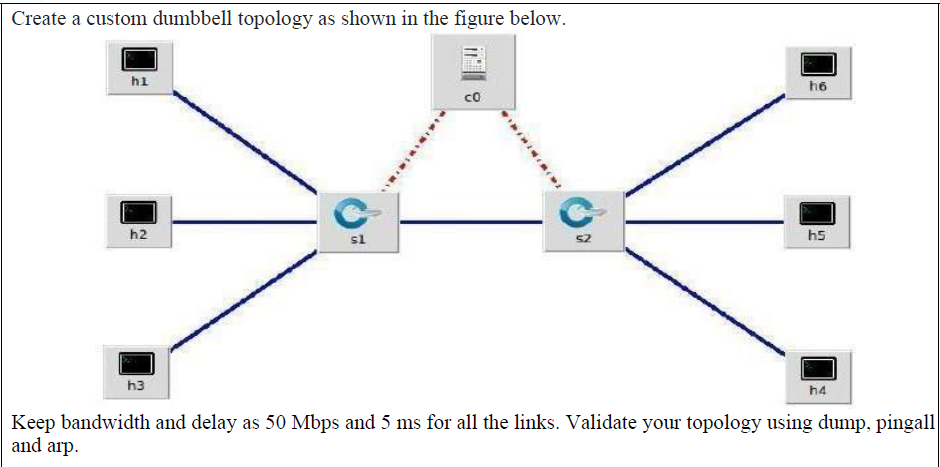
**PRACTICAL-4**

**AIM:**



**THEORY:**

**Bandwidth:**

* The maximum amount of data transmitted over an internet connection in a given amount of time

**Mbps:**

* Megabits per second (Mbps) are units of measurement for network [bandwidth](https://www.techtarget.com/searchnetworking/definition/bandwidth) and [throughput](https://www.techtarget.com/searchnetworking/definition/throughput). They are used to show how fast a network or internet connection is.

**Dump:**

* It lists information about the nodes, switches and controllers in the simulated network

**pingall:**

* It displays the connectivity between all hosts and tells us which hosts are connected to each other

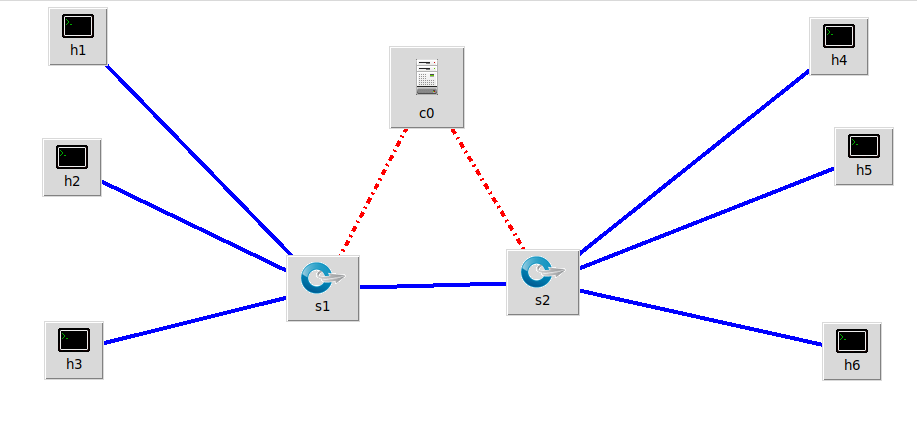
**dpctl:**

* It is used to view the flows in switch table

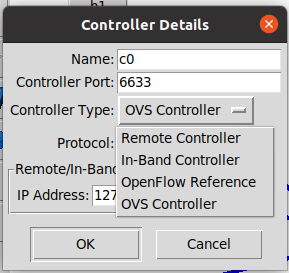
**ARP:**

* The Address Resolution Protocol is a communication protocol used for discovering the link layer address, such as a MAC address, associated with a given internet layer address, typically an IPv4 address.

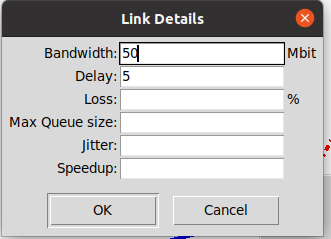
**TOPOLOGY:**



* Change the Controller Type of Controller to OVS Controller.

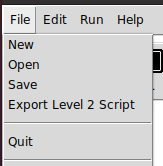


* Now, click on the blue link, and it turns green, then right click it and select properties.
* Enter the bandwidth as 50 Mbps and delay is 5 milli seconds.



* Do the same for all the link.
* Save the file.
* Then, click on file and select Export Level 2 script.
* It will generate the python file.

**Note: we can create the same topology using python code.**



**PYTHON CODE:**

#!/usr/bin/env python

from mininet.net import Mininet

from mininet.node import Controller, RemoteController, OVSController

from mininet.node import CPULimitedHost, Host, Node

from mininet.node import OVSKernelSwitch, UserSwitch

from mininet.node import IVSSwitch

from mininet.cli import CLI

from mininet.log import setLogLevel, info

from mininet.link import TCLink, Intf

from subprocess import call

def myNetwork():

net = Mininet( topo=None,

build=False,

ipBase='10.0.0.0/8')

info( '\*\*\* Adding controller\n' )

c0=net.addController(name='c0',

controller=OVSController,

protocol='tcp',

port=6633)

info( '\*\*\* Add switches\n')

s1 = net.addSwitch('s1', cls=OVSKernelSwitch)

s2 = net.addSwitch('s2', cls=OVSKernelSwitch)

info( '\*\*\* Add hosts\n')

h1 = net.addHost('h1', cls=Host, ip='10.10.10.2', defaultRoute=None)

h2 = net.addHost('h2', cls=Host, ip='10.10.10.3', defaultRoute=None)

h3 = net.addHost('h3', cls=Host, ip='10.10.10.4', defaultRoute=None)

h4 = net.addHost('h4', cls=Host, ip='10.10.10.7', defaultRoute=None)

h5 = net.addHost('h5', cls=Host, ip='10.10.10.6', defaultRoute=None)

h6 = net.addHost('h6', cls=Host, ip='10.10.10.5', defaultRoute=None)

info( '\*\*\* Add links\n')

s2s1 = {'bw':50,'delay':'5'}

net.addLink(s2, s1, cls=TCLink , \*\*s2s1)

h1s1 = {'bw':50,'delay':'5'}

net.addLink(h1, s1, cls=TCLink , \*\*h1s1)

h2s1 = {'bw':50,'delay':'5'}

net.addLink(h2, s1, cls=TCLink , \*\*h2s1)

h3s1 = {'bw':50,'delay':'5'}

net.addLink(h3, s1, cls=TCLink , \*\*h3s1)

s2h4 = {'bw':50,'delay':'5'}

net.addLink(s2, h4, cls=TCLink , \*\*s2h4)

s2h5 = {'bw':50,'delay':'5'}

net.addLink(s2, h5, cls=TCLink , \*\*s2h5)

s2h6 = {'bw':50,'delay':'5'}

net.addLink(s2, h6, cls=TCLink , \*\*s2h6)

info( '\*\*\* Starting network\n')

net.build()

info( '\*\*\* Starting controllers\n')

for controller in net.controllers:

controller.start()

info( '\*\*\* Starting switches\n')

net.get('s1').start([c0])

net.get('s2').start([c0])

info( '\*\*\* Post configure switches and hosts\n')

CLI(net)

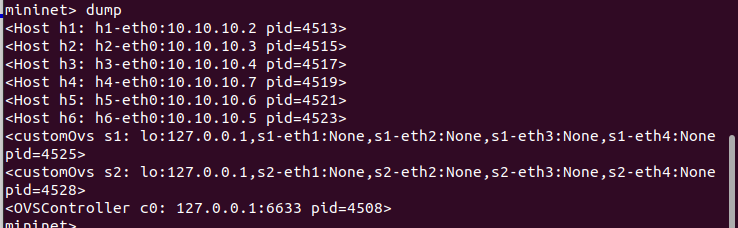
net.stop()

if \_\_name\_\_ == '\_\_main\_\_':

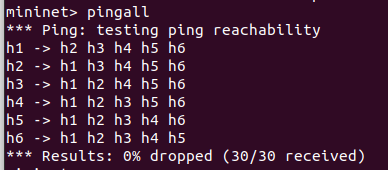
setLogLevel( 'info' )

myNetwork()

* Now, we will check for the dump command.



* Now, we will check for pingall command.

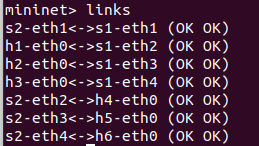


* Now, we will check for arp command.
* Syntax: <hostname> arp

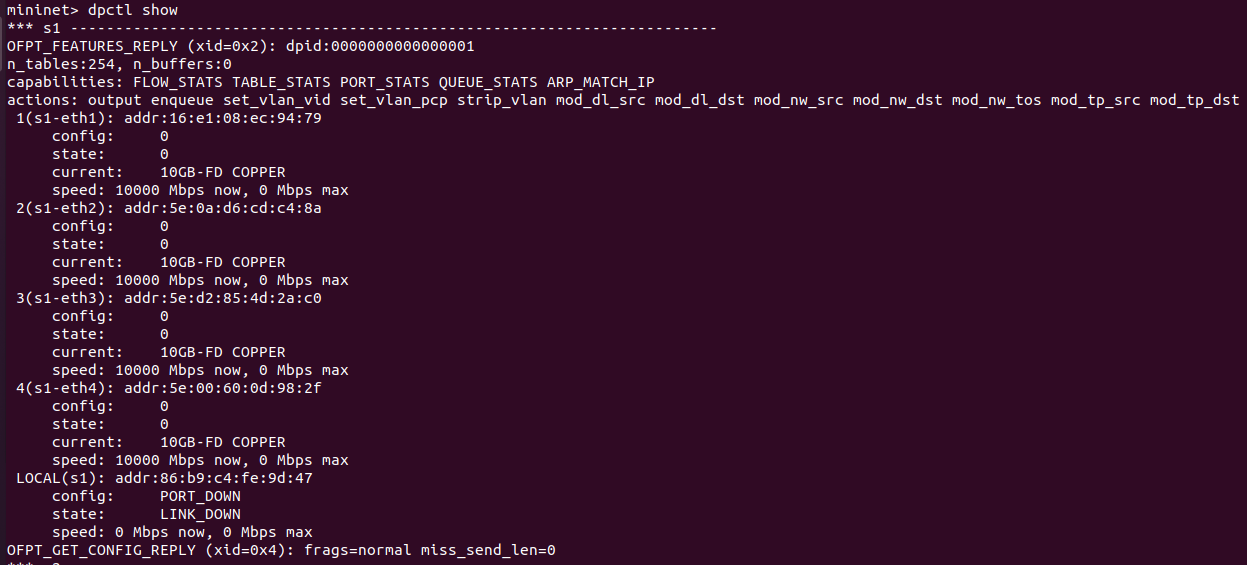


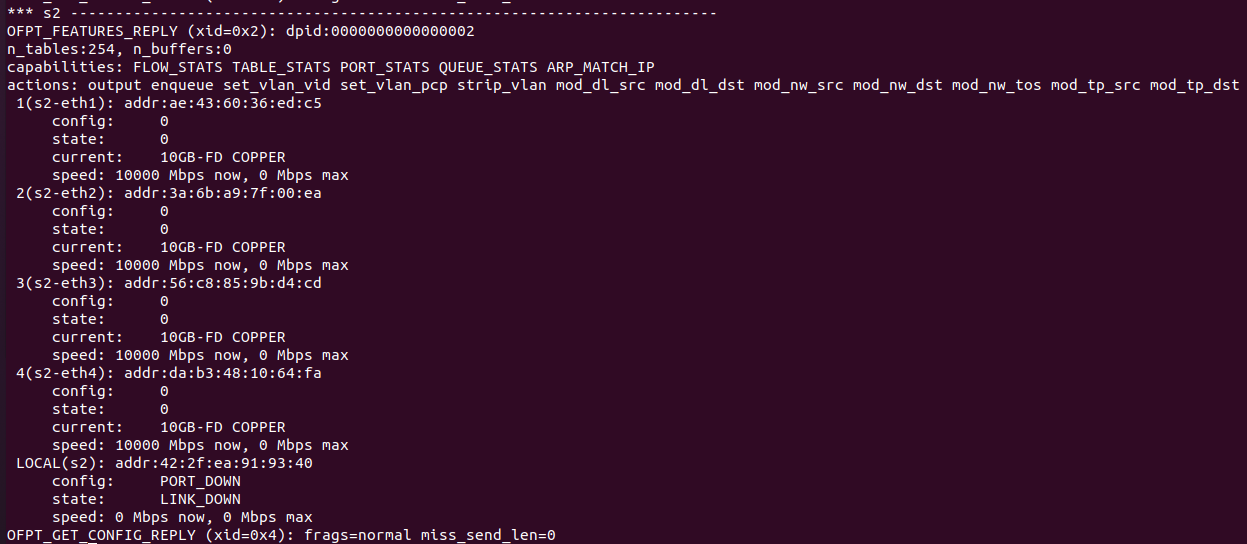
* Now, we will perform the commands for iperf and link



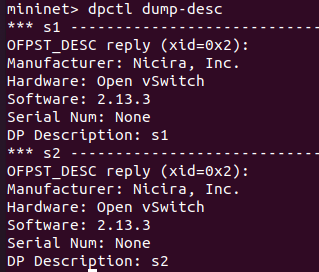


* Command: dpctl show

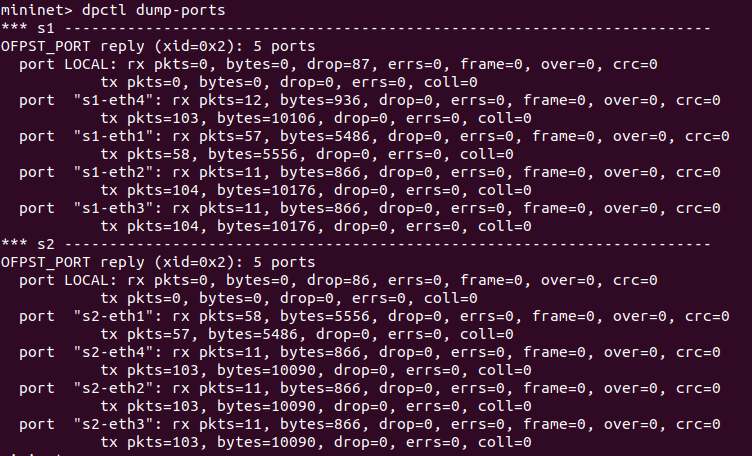




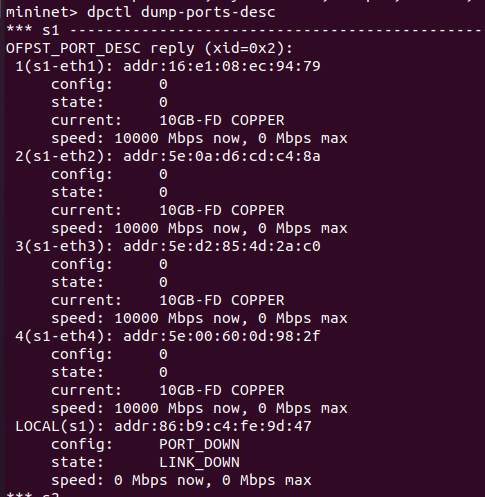
* Command: dpctl dump-desc



* Command: dpctl dump-ports



* Command: dpctl dump-ports-desc



**CONCLUSION:**

* By performing the above practical, I learnt how to create topology in mini-edit, how to execute the commands of pingall, arp, dump.
* I also learnt about how to alter the link configuration between the devices.