, state=INSTALLED, type=HostToHostIntent, appId=org.onlab.onos.gui
    constraints=[LinkTypeConstraint{inclusive=false, types=[OPTICAL]}]
id=0x1, state=WITHDRAWN, type=HostToHostIntent, appId=org.onlab.onos.cli
    constraints=[LinkTypeConstraint{inclusive=false, types=[OPTICAL]}]
```

Note: You will not see any intents until some have been added. In the next section of the tutorial, you will add explicit host connectivity intent.

The command can also tell you what type of sub-intents the intent has been compiled to:

```
onos> intents -i
id=0x2, state=INSTALLED, type=HostToHostIntent, appId=org.onlab.onos.ifwd
    constraints=[LinkTypeConstraint{inclusive=false, types=[OPTICAL]}]
    installable=[
PathIntent{id=0x4, appId=DefaultApplicationId{id=2, name=org.onlab.onos.ifwd},
    selector=DefaultTrafficSelector{criteria=[ETH_SRC{mac=00:00:00:00:00:0D}, ETH_DST{mac=00:00:00:00:0
    treatment=DefaultTrafficTreatment{instructions=[]}, constraints=[LinkTypeConstraint{inclusive=false
    path=DefaultPath{src=ConnectPoint{elementId=00:00:00:00:00:0D/-1, portNumber=0},
                     dst=ConnectPoint{elementId=00:00:00:00:00:07/-1, portNumber=0}, type=INDIRECT,

PathIntent{id=0x5, appId=DefaultApplicationId{id=2, name=org.onlab.onos.ifwd},
    selector=DefaultTrafficSelector{criteria=[ETH_SRC{mac=00:00:00:00:00:07}, ETH_DST{mac=00:00:00:00:0
    treatment=DefaultTrafficTreatment{instructions=[]}, constraints=[LinkTypeConstraint{inclusive=false
    path=DefaultPath{src=ConnectPoint{elementId=00:00:00:00:00:07/-1, portNumber=0},
                     dst=ConnectPoint{elementId=00:00:00:00:00:0D/-1, portNumber=0}, type=INDIRECT,
```

For example, this host to host intent has been compiled to two path intents with the appropriate traffic selections and actions computed on your behalf.

State your intentions

One major advantage of using intents over simply using flow entries to program your network is that intents track the state of the network and reconfigure themselves in order to satisfy your intention. For example, if link were to go down the intent framework would reroute your intent (ie. your flows) onto an alternative path. But, what if there are no alternative path? Well, in this case the intent would enter the failed state and remain there until a path becomes available. Pretty cool, eh? Let's check this out in action.

First, let's deactivate the Reactive Forwarding application though:

```
onos> app deactivate fwd
Deactivated org.onosproject.fwd
```

Next, let's add a host connectivity intent for some two end-station hosts. You can use argument completion by pressing the ***Tab*** key.

```
onos> add-host-intent 00:00:00:00:00:01/None 00:00:00:00:00:10/None
Host to Host intent submitted:
HostToHostIntent{id=0x0, key=0x0, appId=DefaultApplicationId{id=2, name=org.onosproject.cli}, priority=
```

This command will provision a path between 10.0.0.1 (h11) and 10.0.0.16 (h41) and you can see that the intent is installed.

```
onos> intents
Id: 0x0
State: INSTALLED
Key: 0x0
Intent type: HostToHostIntent
Application Id: org.onosproject.cli
Resources: [00:00:00:00:00:01/None, 00:00:00:00:00:10/None]
Treatment: [NOACTION]
Constraints: [LinkTypeConstraint{inclusive=false, types=[OPTICAL]}]
Source host: 00:00:00:00:00:01/None
Destination host: 00:00:00:00:00:10/None
```

If you still have the *h11 ping h41* command running, you will see that the ICMP pings are still working beetween the two hosts.

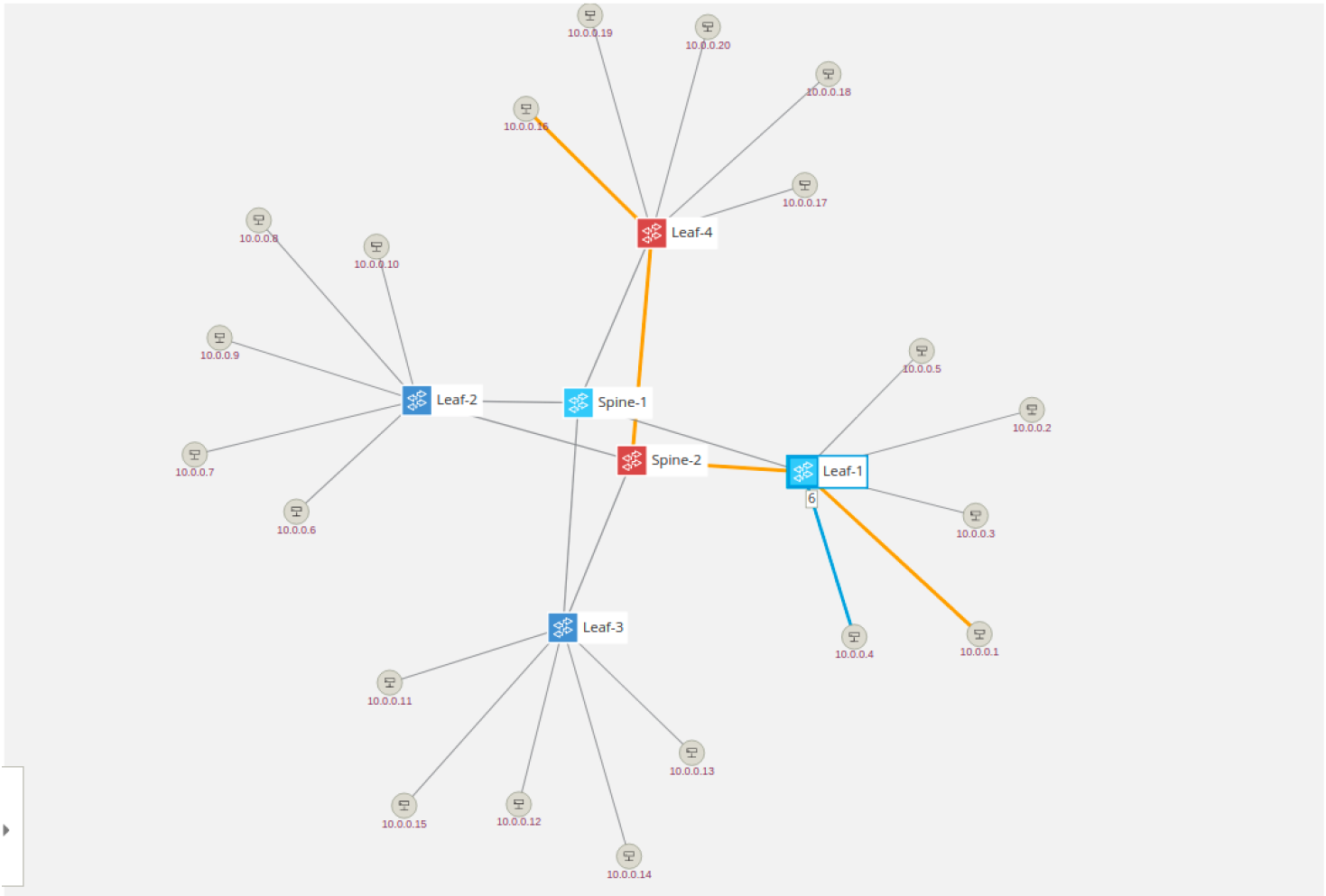
So now that the intent is installed, let's have a look what path it is using by using the flows command with summarized output using the *-s* flag for better readability:

```
onos> flows -s
deviceId=of:0000000000000001, flowRuleCount=3
    ADDED, bytes=429138, packets=5298, table=0, priority=40000, selector=[ETH_TYPE:bddp], treatment=[im
    ADDED, bytes=429138, packets=5298, table=0, priority=40000, selector=[ETH_TYPE:lldp], treatment=[im
    ADDED, bytes=0, packets=0, table=0, priority=40000, selector=[ETH_TYPE:arp], treatment=[immediate=[
deviceId=of:0000000000000002, flowRuleCount=5
    ADDED, bytes=428976, packets=5296, table=0, priority=40000, selector=[ETH_TYPE:lldp], treatment=[im
    ADDED, bytes=0, packets=0, table=0, priority=40000, selector=[ETH_TYPE:arp], treatment=[immediate=[
    ADDED, bytes=428976, packets=5296, table=0, priority=40000, selector=[ETH_TYPE:bddp], treatment=[im
    ADDED, bytes=32536, packets=332, table=0, priority=100, selector=[IN_PORT:1, ETH_DST:00:00:00:00:00
    ADDED, bytes=32536, packets=332, table=0, priority=100, selector=[IN_PORT:4, ETH_DST:00:00:00:00:00
deviceId=of:000000000000000b, flowRuleCount=5
    ADDED, bytes=214407, packets=2647, table=0, priority=40000, selector=[ETH_TYPE:bddp], treatment=[im
    ADDED, bytes=214407, packets=2647, table=0, priority=40000, selector=[ETH_TYPE:lldp], treatment=[im
    ADDED, bytes=294, packets=7, table=0, priority=40000, selector=[ETH_TYPE:arp], treatment=[immediate
    ADDED, bytes=32536, packets=332, table=0, priority=100, selector=[IN_PORT:2, ETH_DST:00:00:00:00:00
    ADDED, bytes=32536, packets=332, table=0, priority=100, selector=[IN_PORT:3, ETH_DST:00:00:00:00:00
deviceId=of:000000000000000c, flowRuleCount=3
    ADDED, bytes=214245, packets=2645, table=0, priority=40000, selector=[ETH_TYPE:bddp], treatment=[im
    ADDED, bytes=210, packets=5, table=0, priority=40000, selector=[ETH_TYPE:arp], treatment=[immediate
    ADDED, bytes=214245, packets=2645, table=0, priority=40000, selector=[ETH_TYPE:lldp], treatment=[im
```

We can see that the traffic flows between dpid 00:00:00:00:00:00:00:02 (Spine 2) and 00:00:00:00:00:00:00:0b (Leaf 1) and similarly between 00:00:00:00:00:00:00:02 (Spine-2) and 00:00:00:00:00:00:00:0e (Leaf-4).

Please note that the output from the tutorial and what you see may vary slightly as all alternate paths here have equal cost and therefore ONOS is free to pick either one.

You can visualize the intent using ONOS GUI by selecting the Leaf-1 node in the GUI and press the ***V*** key to show paths provisioned by intents that pass through the selected node. You should see something like this:



Let's see what happens if we sever the link between Spine-2 (s2) and Leaf-1 (s11), you may have to teardown the link between s1 and s11 so pay attention to the flows command output. This can be done in mininet by running:

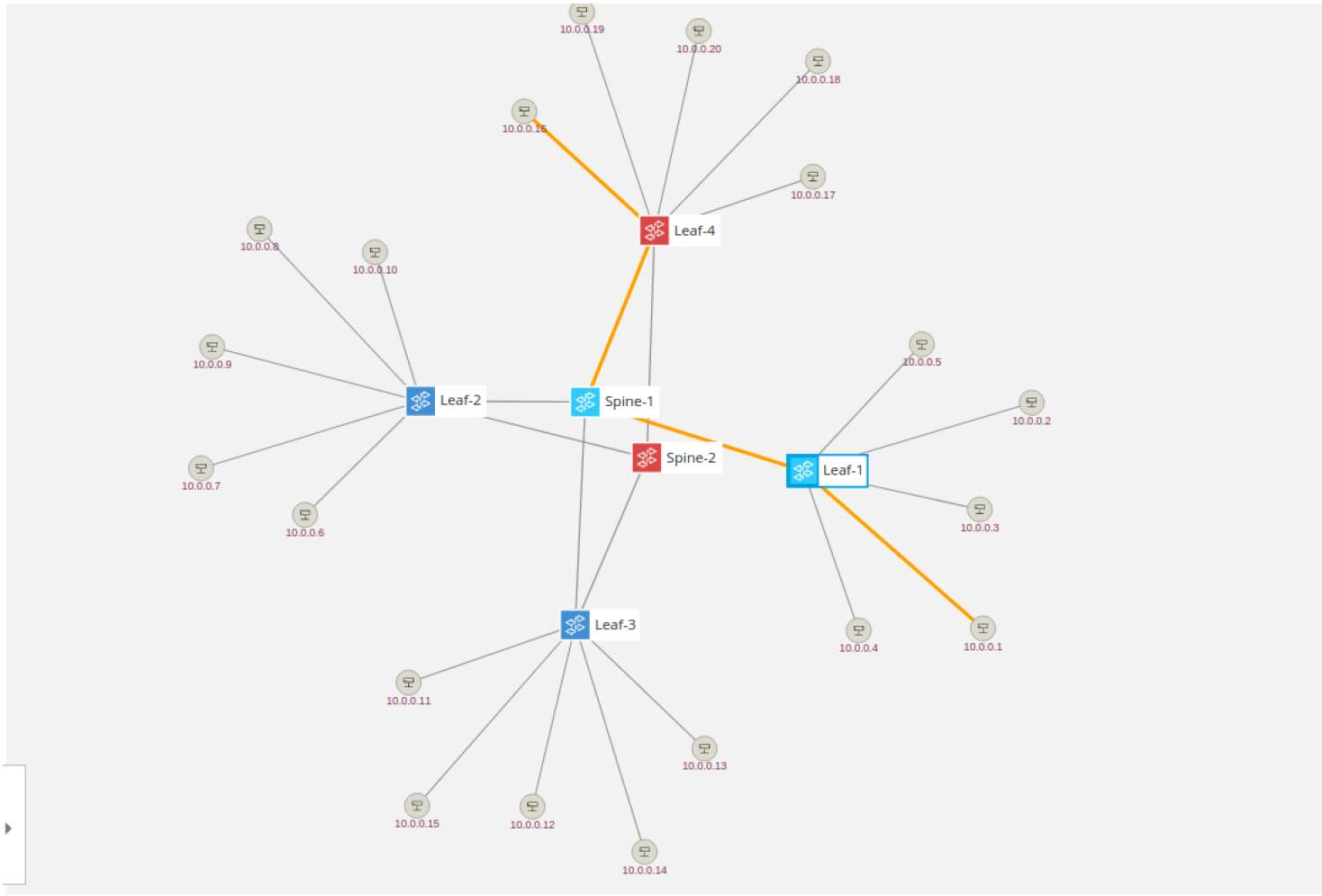
```
mininet> link s2 s11 down
```

After this, the GUI will show the link disappear. Selecting Leaf-1 and pressing V key again will show the newly established path which leads through the other spine switch:

Similarly, running the *flows -s* command again will show flows have moved to pass through the other spine.

How did this happen? Well when we tore down the link between s2 and s11, ONOS detected this change and informed all compo listening to topology events that the link went down. The intent service is one such component and when it receives this information it makes sure to repair any of the intents is affected by this change by provisioning connectivity using an alternate path.

This simple example shows that using intents is more powerful than simply installing flows. Intents maintain your intention (hence the name!) while retaining the ability to install them as is possible or most efficient.



Up down up down

If you wish you can take down more links and see what happens. Obviously, if you partition the network then no flows will be installed, sadly ONOS doesn't grow links between switches yet. You can bring up links in mininet by:

```
mininet> link s2 s11 up
```

Have fun!

ONOS Graphical User Interface

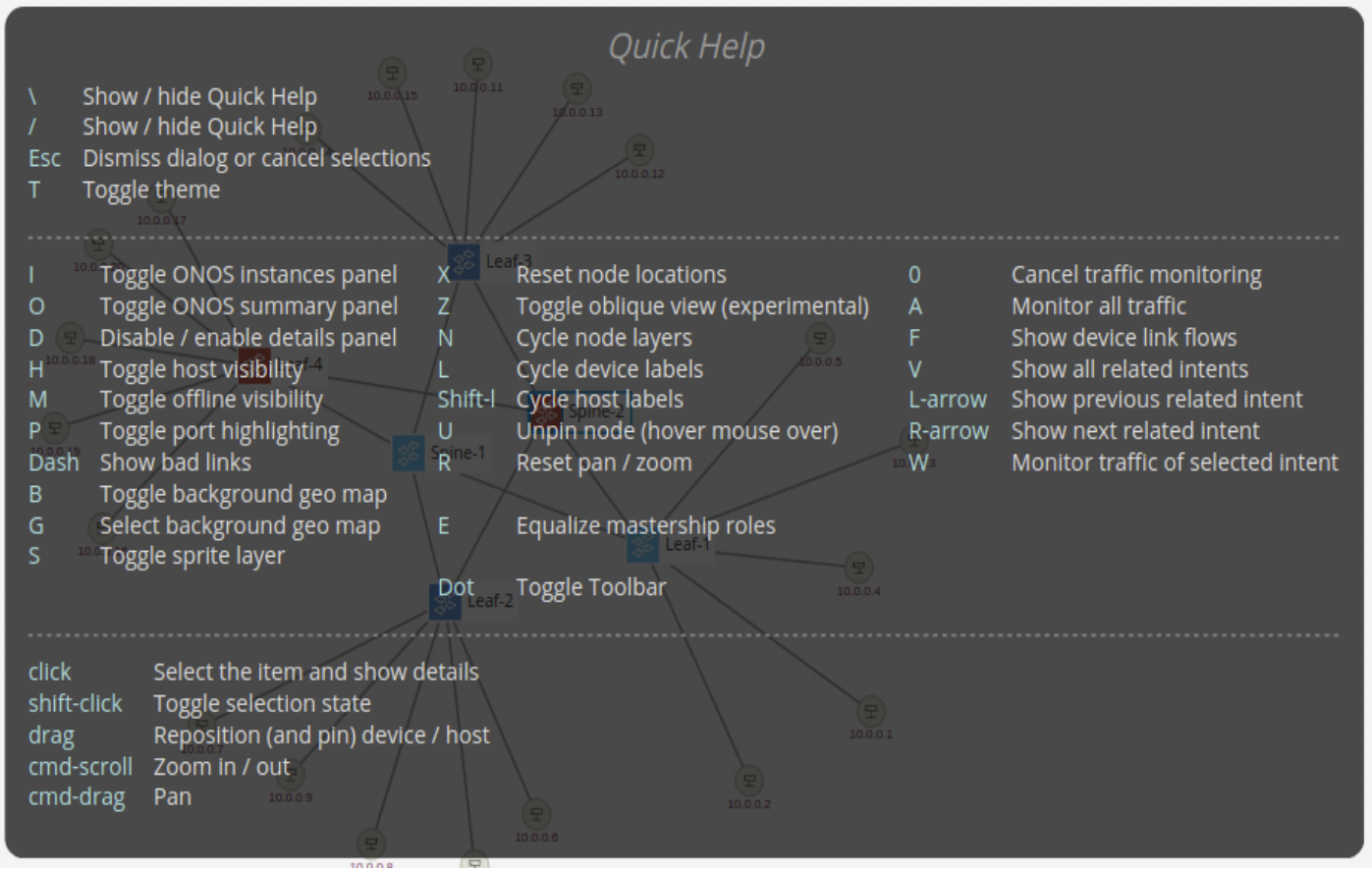
You have already briefly seen the ONOS GUI in action. So far it was used to help you visualize the network, but the GUI also allows many more ways to visualize information about the network and its elements, in addition to provisioning several types of connectivity intents.

To open the UI simply click on the *ONOS GUI* icon on the desktop. Let's go over some of the GUI features.

GUI Features

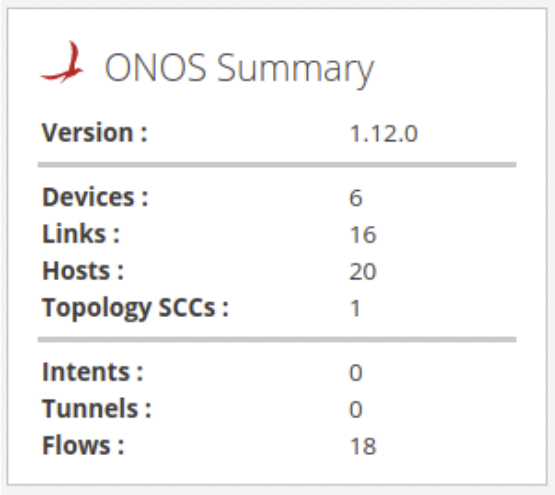
Quick Help

At anytime you can pull up the GUI's cheat sheet by pressing the ***Slash*** (*/*) key (which is **?** without the pesky shift 😉) and you will get an overlay pane that looks like below. You can dismiss this by pressing the ***Esc*** key. Each view, not just the topology view, provides a similar *Quick Help* overlay.



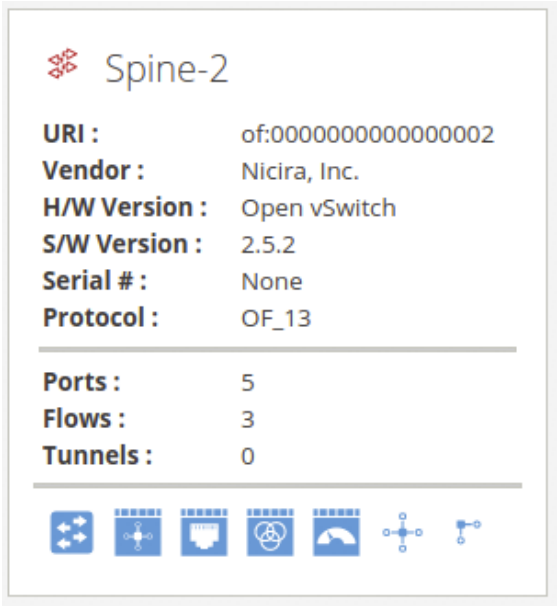
Overview pane

The *Topology View* of the GUI comes with a very useful overview pane. It shows you a summary about the network such as version of ONOS, number of devices, links, hosts, etc. You can toggle on/off the overview pane using the ***O*** key.



Details pane

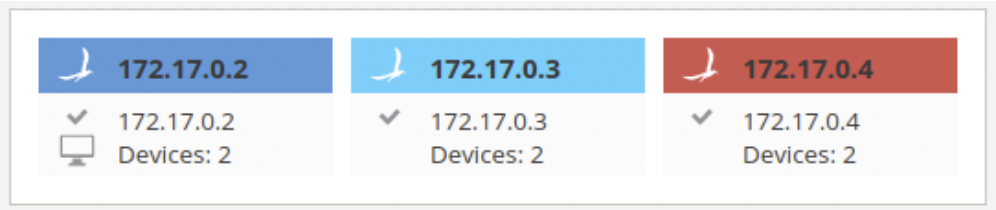
When you ***Left-Click*** on an element of the topology, such as a switch, host or a link, another pane appears on the right hand side below the *Overview* pane. This pane gives additional details about the selected item(s). Its display can be toggled on/off using the ***D*** key.



Note that the action buttons at the bottom of the *Details* pane can be used to jump to different views of the ONOS GUI for additional context. So select multiple items, hold the ***Shift*** key while you ***Left-Click***. To unselect an item hold ***Shift*** and ***Left-Click*** it to toggle off the selection. You can also Left-Click anywhere on the blank part of the *Topology View* to unselect all items.

Instances pane

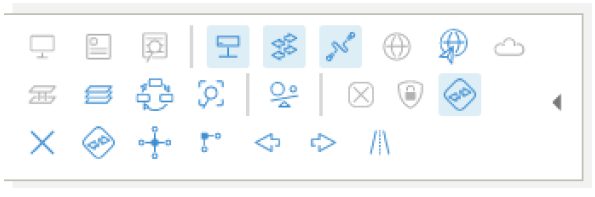
The GUI has the ability to show which ONOS instances are active. By hitting the 'i' key (it will be open by default) you will see a pane show up on the left hand side as shown below.



Notice that the glyphs for the switches changes color, this indicates which switches are controlled by which instance. This is useful to see at a glance which switches are controlled by which ONOS instance. The terminal glyph indicates which instance the GUI is presently connected to and the check-mark glyph indicates that the instance is in ready state, which means fully operating as part of the cluster.

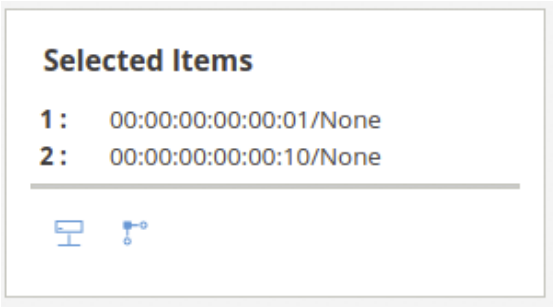
Toolbar pane

While lot of the features of the Topology View can be operating using solely the keyboard keys, a toolbar pane was provided to make it visually easier to identify all possible actions. The toolbar is located in the lower left-hand corner of the view and is hidden by default. You can toggle its display on/off using the ***Period*** (**.**) key.

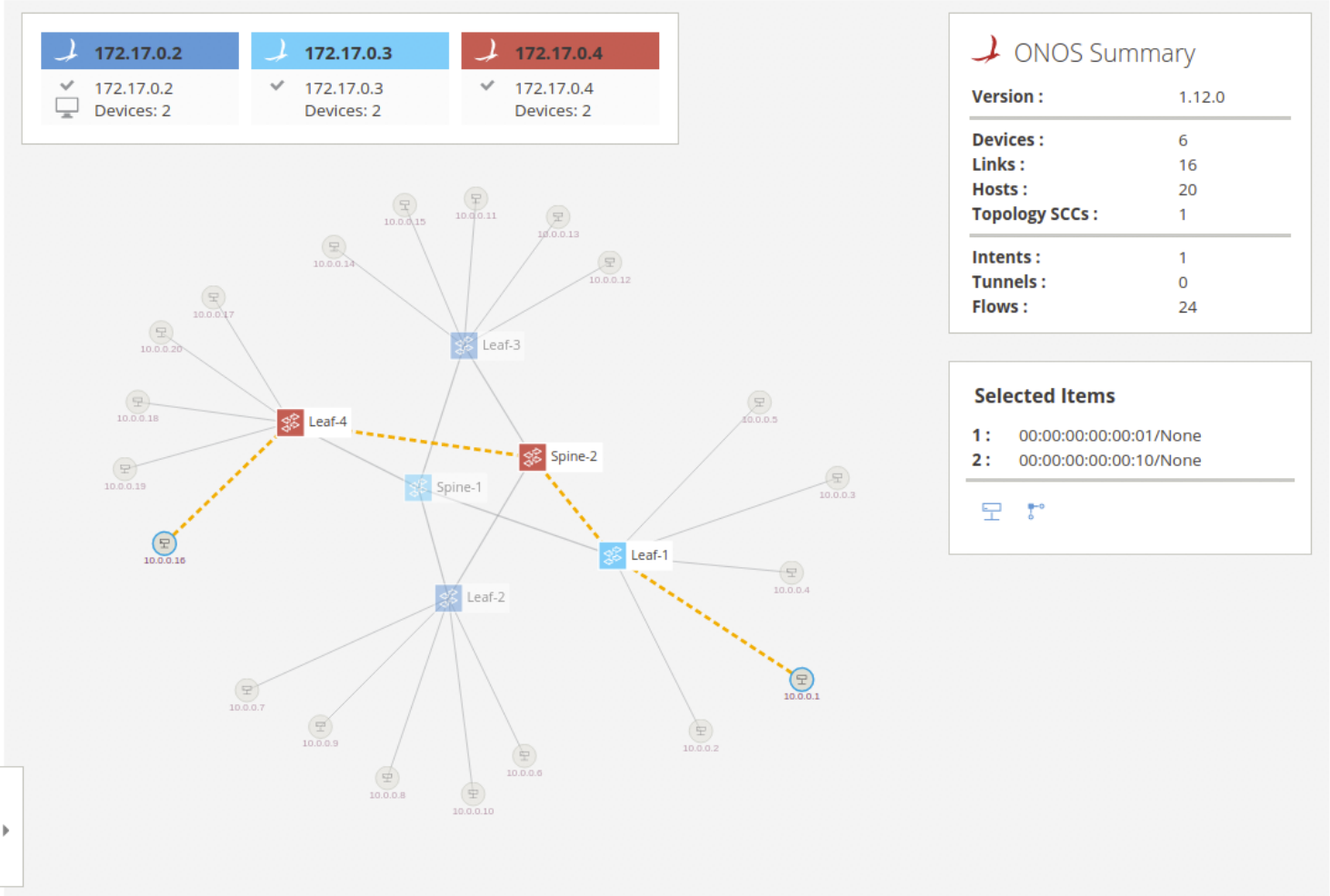


Install Intent

Ok let's install an intent using the UI. First select two hosts by clicking on one host then shift-click on another. Let's pick 10.0.0.20 and 10.0.0.9. Now a pane will appear on the right and side of the screen as here:



Now click on 'Create Host-to-host Flow', this actually provisions a host to host intent and lights up the path used by the intent.



You can check that the intent was installed via the ONOS CLI *intents* command (exercise left to the reader) and verify that the two hosts can indeed reach each other using the Mininet *ping* command.

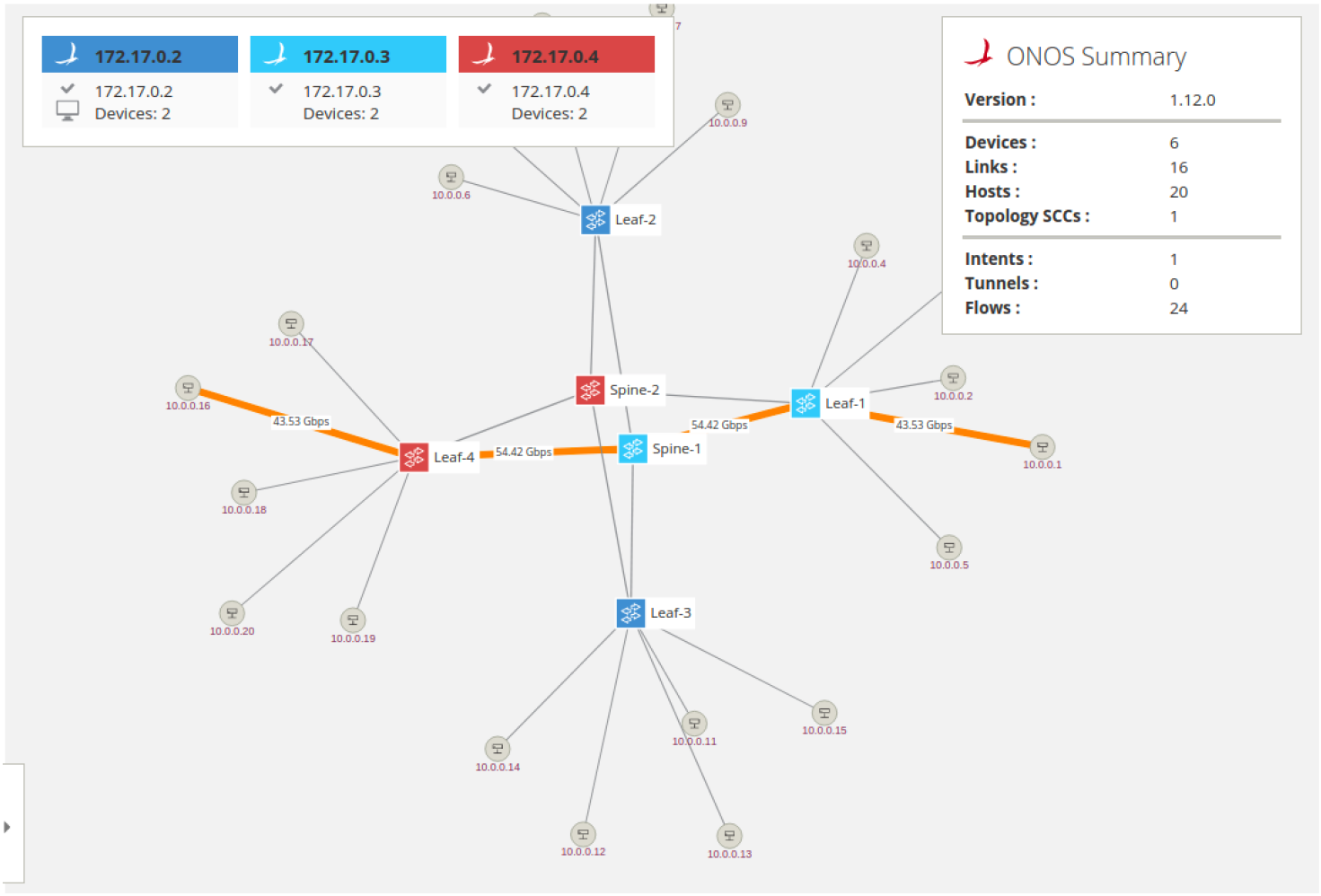
To send a more significant amount of traffic, than a mere ICMP ping, between the two hosts you can use the bgPerf mininet command as follows:

```
mininet> bgIperf h11 h41
```

You can include more than two hosts in the above command and traffic will be generated between all of them, pairwise. Before you do that, however, you should make sure that they can reach each other by creating the appropriate connectvity intents.

Show all traffic

Another thing you can do is to visually monitor the network traffic in the UI. You can cycle between different modes, e.g. port stats in bits/s or packets/s and flow stats in bits/s or packets/s, by pressing the ***A*** key.



Play on

Now you know the main features of the GUI. We encourage you to play around with it to find out what other features you can use and who knows may find a few bugs.

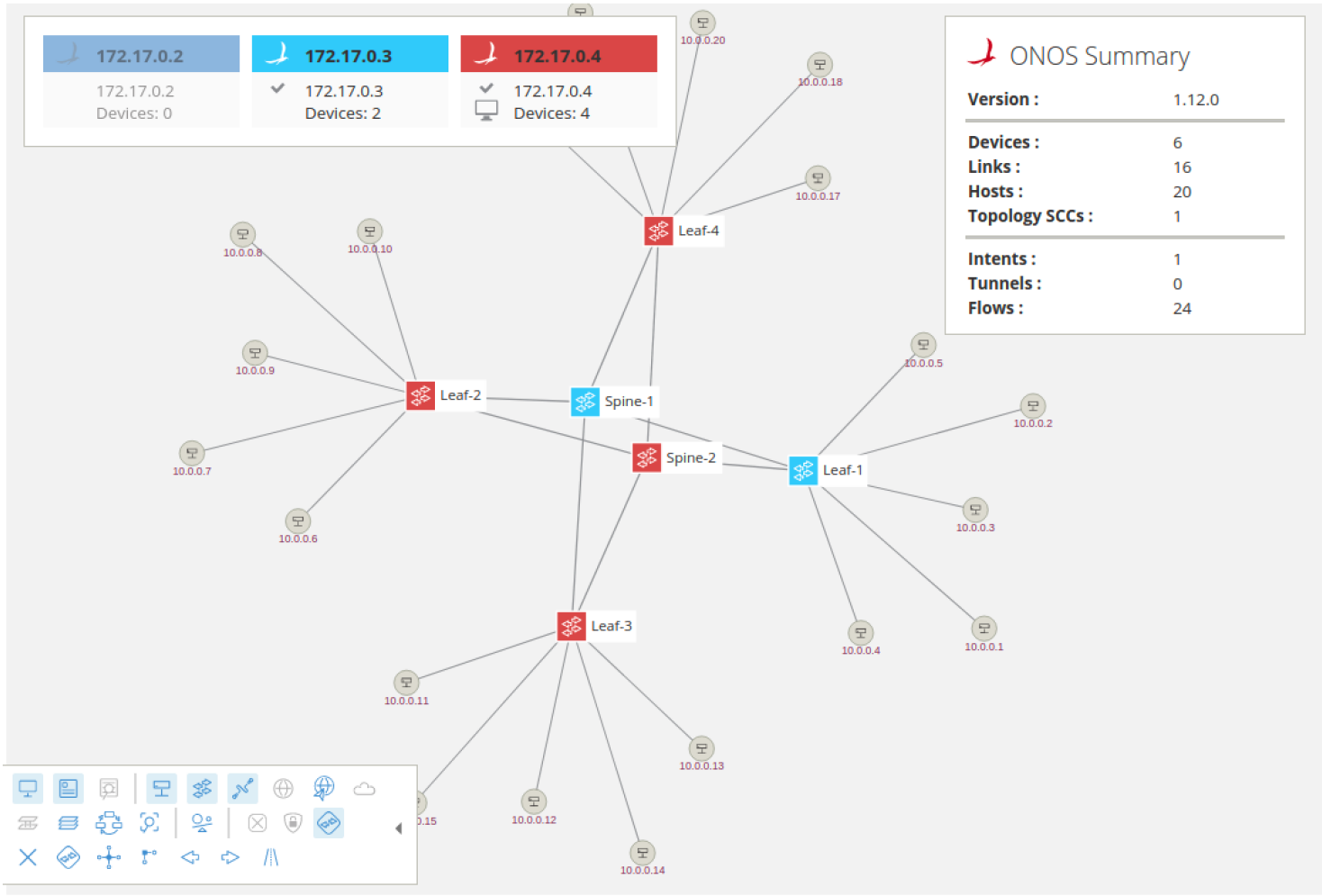
Cluster Operation

Since we are running ONOS in a clustered configuration, let's see how ONOS deals with node failures.

We're going to use the ONOS CLI to shutdown the 1st instance (172.17.0.2) by issuing the *shutdown* command and answering the confirmation prompt with *yes*:

```
onos> shutdown
Confirm: halt instance root (yes/no): yes
onos> Connection to 172.17.0.2 closed by remote host.
```

After this, you should see the GUI connection fail-over to another ONOS cluster node, the device masterships to be re-assigned and the 1st node icon in the Instances pane will turn gray to indicate that the node is not reachable, similar to what is shown below:



In this case, the GUI reconnected to the 3rd cluster node and the two devices previously mastered by the node which is now shutdown have been adopted by the 3rd node as well. Otherwise, all operation should proceed as normal; this includes the GUI.

Note that since the node is set to autostart, it will come back and rejoin the cluster shortly. There is nothing required on your part to do that. If you wish to rebalance the mastership again, you can use the GUI or the following CLI command:

```
onos> balance-masters
```

You can also activate the *Master Load Balancer* app to periodically re-balance the mastership automatically.

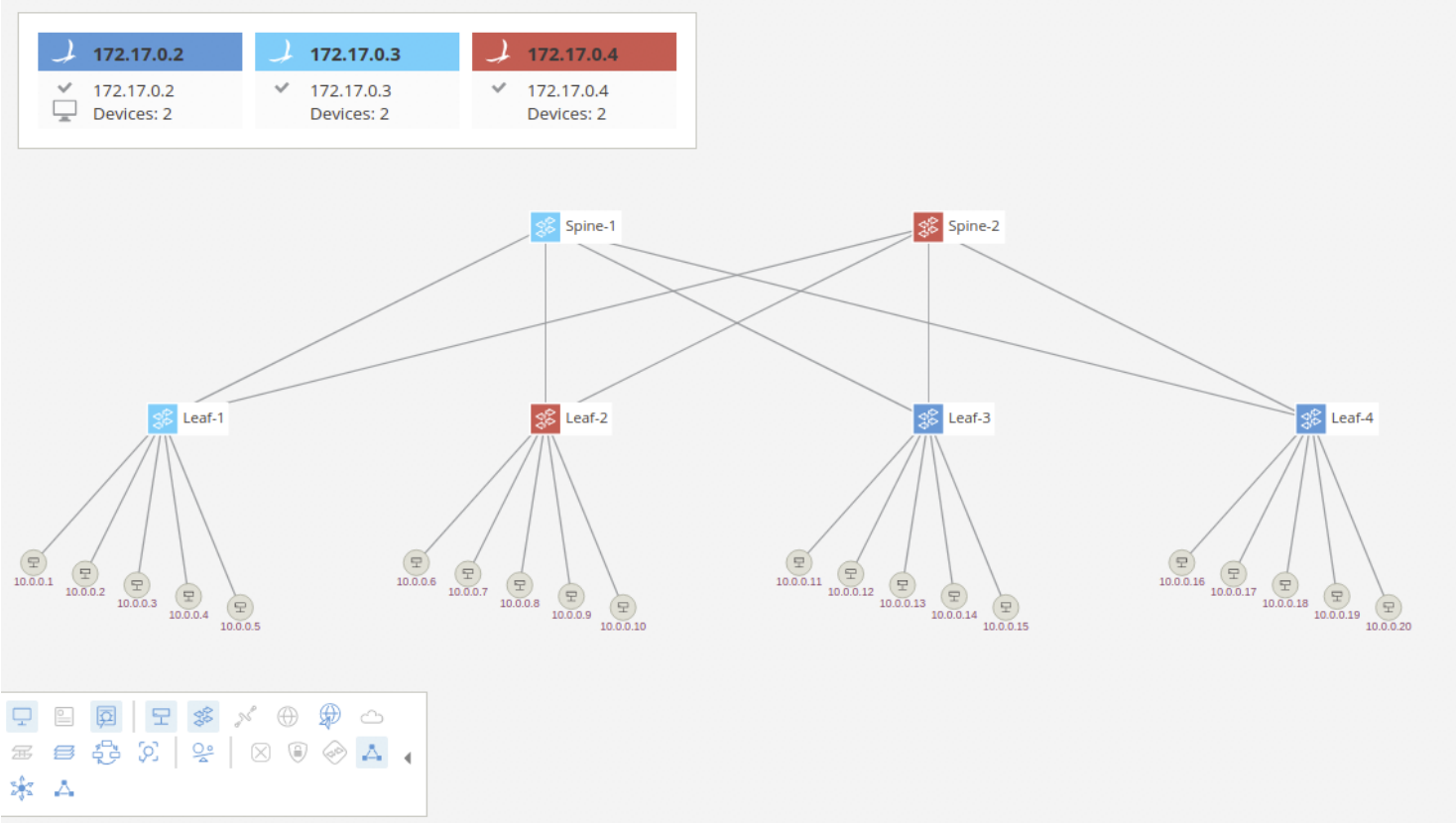
```
onos> app activate mlb
```

UI Autolayout

One of the applications available with standard ONOS distribution (and activated by default in this tutorial setup) allows for automatic layout of the network topology in the GUI. Presently, only two layouts are supported, one for access networks (spine-leaf and variants) and the other is to return to the default force-layout that we have been using up to this point of the tutorial. To try this, you can type the following from the ONOS CLI:

```
onos> topo-layout access
```

Afterwards, you should see the GUI change display to resemble a more familiar depiction of the spine-leaf fabric as show below:



You can of course also switch between different layouts using the provided GUI overlay that can be activated in the *Topology View* toolbar.

EXERCISES

1. Construct simple Network using external ONOS SDN Controller. All nodes should be able to communicate with each other. Use ICMP protocol for the verification of network connectivity. Also attach the snapshots of network and results of ping command

F/OBEM 01/18/00

NED University of Engineering & Technology

Department of Software Engineering

Course Code & Title: CS - 351 Computer Communication Networks

Assessment Rubric for Open Ended Lab

Criterion	Level of Attainment		
	Average (0)	**Good (0.5)**	**Excellent (1)**
Installation	**The student did not install the tool.**	**The student installed the tool partially.**	**The student installed the tool satisfactorily.**
Configuration	**The student did not configure the tool.**	**The student configured the tool partially.**	**The student configured the tool satisfactorily.**
Scenario Implementation	**The student did not build the scenario.**	**The student built the scenario partially.**	**The student built the scenario satisfactorily.**
Scenario Operability	**The implemented scenario is non-operational.**	**The implemented scenario is partially operational.**	**The implemented scenario is fully operational.**
Lab Requirements Fulfillment	**The implementation did not fulfill the lab requirements.**	**The implementation fulfilled the lab requirements partially.**	**The implementation fulfilled the lab requirements satisfactorily.**

Student's Name: _____ Roll No.: _____

Total Score = _____

Instructor's Signature: _____

DEPARTMENT OF SOFTWARE ENGINEERING

BACHELORS IN SOFTWARE ENGINEERING

Course Code: CS-351

Course Title: Computer Communication Networks

Open Ended Lab (EOL)/Complex Engineering Activity (CEA)

TE Batch 2022, Spring Semester 2025

Course Learning Outcome

CLO 3: Practice the configuration of networks using modern tools. (P3-PLO5).

Complex problem-solving attributes (CPA) covered (as per PEC - OBA manual – 2019)

- **CPA-1 Depth of analysis required:** These problems do not have readily apparent solutions and demand abstract thinking and innovative analysis to develop appropriate models.

- **CPA-2 Level of interaction:** Necessitate the resolution of substantial challenges resulting from the interplay of diverse or conflicting technical, engineering, or other factors.

Problem Statement

Students are required to create a network using open-source tools. It is required to perform alterations in the network in such a way that the network become either performance/security/energy efficient. The assigned task uses in-depth knowledge of TCP/IP Layers. Students must first properly install and configure the tool. Once the configuration is completed then implement the given scenario.

Minimum Required Features:

1. Lab Tasks:

- Network creation in Mininet/ ONOS and packet tracer.
- Network modification algorithm.

2. Implementation: Create a network using any chosen tool and modify it as per the said objective.

Instructions and Guidelines:

1. Implementation: Tools that can be used for implementation include SDN based tools (ONOS, Ryu Faucet, Faucet, OpenDaylight), NFV tools (OpenNF), CloudSim, Kubernetes, Docker etc.

2. Documentation: The document must be prepared through the available templates of IEEE, Elsevier or Wiley.

Constraints/ Assumptions:

- Students should work only on open-source tools.
- Students are expected to take guidance from Laboratory work as well.
- It's a group-based design project.

Expected Outcomes:

Students need to present a demonstration of implementation.

DEPARTMENT OF SOFTWARE ENGINEERING

BACHELORS IN SOFTWARE ENGINEERING

Course Code: CS-351

Course Title: Computer Communication Networks

Open Ended Lab (EOL)/Complex Engineering Activity (CEA)

TE Batch 2022, Spring Semester 2025

Grading Rubric

Group Members:

Student No.	Name	Roll No.
S1		
S2		
S3		
S4		

CRITERIA AND SCALES				Marks Obtained			
				S1	S2	S3	S4
Criterion 1: The novelty and innovation in the idea? (CPA-1)							
1	**2**	**3**	**4**				
Idea is novel but not innovative.	Idea is innovative but not novel.	Idea is novel and partially innovative.	Idea is novel and innovative.				
Criterion 2: To what extent the student installed and configured the tool? (CPA-1, CPA-2)							
0	**1**	**2**	**3**				
The student did not install and configure the tool.	The student installed the tool but not configured it properly.	The student installed and configured the tool partially.	The student installed and configured the tool satisfactorily.				
Criterion 3: To what extent the student implemented the assigned scenario in a configured tool? (CPA-2)							
0	**1**	**2**	**3**				
The student did not implement the scenario in a configured tool.	The student implemented the scenario in a configured tool unsatisfactorily.	The student implemented the scenario in a configured tool partially.	The student implemented the scenario in a configured tool satisfactorily.				
			Total Marks:				

Teacher's Signature

F/OBEM 01/18/00

NED University of Engineering & Technology

Department of Software Engineering

Course Code & Title: CS - 351 Computer Communication Networks

Performance based Rubric Evaluation (Final Lab Exam)

Criterion	Level of Attainment		
	Average (0)	Good (0.5)	Excellent (1)
Tool-Specific Knowledge	**Demonstrates limited understanding of tools and concepts.**	**Demonstrates a basic understanding of some tools, but explanations may be unclear.**	**Demonstrates a deep understanding of each tool's functionalities, command options, and appropriate use cases.**
Task Implementation	**The student did not build the scenario.**	**The student built the scenario partially.**	**The student built the scenario satisfactorily.**
Task Operability	**The implemented scenario is non-operational.**	**The implemented scenario is partially operational.**	**The implemented scenario is fully operational.**
Task Accuracy	**Frequent errors and inaccuracies.**	**Few errors, minor inaccuracies.**	**No errors, highly accurate task completion.**
Troubleshooting	**Unable to troubleshoot effectively, demonstrating a limited ability to solve problems.**	**Struggles to identify or resolve issues without significant guidance, lacking systematic troubleshooting skills.**	**Independently identifies and resolves tool-related issues, employing effective troubleshooting strategies and resources.**

Student's Name: _____ Roll No.: _____

Total Score = _____

Instructor's Signature: _____