MININET

TEAM - 121AD0012, 121AD0014, 121AD0019, 121AD0020

INTRODUCTION:

Mininet is a popular tool for emulating networks on a single machine. It's often used for testing, development, and education purposes.
Mininet supports Software-Defined Networking (SDN) technologies, allowing users to develop, experiment, and prototype SDN applications in a controlled and cost-effective manner.

HOW TO INSTALL AND USE:

Using the command : `sudo apt-get install mininet`
We can install mininet. Now to run a custom mininet, we may use the python code like follows:

```
from mininet.topo import Topo
from mininet.net import Mininet
from mininet.cli import CLI

class CustomTopology(Topo):
    def build(self):
        # Add switches
        s1 = self.addSwitch('s1')
        s2 = self.addSwitch('s2')

        # Add hosts
        h1 = self.addHost('h1')
        h2 = self.addHost('h2')
        h3 = self.addHost('h3')

        # Add links between hosts and switches
        self.addLink(h1, s1)
        self.addLink(h2, s1)
        self.addLink(h3, s2)

        # Add link between switches
        self.addLink(s1, s2)

if __name__ == '__main__':
    # Instantiate custom topology
    topo = CustomTopology()

    # Create Mininet object with custom topology
    net = Mininet(topo)

    # Start the network
    net.start()
```

```
# Optionally, interact with the network
CLI(net)
# Stop the network
net.stop()
```

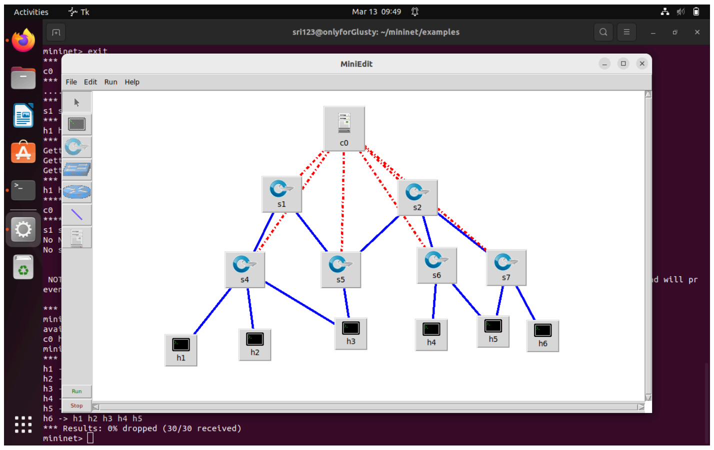
This code is for two switches and two hosts, with an openflow controller. But, in our terminal, we faced an error regarding not being able to operate the code or run it. We even tried changing the controller to Ryu from OpenFlow. But still, the error is persisting. We have not yet traced out why that error occurred. Hence we will switch to using the GUI which is way more convenient than running the code.

MINIEDIT GUI:

This miniedit GUI is present in the folder mininet/examples.

```
cd mininet/examples
sudo python miniedit.py
```

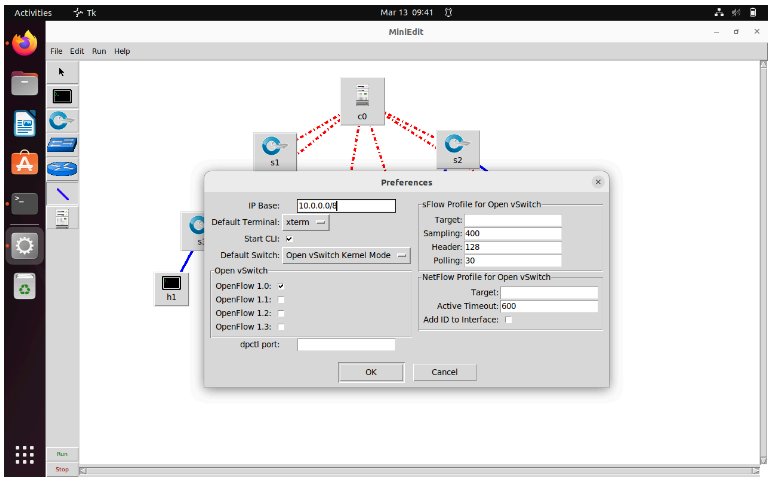
On running this we can see the GUI. It consists of a palette having Selection, Host, Switch, Legacy switch, Router, Communication Link and Controller. We can select the one we want to and create a topology of our wish. One is as shown below.



```
h6 -> h1 h2 h3 h4 h5
*** Results: 0% dropped (30/30 received)
mininet>
```

With this, we need to make some changes in its settings as follows before running it:

1. Go to edit → Preferences. Toggle the Start CLI option. Keep other settings as it is.



IP Base: 10.0.0.0/8
Default Terminal: xterm
Start CLI: ✓
Default Switch: Open vSwitch Kernel Mode

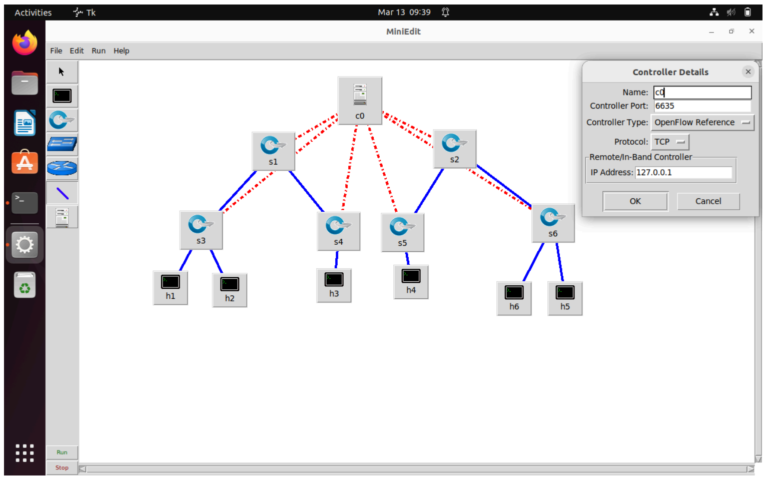
Open vSwitch
OpenFlow 1.0: ✓
OpenFlow 1.1: ☐
OpenFlow 1.2: ☐
OpenFlow 1.3: ☐

dpctl port:

sFlow Profile for Open vSwitch
Target:
Sampling: 400
Header: 128
Polling: 30

NetFlow Profile for Open vSwitch
Target:
Active Timeout: 600
Add ID to Interface: ☐

2. Right click on the controller and change the port number:

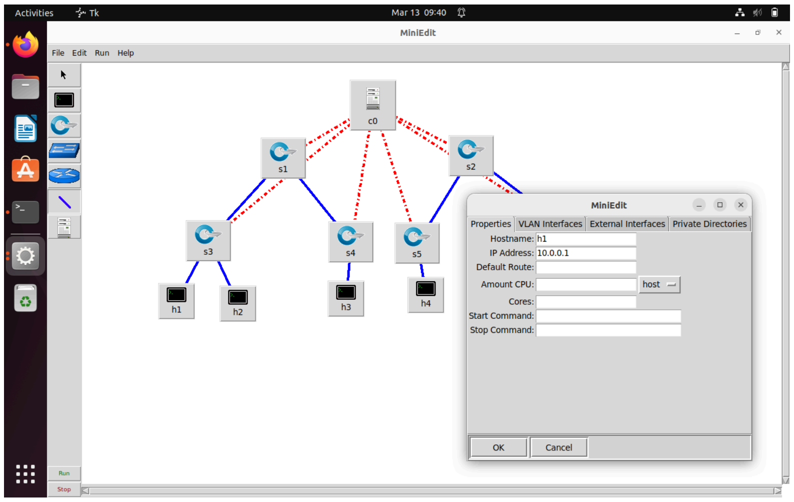


Controller Details

Name: c0
Controller Port: 6635
Controller Type: OpenFlow Reference
Protocol: TCP
Remote/In-Band Controller
IP Address: 127.0.0.1

OK Cancel

3. For each host, assign the IP address according to the subnet given in the preferences.



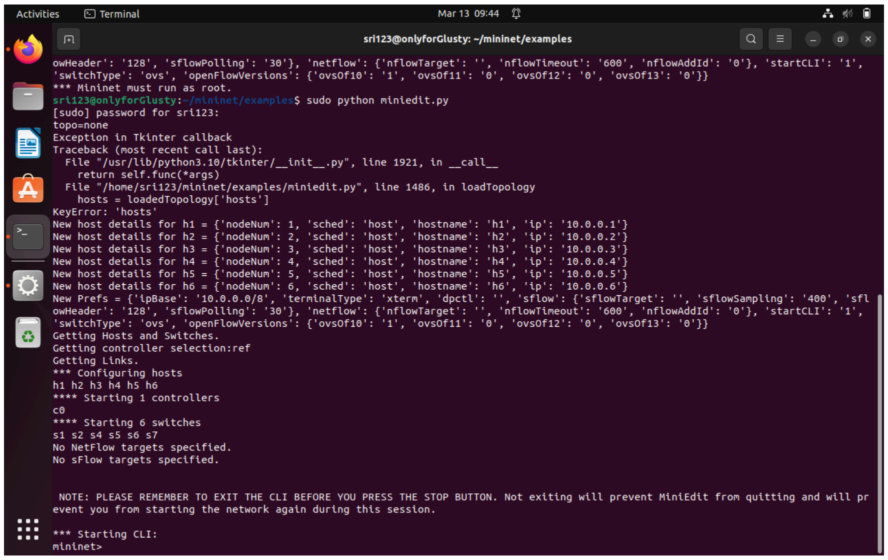
MiniEdit

Properties | VLAN Interfaces | External Interfaces | Private Directories

Hostname: h1
IP Address: 10.0.0.1
Default Route:
Amount CPU: host
Cores:
Start Command:
Stop Command:

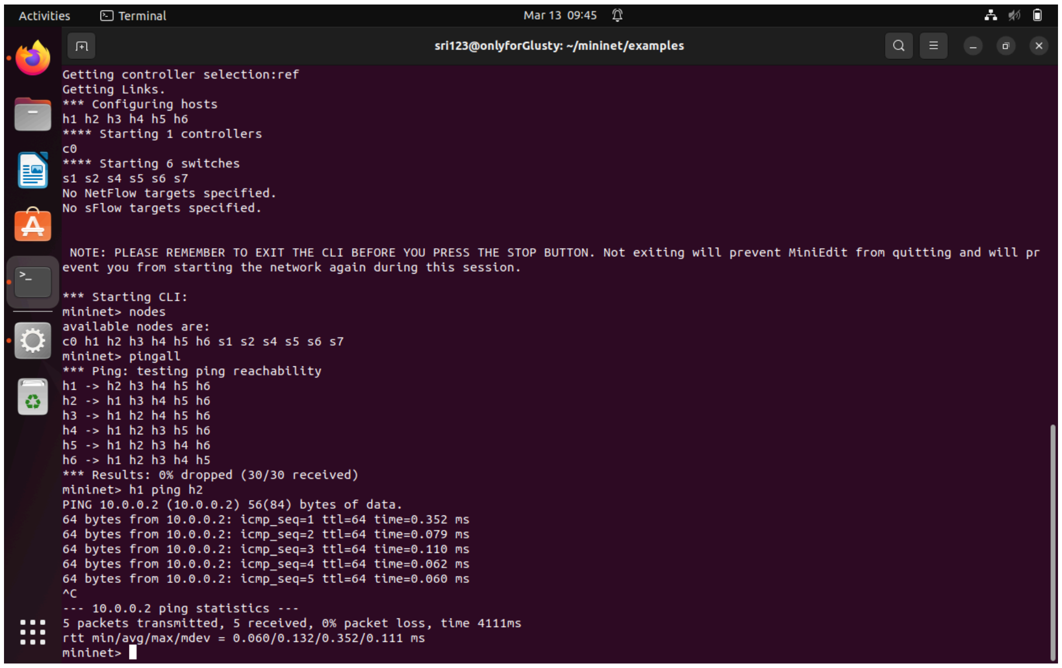
OK Cancel

4. Click on the option run on the top or, we can click the run button on the bottom left.



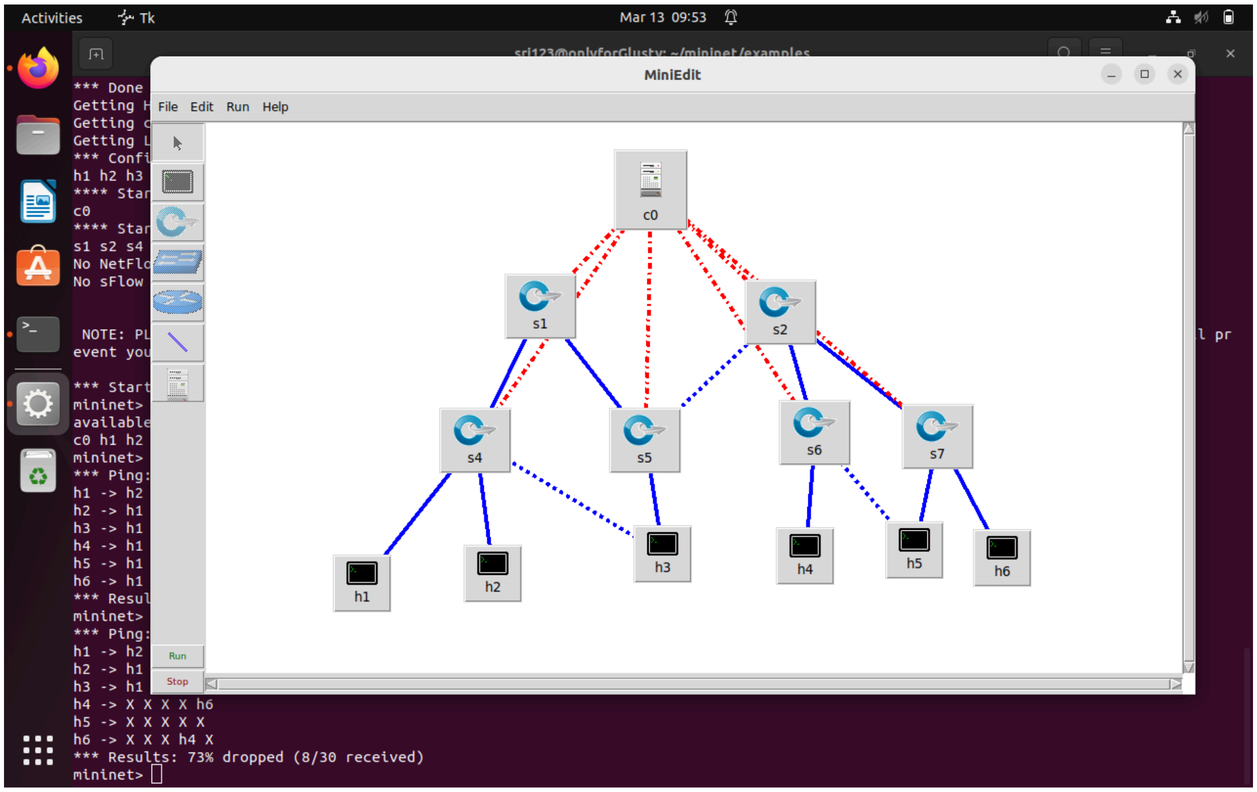
```
owHeader': '128', 'sflowPolling': '30'}, 'netflow': {'nflowTarget': '', 'nflowTimeout': '600', 'nflowAddId': '0'}, 'startCLI': '1',
'switchType': 'ovs', 'openFlowVersions': {'ovsOf10': '1', 'ovsOf11': '0', 'ovsOf12': '0', 'ovsOf13': '0'}}
*** Mininet must run as root.
sri123@onlyforGlusty:~/mininet/examples$ sudo python miniedit.py
[sudo] password for sri123:
topo=none
Exception in Tkinter callback
Traceback (most recent call last):
  File "/usr/lib/python3.10/tkinter/__init__.py", line 1921, in __call__
    return self.func(*args)
  File "/home/sri123/mininet/examples/miniedit.py", line 1486, in loadTopology
    hosts = loadedTopology['hosts']
KeyError: 'hosts'
New host details for h1 = {'nodeNum': 1, 'sched': 'host', 'hostname': 'h1', 'ip': '10.0.0.1'}
New host details for h2 = {'nodeNum': 2, 'sched': 'host', 'hostname': 'h2', 'ip': '10.0.0.2'}
New host details for h3 = {'nodeNum': 3, 'sched': 'host', 'hostname': 'h3', 'ip': '10.0.0.3'}
New host details for h4 = {'nodeNum': 4, 'sched': 'host', 'hostname': 'h4', 'ip': '10.0.0.4'}
New host details for h5 = {'nodeNum': 5, 'sched': 'host', 'hostname': 'h5', 'ip': '10.0.0.5'}
New host details for h6 = {'nodeNum': 6, 'sched': 'host', 'hostname': 'h6', 'ip': '10.0.0.6'}
New Prefs = {'ipBase': '10.0.0.0/8', 'terminalType': 'xterm', 'dpctl': '', 'sflow': {'sflowTarget': '', 'sflowSampling': '400', 'sfl
owHeader': '128', 'sflowPolling': '30'}, 'netflow': {'nflowTarget': '', 'nflowTimeout': '600', 'nflowAddId': '0'}, 'startCLI': '1',
'switchType': 'ovs', 'openFlowVersions': {'ovsOf10': '1', 'ovsOf11': '0', 'ovsOf12': '0', 'ovsOf13': '0'}}
Getting Hosts and Switches.
Getting controller selection:ref
Getting Links.
*** Configuring hosts
h1 h2 h3 h4 h5 h6
**** Starting 1 controllers
c0
**** Starting 6 switches
s1 s2 s4 s5 s6 s7
No NetFlow targets specified.
No sFlow targets specified.


 NOTE: PLEASE REMEMBER TO EXIT THE CLI BEFORE YOU PRESS THE STOP BUTTON. Not exiting will prevent MiniEdit from quitting and will pr
event you from starting the network again during this session.

*** Starting CLI:
mininet>
```

Let’s now check the connectivity of our topology using the pingall command:



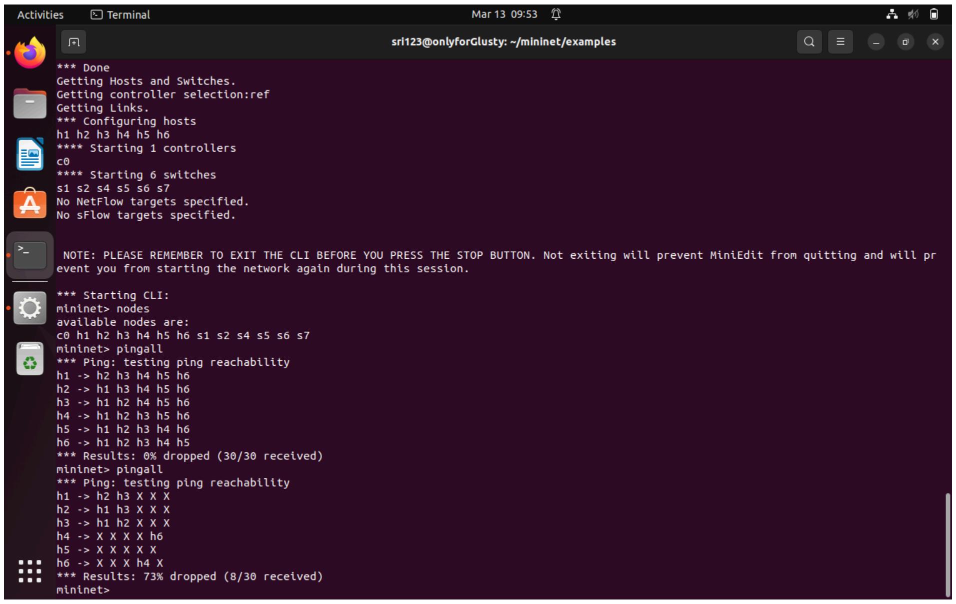
```
Getting controller selection:ref
Getting Links.
*** Configuring hosts
h1 h2 h3 h4 h5 h6
**** Starting 1 controllers
c0
**** Starting 6 switches
s1 s2 s4 s5 s6 s7
No NetFlow targets specified.
No sFlow targets specified.


 NOTE: PLEASE REMEMBER TO EXIT THE CLI BEFORE YOU PRESS THE STOP BUTTON. Not exiting will prevent MiniEdit from quitting and will pr
event you from starting the network again during this session.

*** Starting CLI:
mininet> nodes
available nodes are:
c0 h1 h2 h3 h4 h5 h6 s1 s2 s4 s5 s6 s7
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 h3 h4 h5 h6
h2 -> h1 h3 h4 h5 h6
h3 -> h1 h2 h4 h5 h6
h4 -> h1 h2 h3 h5 h6
h5 -> h1 h2 h3 h4 h6
h6 -> h1 h2 h3 h4 h5
*** Results: 0% dropped (30/30 received)
mininet> h1 ping h2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=0.352 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.079 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.110 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.062 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.060 ms
^C
--- 10.0.0.2 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4111ms
rtt min/avg/max/mdev = 0.060/0.132/0.352/0.111 ms
mininet>
```

Now go to the GUI again, try deactivating a link. We can do so by clicking on the link, Down Link. It now becomes a dotted line instead of a full line.



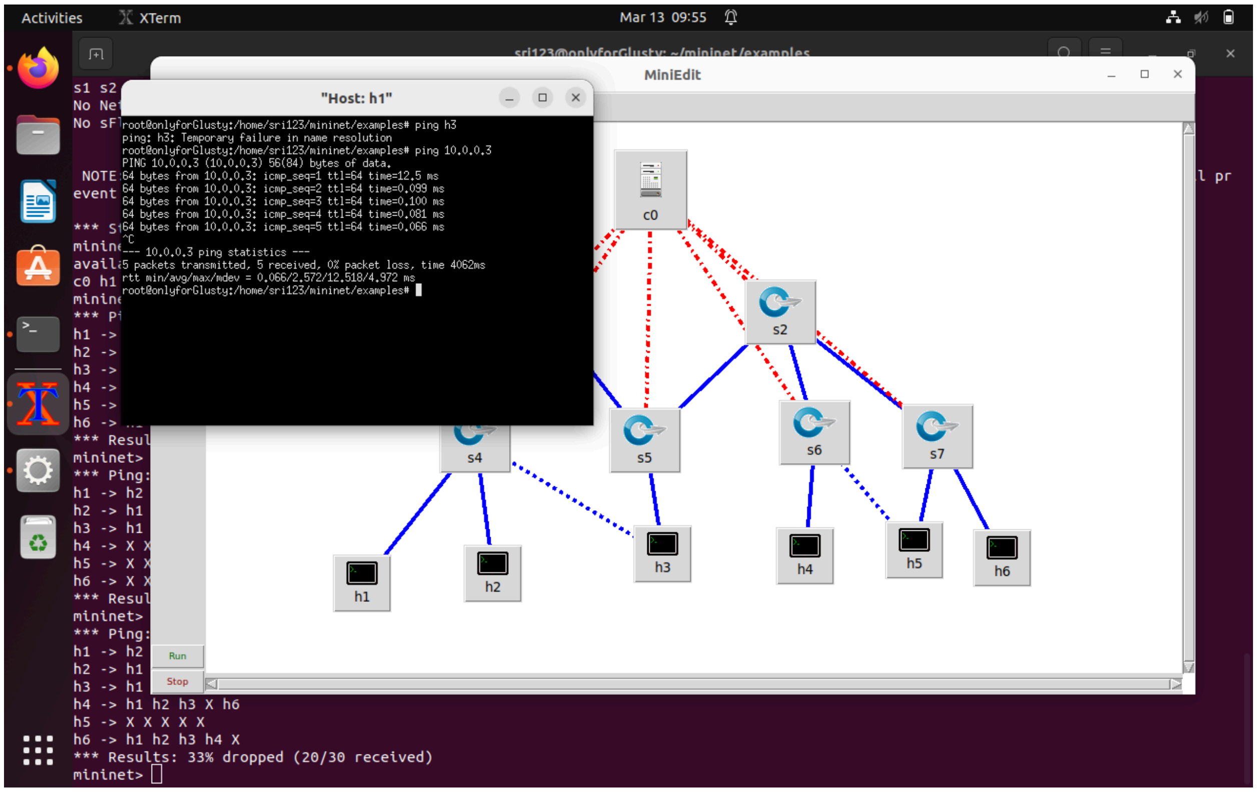
```
*** Done
Getting H
Getting c
Getting L
*** Confi
h1 h2 h3
**** Star
c0
**** Star
s1 s2 s4
No NetFlo
No sFlow


 NOTE: PL                                                                                  l pr
event you

*** Start
mininet>
available
c0 h1 h2
mininet>
*** Ping:
h1 -> h2
h2 -> h1
h3 -> h1
h4 -> h1
h5 -> h1
h6 -> h1
*** Resul
mininet>
*** Ping:
h1 -> h2
h2 -> h1
h3 -> h1
h4 -> X X X X h6
h5 -> X X X X X
h6 -> X X X h4 X
*** Results: 73% dropped (8/30 received)
mininet>
```

Now we run the same pingall command:



```
*** Done
Getting Hosts and Switches.
Getting controller selection:ref
Getting Links.
*** Configuring hosts
h1 h2 h3 h4 h5 h6
**** Starting 1 controllers
c0
**** Starting 6 switches
s1 s2 s4 s5 s6 s7
No NetFlow targets specified.
No sFlow targets specified.


 NOTE: PLEASE REMEMBER TO EXIT THE CLI BEFORE YOU PRESS THE STOP BUTTON. Not exiting will prevent MiniEdit from quitting and will pr
event you from starting the network again during this session.

*** Starting CLI:
mininet> nodes
available nodes are:
c0 h1 h2 h3 h4 h5 h6 s1 s2 s4 s5 s6 s7
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 h3 h4 h5 h6
h2 -> h1 h3 h4 h5 h6
h3 -> h1 h2 h4 h5 h6
h4 -> h1 h2 h3 h5 h6
h5 -> h1 h2 h3 h4 h6
h6 -> h1 h2 h3 h4 h5
*** Results: 0% dropped (30/30 received)
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 h3 X X X
h2 -> h1 h3 X X X
h3 -> h1 h2 X X X
h4 -> X X X X h6
h5 -> X X X X X
h6 -> X X X h4 X
*** Results: 73% dropped (8/30 received)
mininet>
```

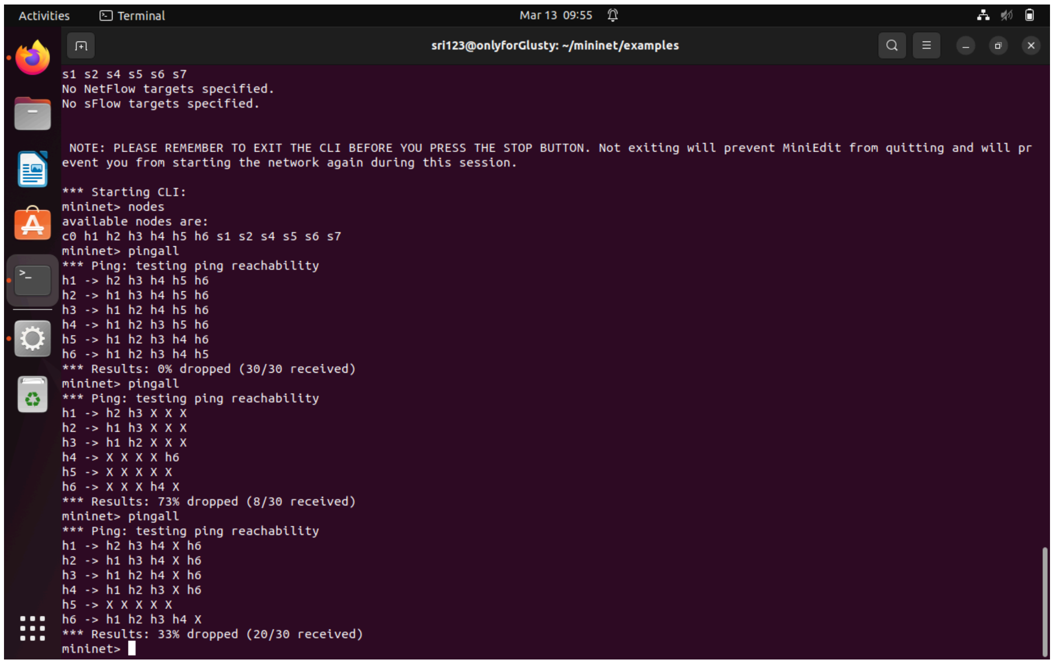
We can see that now connectivity has changed. We can check the connectivity by using xterm. Right click on any one of the hosts and terminal. Now ping to any other host ip address.



```
root@onlyforGlusty:/home/sri123/mininet/examples# ping h3
ping: h3: Temporary failure in name resolution
root@onlyforGlusty:/home/sri123/mininet/examples# ping 10.0.0.3
PING 10.0.0.3 (10.0.0.3) 56(84) bytes of data.
64 bytes from 10.0.0.3: icmp_seq=1 ttl=64 time=12.5 ms
64 bytes from 10.0.0.3: icmp_seq=2 ttl=64 time=0.099 ms
64 bytes from 10.0.0.3: icmp_seq=3 ttl=64 time=0.100 ms
64 bytes from 10.0.0.3: icmp_seq=4 ttl=64 time=0.081 ms
64 bytes from 10.0.0.3: icmp_seq=5 ttl=64 time=0.066 ms
^C
--- 10.0.0.3 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4062ms
rtt min/avg/max/mdev = 0.066/2.572/12.518/4.972 ms
root@onlyforGlusty:/home/sri123/mininet/examples#
```

```
h4 -> h1 h2 h3 X h6
h5 -> X X X X X
h6 -> h1 h2 h3 h4 X
*** Results: 33% dropped (20/30 received)
mininet>
```

Let us try to activate the communication link between the switch s2 and s5. This will give us the pingall as follows. Of course now the h4 and h6 are accessible, yet h5 is still detached from all. This shows how an openflow connector works by changing and updating the flow rules.



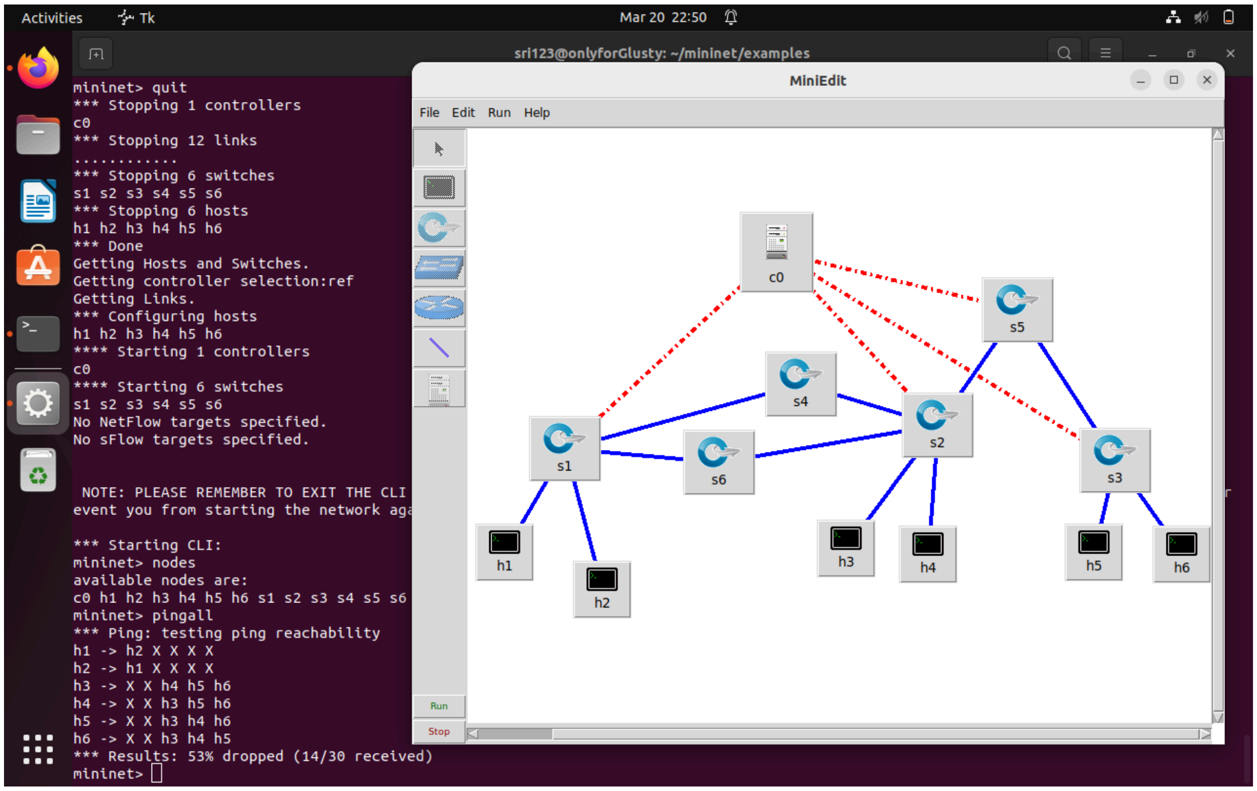
```
s1 s2 s4 s5 s6 s7
No NetFlow targets specified.
No sFlow targets specified.


 NOTE: PLEASE REMEMBER TO EXIT THE CLI BEFORE YOU PRESS THE STOP BUTTON. Not exiting will prevent MiniEdit from quitting and will pr
event you from starting the network again during this session.

*** Starting CLI:
mininet> nodes
available nodes are:
c0 h1 h2 h3 h4 h5 h6 s1 s2 s4 s5 s6 s7
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 h3 h4 h5 h6
h2 -> h1 h3 h4 h5 h6
h3 -> h1 h2 h4 h5 h6
h4 -> h1 h2 h3 h5 h6
h5 -> h1 h2 h3 h4 h6
h6 -> h1 h2 h3 h4 h5
*** Results: 0% dropped (30/30 received)
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 h3 X X X
h2 -> h1 h3 X X X
h3 -> h1 h2 X X X
h4 -> X X X X h6
h5 -> X X X X X
h6 -> X X X h4 X
*** Results: 73% dropped (8/30 received)
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 h3 h4 X h6
h2 -> h1 h3 h4 X h6
h3 -> h1 h2 h4 X h6
h4 -> h1 h2 h3 X h6
h5 -> X X X X X
h6 -> h1 h2 h3 h4 X
*** Results: 33% dropped (20/30 received)
mininet> 
```

Some of the common commands that can be checked upon any of the topologies:

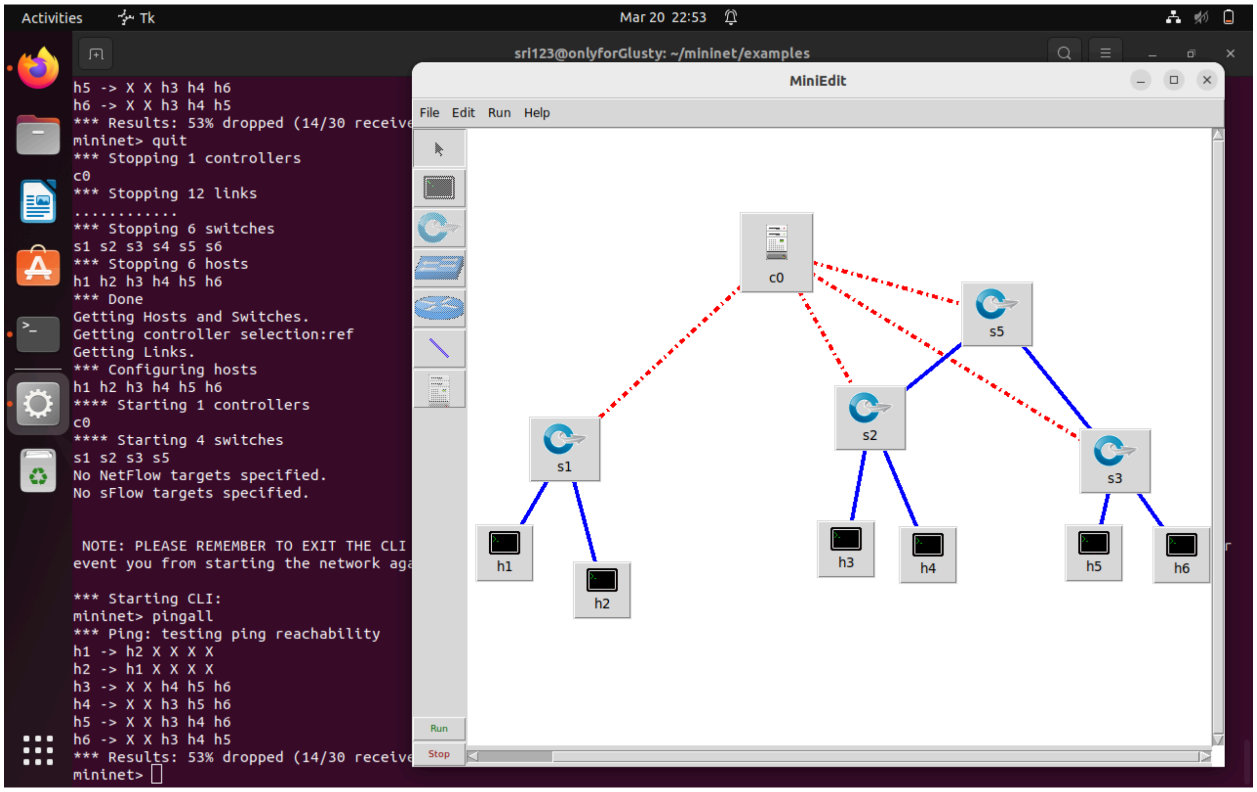
- **nodes**: Lists all the nodes in the network.
- **net**: Displays network topology.
- **pingall**: Tests connectivity between all nodes using ICMP ping.
- **xterm <node>:** Opens an xterm terminal for the specified node. This allows you to interact with the node directly.
- **exit** or **quit**: Exits Mininet.

In order to get the flow rules, we cannot get them through mininet GUI. We need to use other controllers such as POX.

Another topology tried:



```
mininet> quit
*** Stopping 1 controllers
c0
*** Stopping 12 links
............
*** Stopping 6 switches
s1 s2 s3 s4 s5 s6
*** Stopping 6 hosts
h1 h2 h3 h4 h5 h6
*** Done
Getting Hosts and Switches.
Getting controller selection:ref
Getting Links.
*** Configuring hosts
h1 h2 h3 h4 h5 h6
**** Starting 1 controllers
c0
**** Starting 6 switches
s1 s2 s3 s4 s5 s6
No NetFlow targets specified.
No sFlow targets specified.


 NOTE: PLEASE REMEMBER TO EXIT THE CLI
event you from starting the network aga

*** Starting CLI:
mininet> nodes
available nodes are:
c0 h1 h2 h3 h4 h5 h6 s1 s2 s3 s4 s5 s6
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 X X X X
h2 -> h1 X X X X
h3 -> X X h4 h5 h6
h4 -> X X h3 h5 h6
h5 -> X X h3 h4 h6
h6 -> X X h3 h4 h5
*** Results: 53% dropped (14/30 received)
mininet> 
```



```
h5 -> X X h3 h4 h6
h6 -> X X h3 h4 h5
*** Results: 53% dropped (14/30 receive
mininet> quit
*** Stopping 1 controllers
c0
*** Stopping 12 links
............
*** Stopping 6 switches
s1 s2 s3 s4 s5 s6
*** Stopping 6 hosts
h1 h2 h3 h4 h5 h6
*** Done
Getting Hosts and Switches.
Getting controller selection:ref
Getting Links.
*** Configuring hosts
h1 h2 h3 h4 h5 h6
**** Starting 1 controllers
c0
**** Starting 4 switches
s1 s2 s3 s5
No NetFlow targets specified.
No sFlow targets specified.


 NOTE: PLEASE REMEMBER TO EXIT THE CLI
event you from starting the network aga

*** Starting CLI:
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 X X X X
h2 -> h1 X X X X
h3 -> X X h4 h5 h6
h4 -> X X h3 h5 h6
h5 -> X X h3 h4 h6
h6 -> X X h3 h4 h5
*** Results: 53% dropped (14/30 receive
mininet> 
```