RV Educational Institutions ®
RV College of Engineering ®

Autonomous Institution Affiliated to Visvesvaraya Technological University, Belagavi

Approved by AICTE, New Delhi

Go, change the world

Report

On

Computer Networks

20MCA13

MINININET

Submitted in Partial Fulfillment of the Requirement

for the I Semester MCA

MASTER OF COMPUTER APPLICATIONS

By

RVCE21MCA067	PAVAN V CHAKRASALI
RVCE21MCA065	PAVITHRA T

Under the Incharge
of

Dr. Jayasimha S R
Assistant Professor

Department of Master of Computer Applications
RV College of Engineering®, Mysuru Road
RV Vidyanikethan Post, Bengaluru – 560059

MAY -2022

DEPARTMENT OF MASTER OF COMPUTER APPLICATIONS

CERTIFICATE

This is to certify that the assignment entitled "MININET" submitted in partial fulfillment Computer Networks (20MCA13) of I Semester MCA is a result of the bonafide work carried out by PAVAN V CHAKRASALI, RVCE21MCA067 , during the Academic year 2021-22.

Assignment - 1	Assignment - 2	FINAL

Dr. Jayasimha S R
Assistant Professor
Department of MCA,
RV College of Engineering®

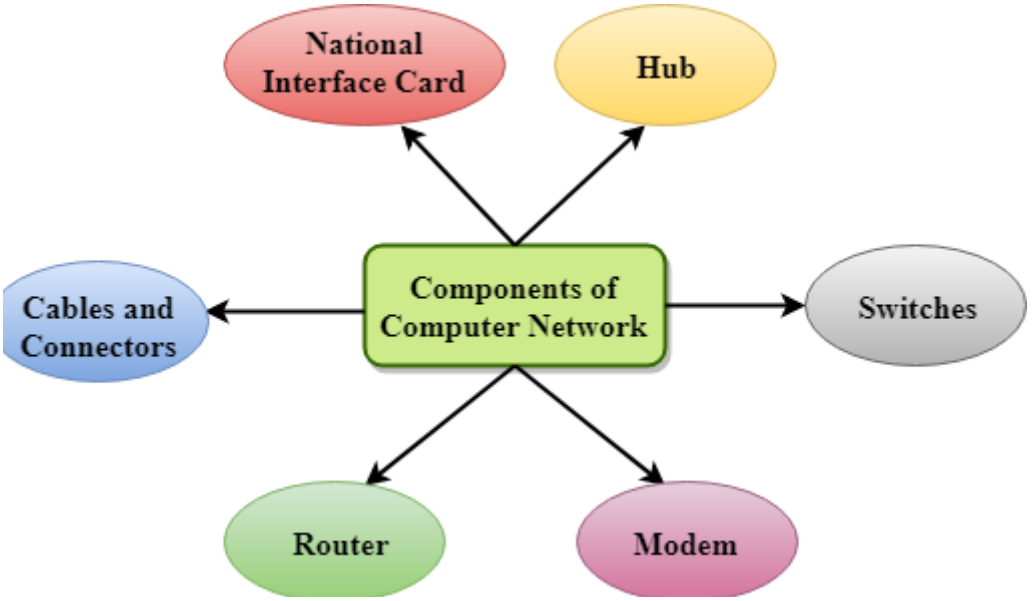
Dr. Andhe Dharani
Professor and Director
Department of MCA,
RV College of Engineering®

INDEX PAGE

INTRODUCTION TO COMPUTER NETWORKS

COMPUTER NETWORK IS A GROUP OF COMPUTERS CONNECTED WITH EACH OTHER THROUGH WIRES, OPTICAL FIBRES OR OPTICAL LINKS SO THAT VARIOUS DEVICES CAN INTERACT WITH EACH OTHER THROUGH A NETWORK.

COMPONENTS OF COMPUTER NETWORKS:



USES OF COMPUTER NETWORK

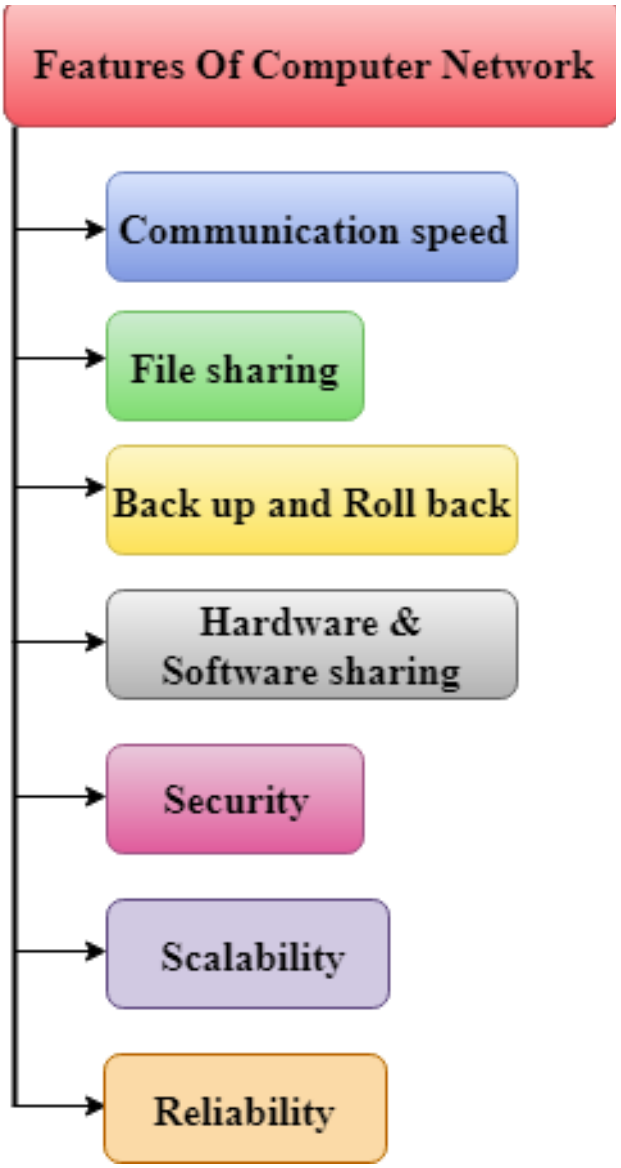
Resource sharing: Resource sharing is the sharing of resources such as programs, printers, and data among the users on the network without the requirement of the physical location of the resource and user.

Server-Client model: Computer networking is used in the server-client model. A server is a central computer used to store the information and maintained by the system administrator. Clients are the machines used to access the information stored in the server remotely.

Communication medium: Computer network behaves as a communication medium among the users. For example, a company containing more than one computer has an email system which the employees use for daily communication.

E-commerce: Computer networks are also important in businesses. We can do business over the internet. For example, amazon.com is doing their business over the internet, i.e., they are doing their business over the internet.

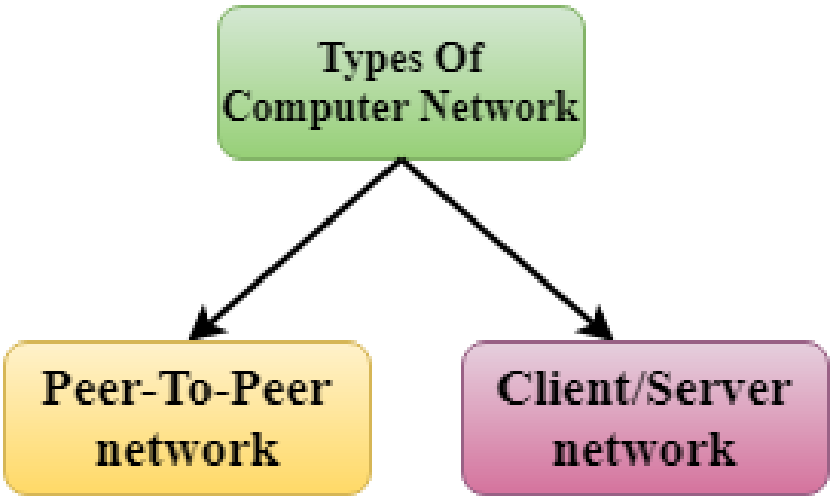
FEATURES OF COMPUTER NETWORKS:



COMPUTER NETWORK ARCHITECTURE:

COMPUTER NETWORK ARCHITECTURE IS DEFINED AS THE PHYSICAL AND LOGICAL DESIGN OF THE SOFTWARE, HARDWARE, PROTOCOLS, AND MEDIA OF THE TRANSMISSION OF DATA. SIMPLY WE CAN SAY THAT HOW COMPUTERS ARE ORGANIZED AND HOW TASKS ARE ALLOCATED TO THE COMPUTER.

The two types of network architectures are used:

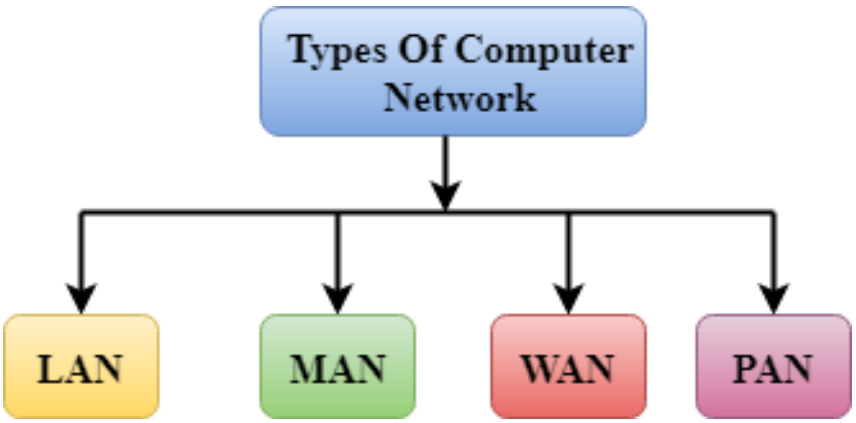


Peer-To-Peer network
Client/Server network

COMPUTER NETWORK TYPES:

A COMPUTER NETWORK IS A GROUP OF COMPUTERS LINKED TO EACH OTHER THAT ENABLES THE COMPUTER TO COMMUNICATE WITH ANOTHER COMPUTER AND SHARE THEIR RESOURCES, DATA, AND APPLICATIONS.

A computer network can be categorized by their size. A computer network is mainly of four types:

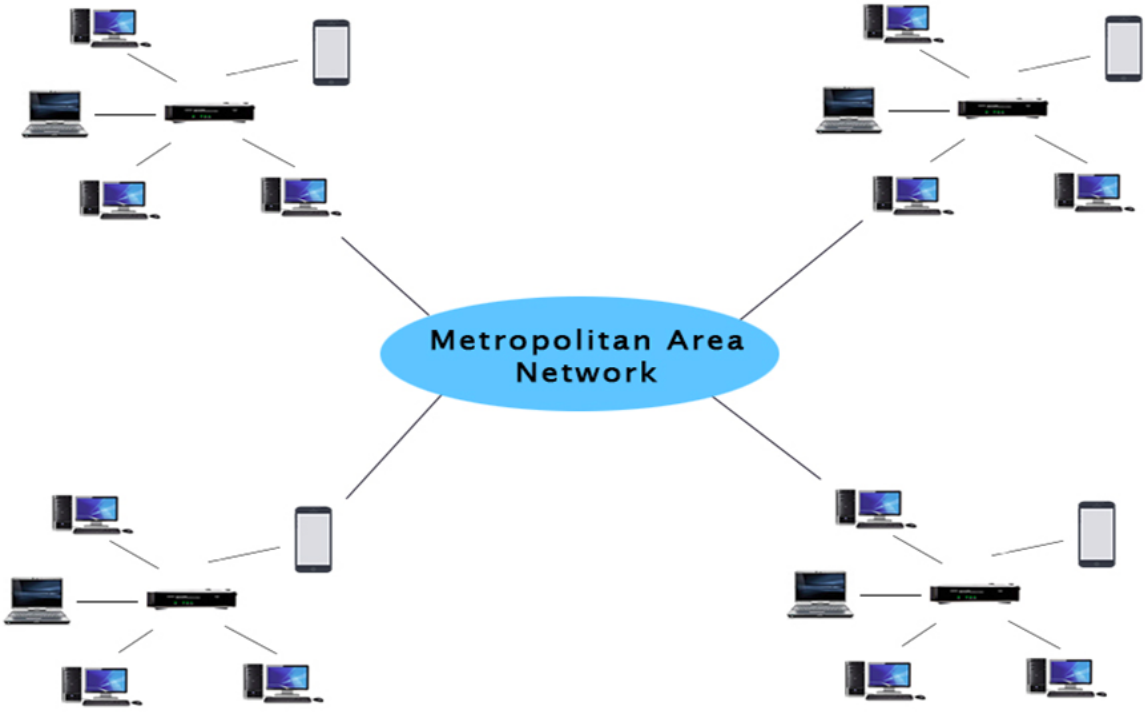


LAN(Local Area Network)
PAN(Personal Area Network)
MAN(Metropolitan Area Network)
WAN(Wide Area Network)

LAN(LOCAL AREA NETWORK):

PAN(PERSONAL AREA NETWORK):

MAN(METROPOLITAN AREA NETWORK):



TOOL USAGE:

MININET:

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.

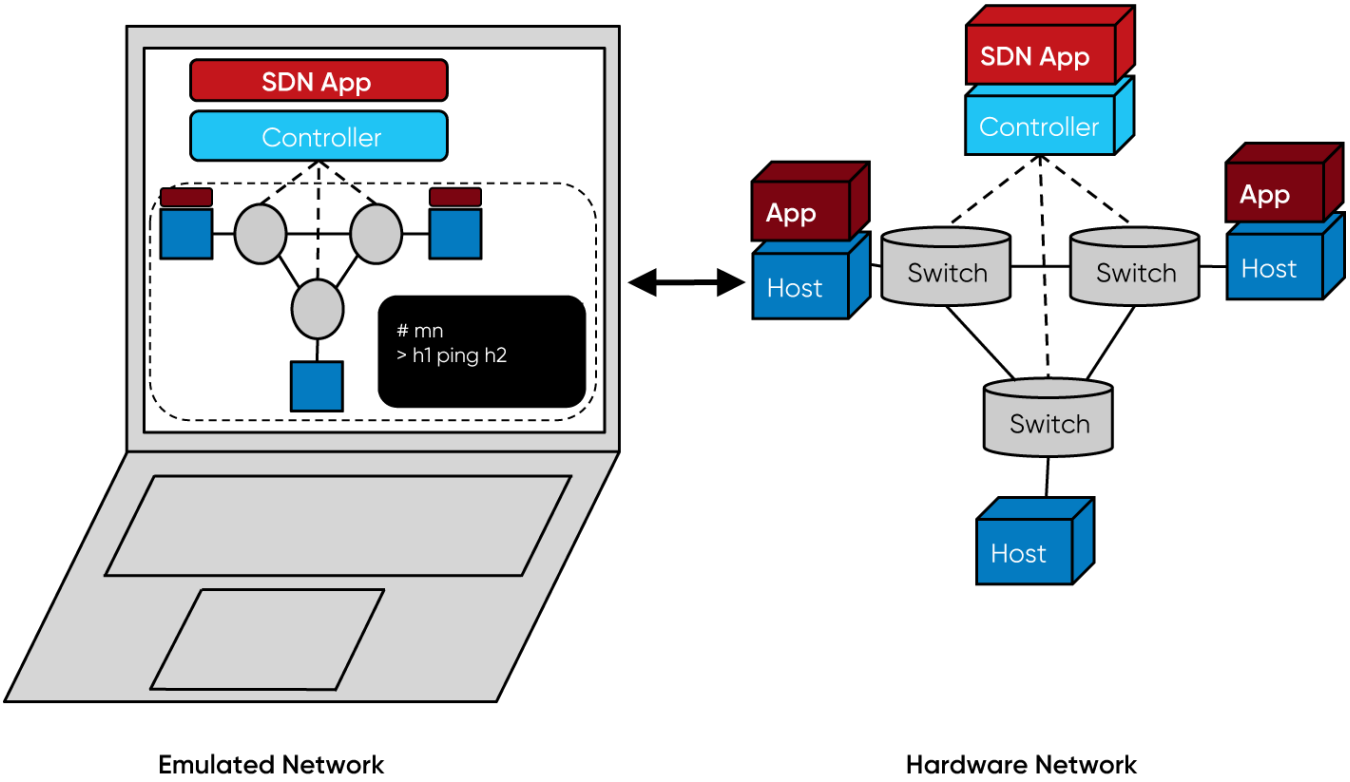


FIGURE 1. HARDWARE NETWORK VS. MININET EMULATED NETWORK.

Mininet is useful for development,teaching,and research as it is easy to customize and interact with through the CLI or the GUI. Mininet was originally designed to experiment with OpenFlow 2 and Software-Defined Networking (SDN) 3 . 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 4 . 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.

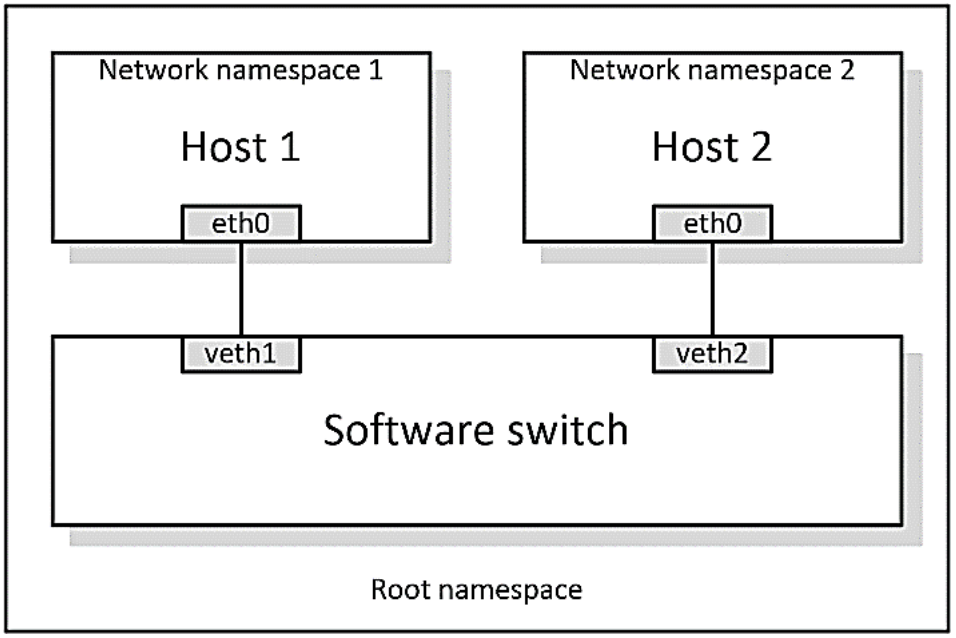


FIGURE 2. NETWORK NAMESPACES AND VIRTUAL ETHERNET LINKS.

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.

TOOL INSTALLATION AND FEATURES

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

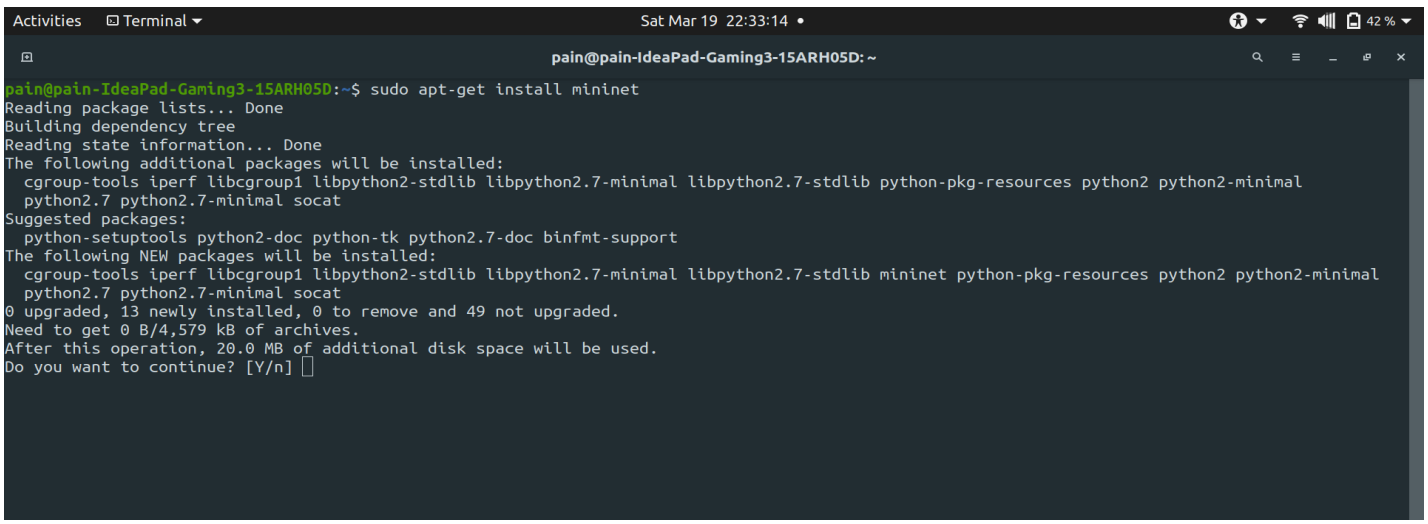
Invoke Mininet using the default topology:

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

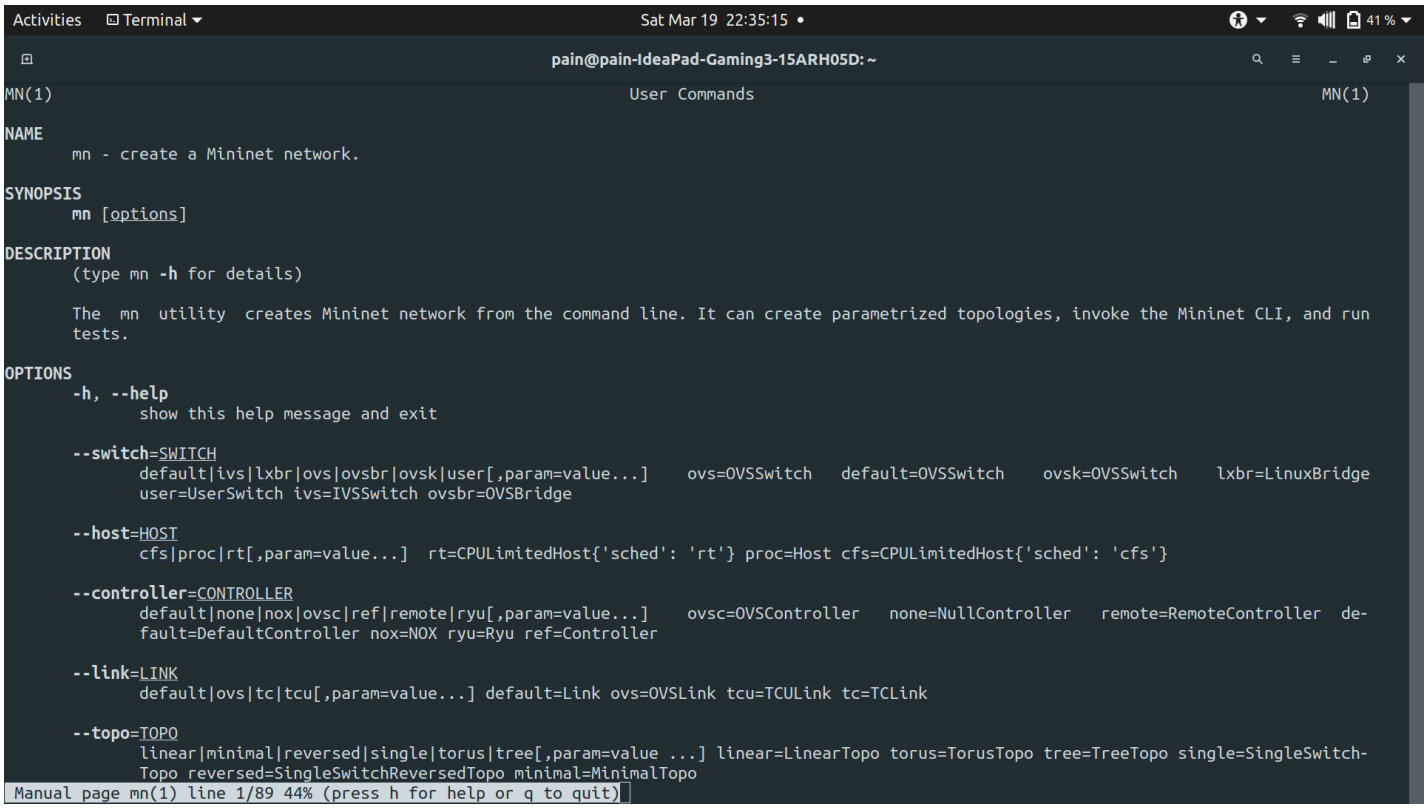
The Linux terminal is a program that opens a window and permits you to interact with a command-line interface (CLI). A CLI is a program that takes commands from the keyboard and sends them to the operating system for execution.

Step 2. To install mininet ,first update the system using the command
$sudo apt-get update

Step 3. After updating the system install mininet using the command
$sudo apt-get install mininet

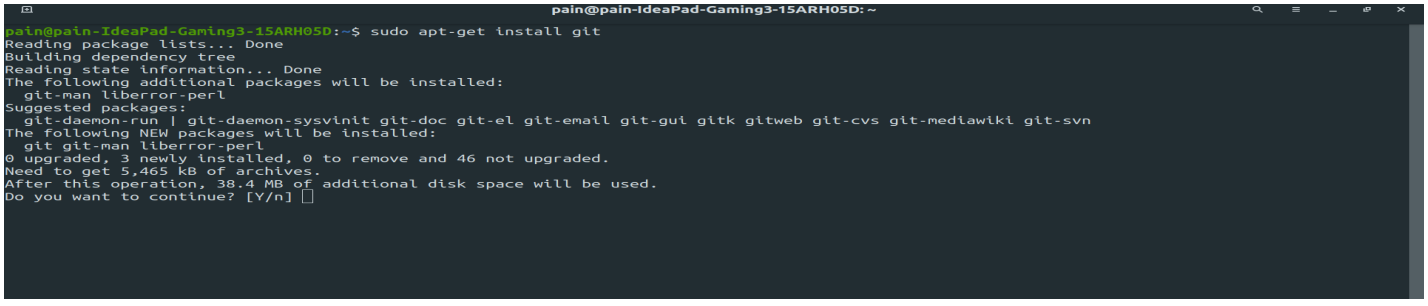


ABOUT MININET:

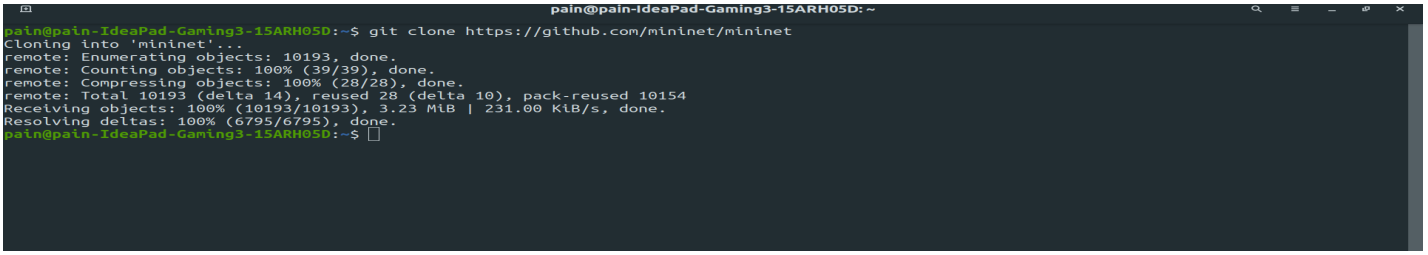


step 3. Install git using command

$sudo apt-get install git



$git clone https://github.com/mininet/mininet



$cd mininet

```
pain@pain-IdeaPad-Gaming3-15ARH05D:~$ cd mininet
pain@pain-IdeaPad-Gaming3-15ARH05D:~/mininet$
```

step 4.To check the available version of git

$git tag # list available

```
pain@pain-IdeaPad-Gaming3-15ARH05D:~/mininet$ git tag # list available versions
1.0.0
2.0.0
2.1.0
2.1.0p1
2.1.0p2
2.2.0
2.2.1
2.2.2
2.3.0
2.3.0b1
2.3.0b2
2.3.0d3
2.3.0d4
2.3.0d5
2.3.0d6
2.3.0rc1
2.3.0rc2
cs244-spring-2012-final
pain@pain-IdeaPad-Gaming3-15ARH05D:~/mininet$
```

$git checkout -b cs244-spring-2012-final

$sudo mn

```
pain@pain-IdeaPad-Gaming3-15ARH05D:~$ sudo mn
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1)
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Starting CLI:
mininet>
```

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

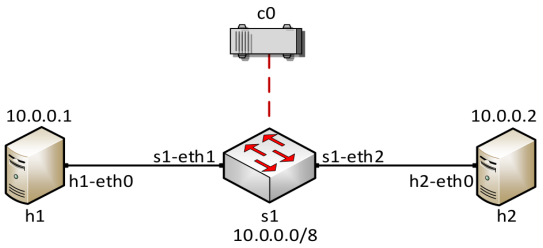


Figure 5. Mininet's default minimal topology.

Step 5. To display the list of Mininet CLI commands and examples on their usage, type the following command:

mininet>help

```
mininet> help

Documented commands (type help <topic>):
========================================
EOF    gterm  iperfudp  nodes        pingpair      py      switch  xterm
dpctl  help   link      noecho       pingpairfull  quit    time
dump   intfs  links     pingall      ports         sh      wait
exit   iperf  net       pingallfull  px            source  x

You may also send a command to a node using:
  <node> command {args}
For example:
  mininet> h1 ifconfig

The interpreter automatically substitutes IP addresses
for node names when a node is the first arg, so commands
like
  mininet> h2 ping h3
should work.

Some character-oriented interactive commands require
noecho:
  mininet> noecho h2 vi foo.py
However, starting up an xterm/gterm is generally better:
  mininet> xterm h2

mininet>
```

Step 6. To display the available nodes, type the following command:
mininet>nodes

```
mininet> nodes
available nodes are:
c0 h1 h2 s1
mininet>
```

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

Step 7. 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.

mininet>net

```
mininet> net
h1 h1-eth0:s1-eth1
h2 h2-eth0:s1-eth2
s1 lo:  s1-eth1:h1-eth0 s1-eth2:h2-eth0
c0
mininet>
```

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 command:
h1 ifconfig

```
c0 h1 h2 s1
mininet> net
h1 h1-eth0:s1-eth1
h2 h2-eth0:s1-eth2
s1 lo:  s1-eth1:h1-eth0 s1-eth2:h2-eth0
c0
mininet> h1 config
bash: config: command not found
mininet> h1 ifconfig
h1-eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.0.1  netmask 255.0.0.0  broadcast 10.255.255.255
        inet6 fe80::4e0:18ff:fe4f:c5cf  prefixlen 64  scopeid 0x20<link>
        ether 06:e0:18:4f:c5:cf  txqueuelen 1000  (Ethernet)
        RX packets 46  bytes 6067 (6.0 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 14  bytes 1076 (1.0 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

mininet>
```

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. 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. To stop the test, press Ctrl+c . The figure below 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.

mininet>h1 ping 10.0.0.1

```
mininet> h1 ping 10.0.0.1
PING 10.0.0.1 (10.0.0.1) 56(84) bytes of data.
64 bytes from 10.0.0.1: icmp_seq=1 ttl=64 time=0.061 ms
64 bytes from 10.0.0.1: icmp_seq=2 ttl=64 time=0.052 ms
64 bytes from 10.0.0.1: icmp_seq=3 ttl=64 time=0.091 ms
64 bytes from 10.0.0.1: icmp_seq=4 ttl=64 time=0.053 ms
64 bytes from 10.0.0.1: icmp_seq=5 ttl=64 time=0.107 ms
64 bytes from 10.0.0.1: icmp_seq=6 ttl=64 time=0.056 ms
64 bytes from 10.0.0.1: icmp_seq=7 ttl=64 time=0.087 ms
64 bytes from 10.0.0.1: icmp_seq=8 ttl=64 time=0.038 ms
64 bytes from 10.0.0.1: icmp_seq=9 ttl=64 time=0.094 ms
64 bytes from 10.0.0.1: icmp_seq=10 ttl=64 time=0.083 ms
^C
--- 10.0.0.1 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9215ms
rtt min/avg/max/mdev = 0.038/0.072/0.107/0.021 ms
mininet>
```

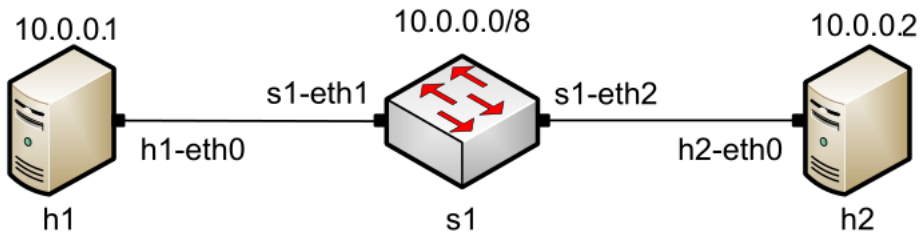
Step 2. Stop the emulation by typing the following command:

mininet>exit

```
mininet> exit
*** Stopping 1 controllers
c0
*** Stopping 2 links
..
*** Stopping 1 switches
s1
*** Stopping 2 hosts
h1 h2
*** Done
completed in 106.241 seconds
pain@pain-IdeaPad-Gaming3-15ARH05D:~$
```

Protocol demonstration

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.



Build the network topology

Step 1. To open miniedit, first redirect to the location where miniedit is present using the commands

$cd mininet/examples

$python2 miniedit.py

```
pain@pain-IdeaPad-Gaming3-15ARH05D:~$ cd mininet/examples
pain@pain-IdeaPad-Gaming3-15ARH05D:~/mininet/examples$ python2 miniedit.py
```

miniedit will be illustrated as below

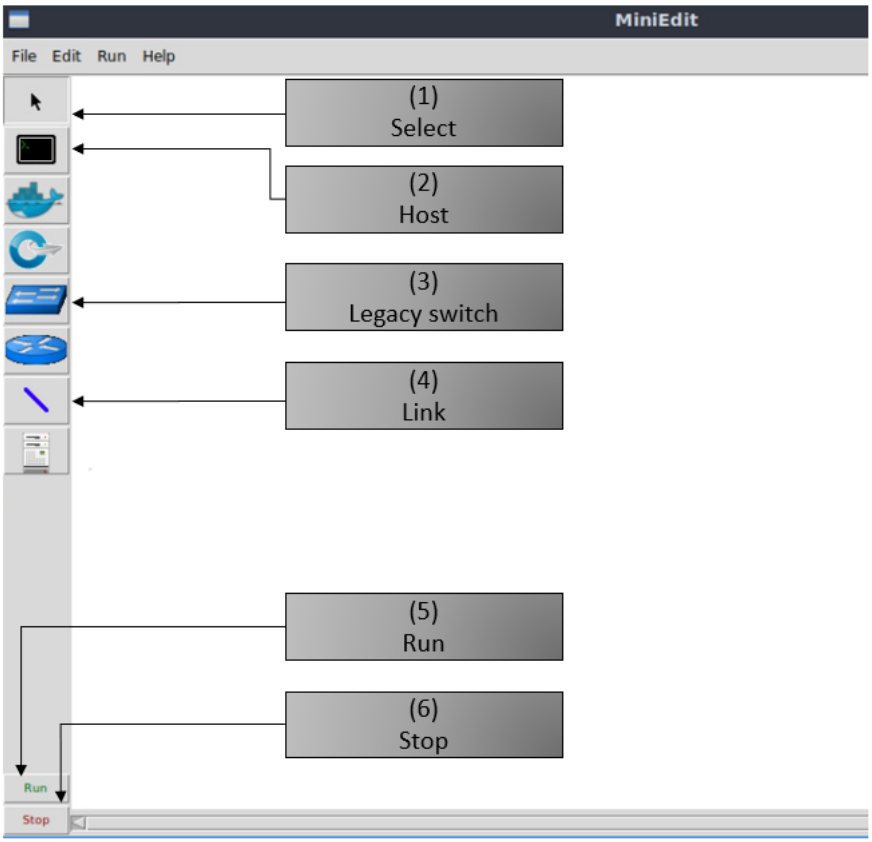


Figure 14. MiniEdit Graphical User Interface (GUI).

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.

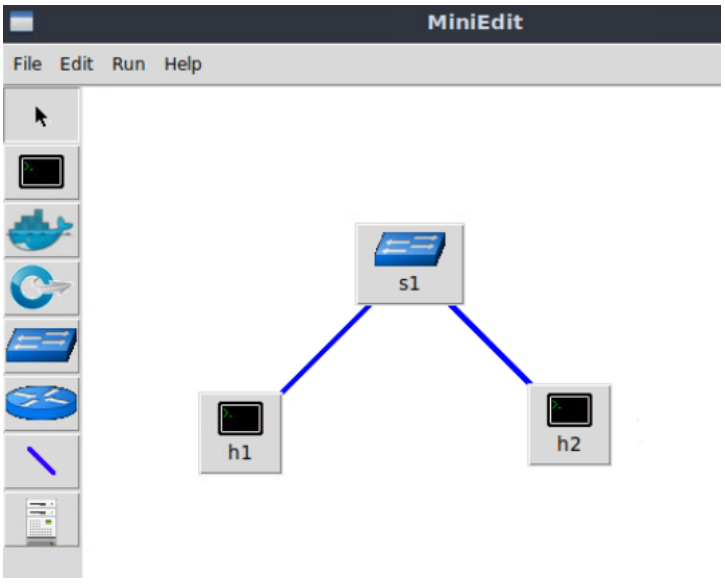


Figure 15. MiniEdit's topology.

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.

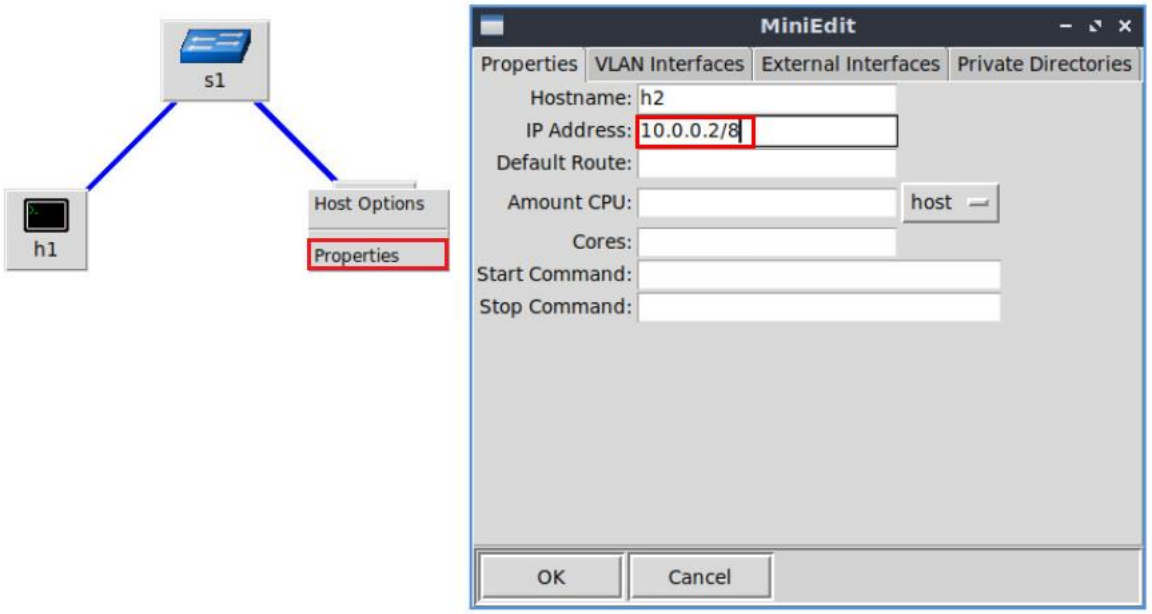


Figure 16. Configuration of a host's properties.

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.

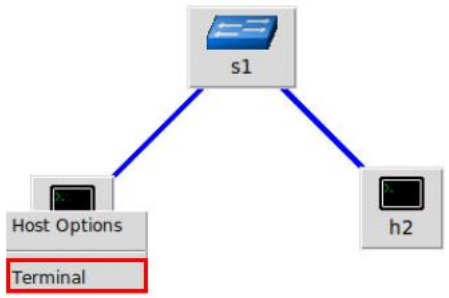
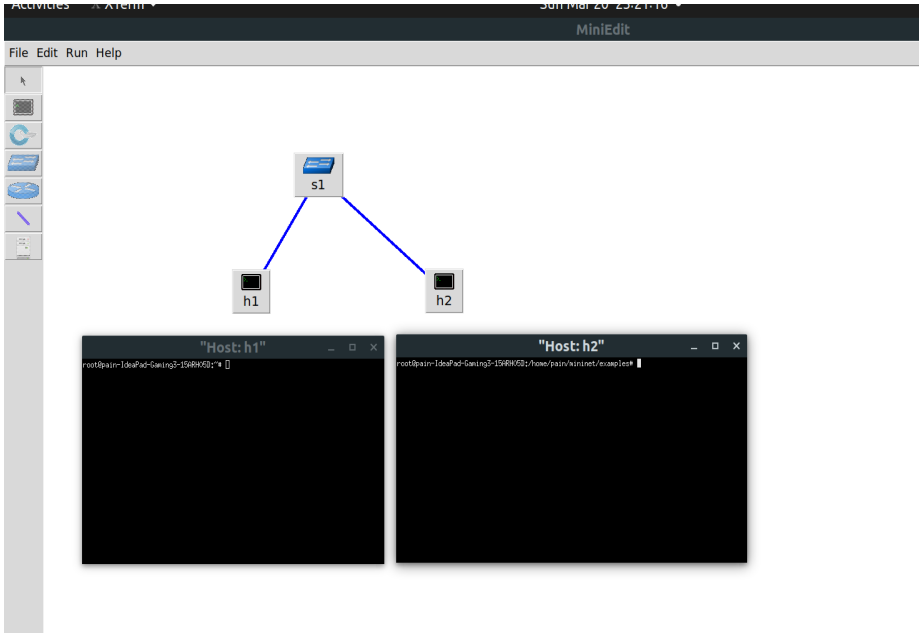


Figure 18. Opening a terminal on host h1.

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



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

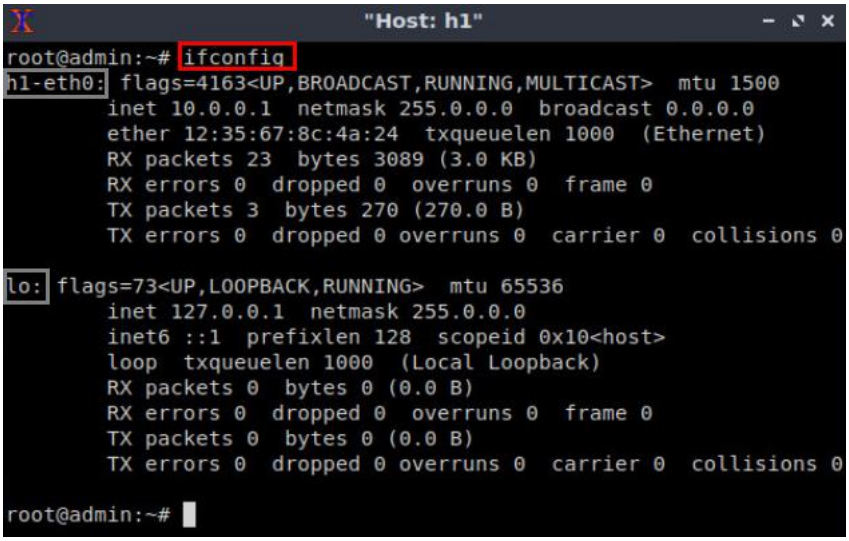


Figure 20. Output of ifconfig command on host h1.

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

$ping 10.0.0.2

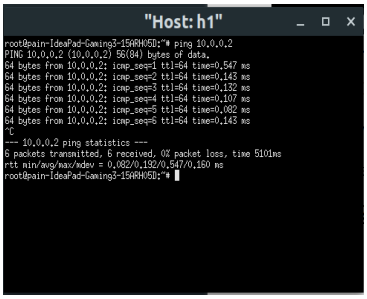


Figure 21. Connectivity test using ping command.

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

Figure 22. Stopping the emulation.

References:

1. Mininet walkthrough. [Online]. Available: http://Mininet.org.
2. N. Mckeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "OpenFlow," ACM SIGCOMM Computer Communication Review, vol. 38, no. 2, p. 69, 2008.
3. J. Esch, "Prolog to, software-defined networking: a comprehensive survey," Proceedings of the IEEE, vol. 103, no. 1, pp. 10–13, 2015.
4. P. Dordal, "An Introduction to computer networks,". [Online]. Available:https://intronetworks.cs.luc.edu/.
5. B. Lantz, G. Gee, "MiniEdit: a simple network editor for Mininet," 2013. [Online]. Available: https://github.com/Mininet/Mininet/blob/master/examples.