



INDIAN INSTITUTE OF INFORMATION TECHNOLOGY,
DESIGN AND MANUFACTURING,
KANCHEEPURAM

DCN LAB-7

NAME:K.Nithesh RollNo:ESD19I008

AIM: To perform Miniedit, Topologies, Packet loss and Delay Analysis

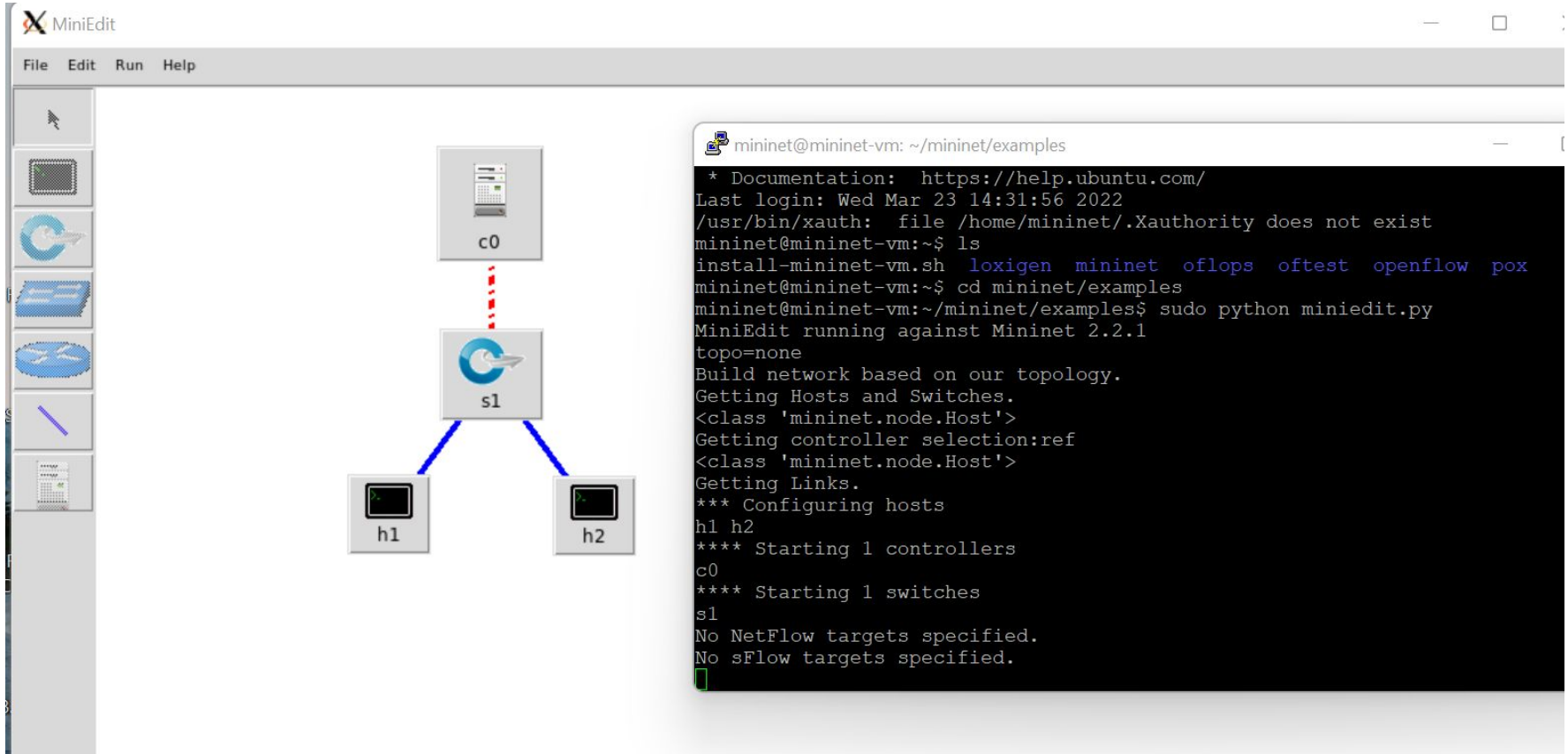
Theory:

Topology: Network topology describes **the physical and logical relationship of nodes in a network, the schematic arrangement of the links and nodes, or some hybrid combination**

Packet loss: Packet loss describes **packets of data not reaching their destination after being transmitted across a network**

Network delay: Network delay is **a design and performance characteristic of a telecommunications network**. It specifies the latency for a bit of data to travel across the network from one communication endpoint to another

Creating an network topology:



The image displays the MiniEdit application window, which is used for creating and editing network topologies. The interface includes a menu bar (File, Edit, Run, Help) and a toolbar on the left with various icons for adding network components.

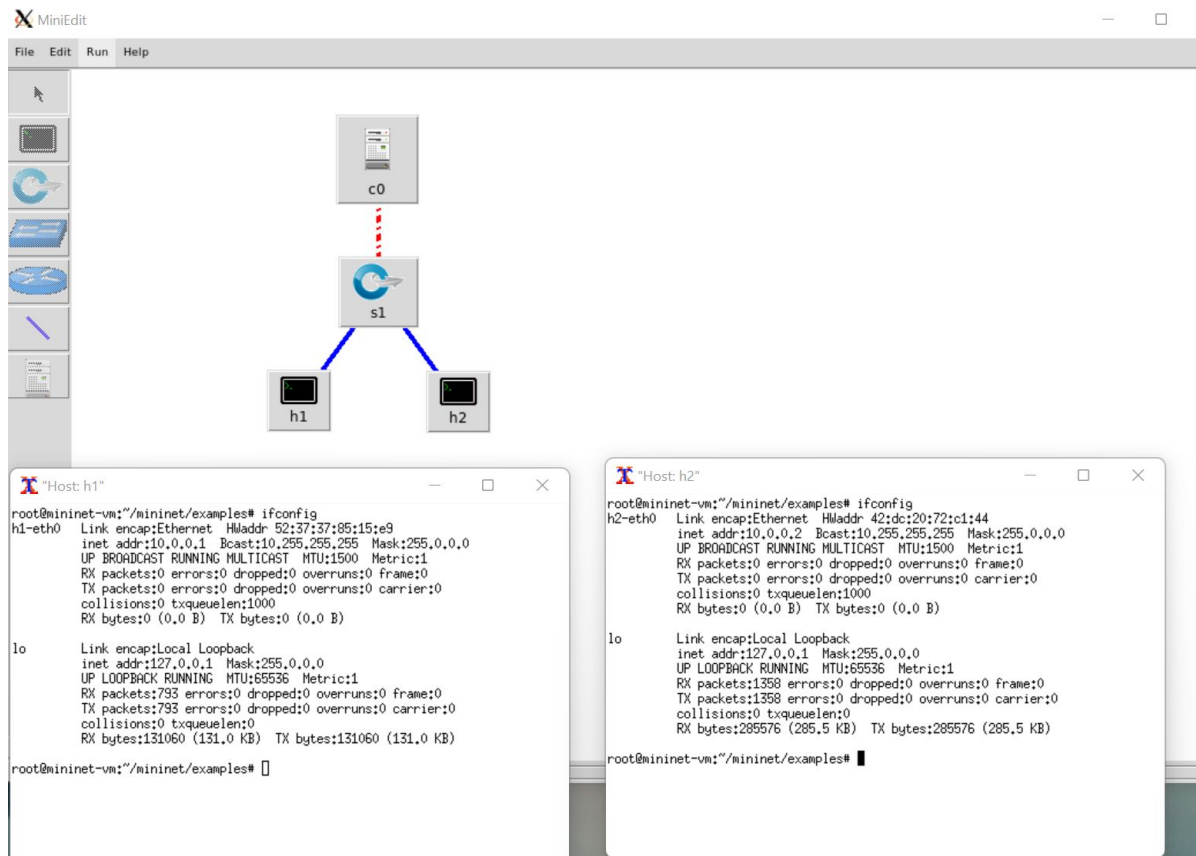
The main workspace shows a network topology diagram with the following components and connections:

- A controller node labeled **c0** (represented by a server icon) is connected to a switch node labeled **s1** (represented by a switch icon) via a red dashed line.
- The switch node **s1** is connected to two host nodes labeled **h1** and **h2** (represented by laptop icons) via solid blue lines.

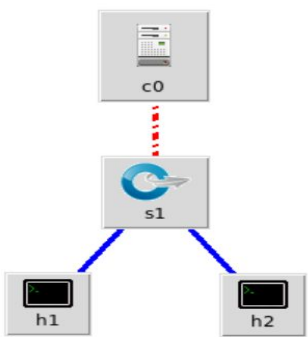
On the right side of the MiniEdit window, there is a terminal window showing the execution of the `miniedit.py` script. The terminal output is as follows:

```
mininet@mininet-vm: ~/mininet/examples
* Documentation: https://help.ubuntu.com/
Last login: Wed Mar 23 14:31:56 2022
/usr/bin/xauth: file /home/mininet/.Xauthority does not exist
mininet@mininet-vm:~$ ls
install-mininet-vm.sh  loxigen  mininet  oflops  oftest  openflow  pox
mininet@mininet-vm:~$ cd mininet/examples
mininet@mininet-vm:~/mininet/examples$ sudo python miniedit.py
MiniEdit running against Mininet 2.2.1
topo=None
Build network based on our topology.
Getting Hosts and Switches.
<class 'mininet.node.Host'>
Getting controller selection:ref
<class 'mininet.node.Host'>
Getting Links.
*** Configuring hosts
h1 h2
**** Starting 1 controllers
c0
**** Starting 1 switches
s1
No NetFlow targets specified.
No sFlow targets specified.
█
```

Network configurations of hosts h1 and h2:



Pinging h2 with h1 with no time delay or packet loss:



```
graph TD
    c0[c0] -.- s1[s1]
    s1 --- h1[h1]
    s1 --- h2[h2]
```

MiniEdit

File Edit Run Help

"Host: h1"

```
root@mininet-vm:~/mininet/examples# ifconfig
h1-eth0  Link encap:Ethernet  HWaddr 52:37:37:85:15:e9
         inet addr:10.0.0.1  Bcast:10.255.255.255  Mask:255.0.0.0
         UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
         RX packets:0 errors:0 dropped:0 overruns:0 frame:0
         TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
         collisions:0 txqueuelen:1000
         RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

lo       Link encap:Local Loopback
         inet addr:127.0.0.1  Mask:255.0.0.0
         UP LOOPBACK RUNNING  MTU:65536  Metric:1
         RX packets:793 errors:0 dropped:0 overruns:0 frame:0
         TX packets:793 errors:0 dropped:0 overruns:0 carrier:0
         collisions:0 txqueuelen:0
         RX bytes:131060 (131.0 KB)  TX bytes:131060 (131.0 KB)

root@mininet-vm:~/mininet/examples#
```

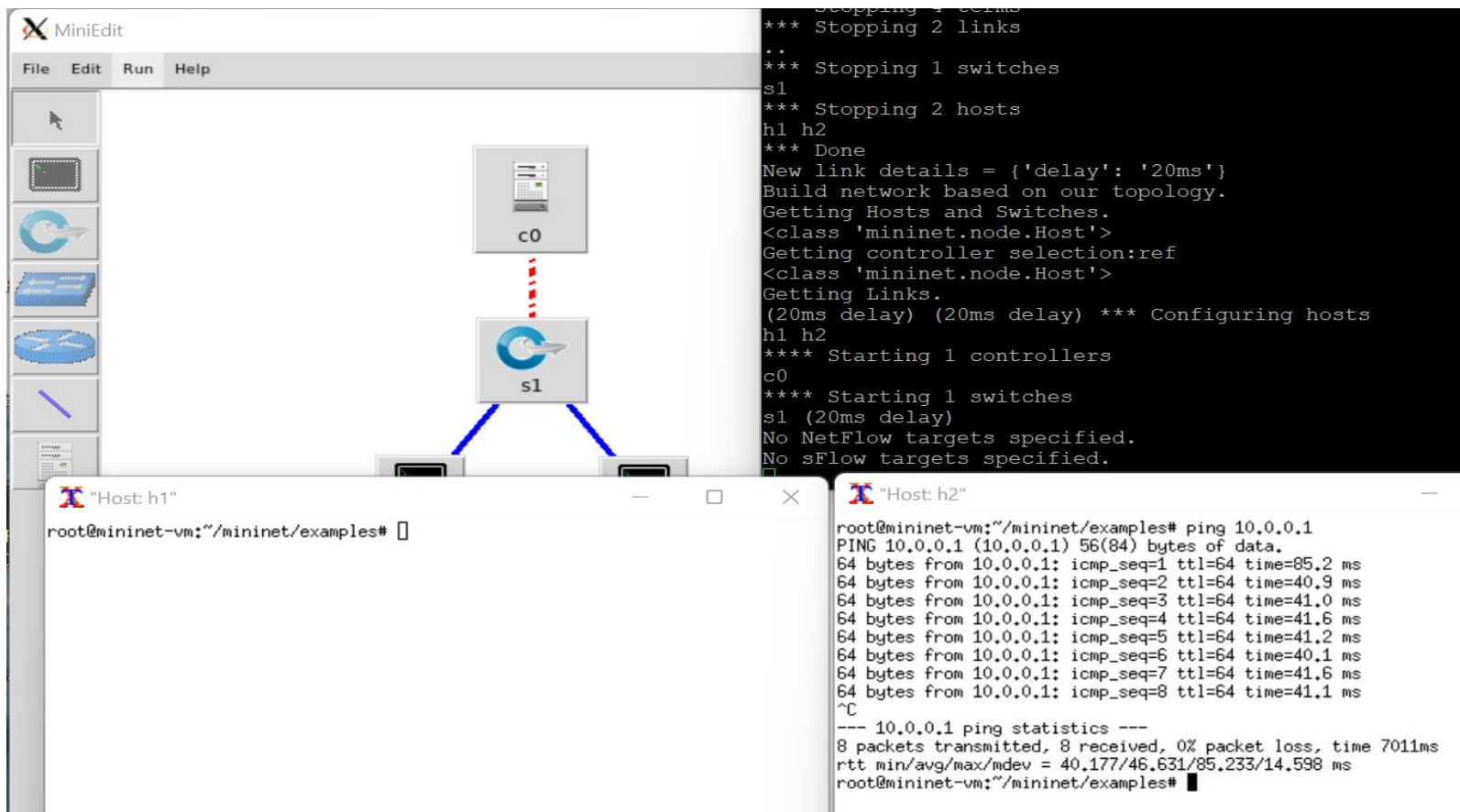
"Host: h2"

```
RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

lo       Link encap:Local Loopback
         inet addr:127.0.0.1  Mask:255.0.0.0
         UP LOOPBACK RUNNING  MTU:65536  Metric:1
         RX packets:1358 errors:0 dropped:0 overruns:0 frame:0
         TX packets:1358 errors:0 dropped:0 overruns:0 carrier:0
         collisions:0 txqueuelen:0
         RX bytes:285576 (285.5 KB)  TX bytes:285576 (285.5 KB)

root@mininet-vm:~/mininet/examples# 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=2.69 ms
64 bytes from 10.0.0.1: icmp_seq=2 ttl=64 time=0.400 ms
64 bytes from 10.0.0.1: icmp_seq=3 ttl=64 time=0.049 ms
64 bytes from 10.0.0.1: icmp_seq=4 ttl=64 time=0.049 ms
64 bytes from 10.0.0.1: icmp_seq=5 ttl=64 time=0.049 ms
64 bytes from 10.0.0.1: icmp_seq=6 ttl=64 time=0.049 ms
64 bytes from 10.0.0.1: icmp_seq=7 ttl=64 time=0.452 ms
^C
--- 10.0.0.1 ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6001ms
rtt min/avg/max/mdev = 0.048/0.534/2.692/0.896 ms
root@mininet-vm:~/mininet/examples#
```

Pinging h2 with h1 with h1 time delay of 20ms :



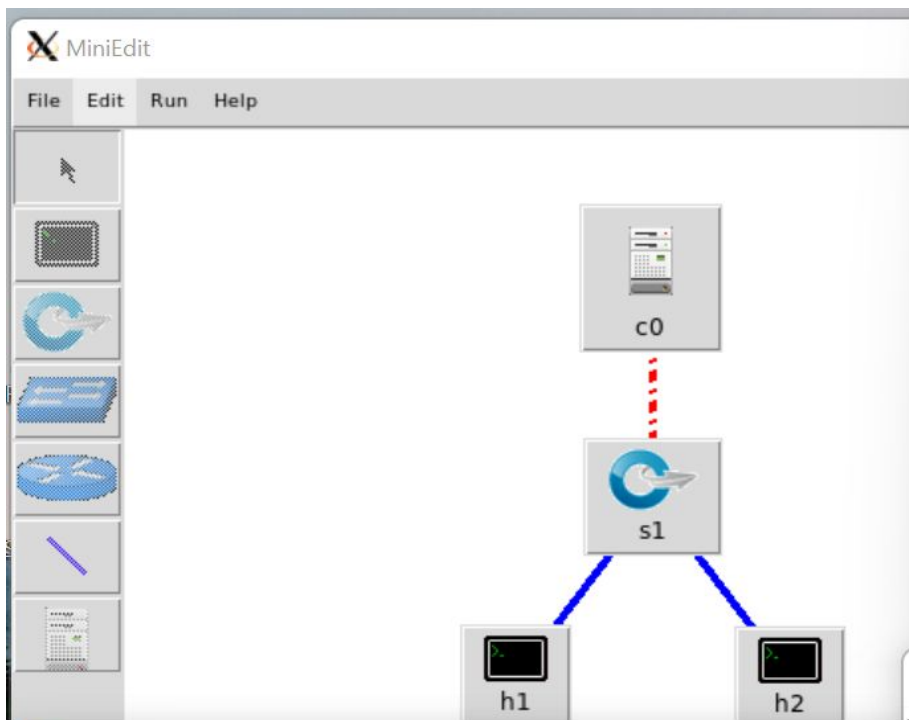
The image displays a Mininet network setup and its corresponding terminal outputs. The network diagram, shown in the MiniEdit window, features a central switch labeled 's1' connected to two hosts, 'h1' and 'h2', via blue links. A controller, 'c0', is connected to the switch 's1' via a red dashed link. The terminal window on the right shows the Mininet command-line interface with the following output:

```
*** Stopping 1 links
*** Stopping 2 links
..
*** Stopping 1 switches
s1
*** Stopping 2 hosts
h1 h2
*** Done
New link details = {'delay': '20ms'}
Build network based on our topology.
Getting Hosts and Switches.
<class 'mininet.node.Host'>
Getting controller selection:ref
<class 'mininet.node.Host'>
Getting Links.
(20ms delay) (20ms delay) *** Configuring hosts
h1 h2
**** Starting 1 controllers
c0
**** Starting 1 switches
s1 (20ms delay)
No NetFlow targets specified.
No sFlow targets specified.
```

Below the diagram, two terminal windows are open. The left window, titled "Host: h1", shows the prompt `root@mininet-vm:~/mininet/examples#`. The right window, titled "Host: h2", shows the prompt `root@mininet-vm:~/mininet/examples# ping 10.0.0.1` and the resulting ping statistics:

```
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=85.2 ms
64 bytes from 10.0.0.1: icmp_seq=2 ttl=64 time=40.9 ms
64 bytes from 10.0.0.1: icmp_seq=3 ttl=64 time=41.0 ms
64 bytes from 10.0.0.1: icmp_seq=4 ttl=64 time=41.6 ms
64 bytes from 10.0.0.1: icmp_seq=5 ttl=64 time=41.2 ms
64 bytes from 10.0.0.1: icmp_seq=6 ttl=64 time=40.1 ms
64 bytes from 10.0.0.1: icmp_seq=7 ttl=64 time=41.6 ms
64 bytes from 10.0.0.1: icmp_seq=8 ttl=64 time=41.1 ms
^C
--- 10.0.0.1 ping statistics ---
8 packets transmitted, 8 received, 0% packet loss, time 7011ms
rtt min/avg/max/mdev = 40.177/46.631/85.233/14.598 ms
root@mininet-vm:~/mininet/examples#
```

Pinging h2 with h1 with packet loss of 50% :



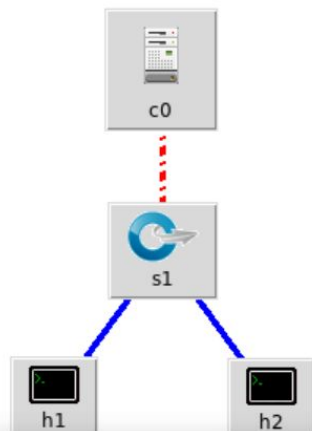
```
*** Stopping 4 terms
*** Stopping 2 links
..
*** Stopping 1 switches
s1
*** Stopping 2 hosts
h1 h2
*** Done
New link details = {'loss': 50}
Build network based on our topology.
Getting Hosts and Switches.
<class 'mininet.node.Host'>
Getting controller selection:ref
<class 'mininet.node.Host'>
Getting Links.
(50% loss) (50% loss) *** Configuring hosts
h1 h2
**** Starting 1 controllers
c0
**** Starting 1 switches
s1 (50% loss)
No NetFlow targets specified.
No sFlow targets specified.
```

Host: h1"

```
root@mininet-vm:~/mininet/examples#
```

Host: h2"

```
root@mininet-vm:~/mininet/examples# ping 10.0.0.1
PING 10.0.0.1 (10.0.0.1) 56(84) bytes of data.
From 10.0.0.2 icmp_seq=1 Destination Host Unreachable
From 10.0.0.2 icmp_seq=2 Destination Host Unreachable
From 10.0.0.2 icmp_seq=3 Destination Host Unreachable
64 bytes from 10.0.0.2: icmp_seq=9 ttl=64 time=0.938 ms
64 bytes from 10.0.0.2: icmp_seq=14 ttl=64 time=0.181 ms
64 bytes from 10.0.0.2: icmp_seq=26 ttl=64 time=0.180 ms
64 bytes from 10.0.0.2: icmp_seq=34 ttl=64 time=0.238 ms
```



"Host: h2"

```
root@mininet-vm:~/mininet/examples# 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=1.37 ms
64 bytes from 10.0.0.1: icmp_seq=2 ttl=64 time=0.465 ms
64 bytes from 10.0.0.1: icmp_seq=3 ttl=64 time=0.048 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=0.055 ms
64 bytes from 10.0.0.1: icmp_seq=6 ttl=64 time=0.074 ms
64 bytes from 10.0.0.1: icmp_seq=7 ttl=64 time=0.248 ms
64 bytes from 10.0.0.1: icmp_seq=8 ttl=64 time=0.086 ms
^C
```

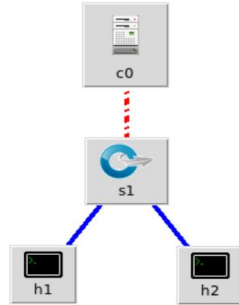
"Host: h1"

```
listening on h1-eth0, link-type EN10MB (Ethernet), capture size 65535 bytes
15:17:06.671934 IP 10.0.0.2 > 10.0.0.1: ICMP echo request, id 3040, seq 4, length 64
15:17:06.671953 IP 10.0.0.1 > 10.0.0.2: ICMP echo reply, id 3040, seq 4, length 64
15:17:07.670937 IP 10.0.0.2 > 10.0.0.1: ICMP echo request, id 3040, seq 5, length 64
15:17:07.670957 IP 10.0.0.1 > 10.0.0.2: ICMP echo reply, id 3040, seq 5, length 64
15:17:08.671201 IP 10.0.0.2 > 10.0.0.1: ICMP echo request, id 3040, seq 6, length 64
15:17:08.671229 IP 10.0.0.1 > 10.0.0.2: ICMP echo reply, id 3040, seq 6, length 64
15:17:08.679076 ARP, Request who-has 10.0.0.2 tell 10.0.0.1, length 28
15:17:08.681284 ARP, Reply 10.0.0.2 is-at 0e:3f:1a:d6:81:c1, length 28
15:17:09.671009 IP 10.0.0.2 > 10.0.0.1: ICMP echo request, id 3040, seq 7, length 64
15:17:09.671023 IP 10.0.0.1 > 10.0.0.2: ICMP echo reply, id 3040, seq 7, length 64
15:17:10.671034 IP 10.0.0.2 > 10.0.0.1: ICMP echo request, id 3040, seq 8, length 64
15:17:10.671059 IP 10.0.0.1 > 10.0.0.2: ICMP echo reply, id 3040, seq 8, length 64
```


H2 is the server and H1 is my client, H2(server takes input from the client and converts it into lower case and sends it back to client(H1):



File Edit Run Help



```
Host: h1
root@mininet-vml:/mininet/examples# ls
baresshd.py      controlnet.py      linuxrouter.py      popen.py
bind.py          cpup.py           miniedit.py         README.md
clustercli.py    emptynet.py        mobility.py          scratchnet.py
clustercli.pyc   h1                multilink.py        scratchnetuser.py
clusterdemo.py   h2                multiping.py         simpleperf.py
cluster.py        hwintf.py          multipoll.py         sshd.py
cluster.pyc       __init__.py        multitest.py         test
clusterSanity.py __init__.pyc        natnet.py            tree1024.py
consoles.py       intfOptions.py     nat.py              treeping64.py
controllers2.py   limit.py           numberedports.py    vlanhost.py
controllers.py    linearbandwidth.py popenpoll.py
root@mininet-vml:/mininet/examples# cd h1
root@mininet-vml:/mininet/examples/h1# ls
practicetcp1.py udpclientpractice.py
root@mininet-vml:/mininet/examples/h1# python3 practicetcp1.py
Enter a Upper case string to convert it to lower case : NITESH
nitesh
root@mininet-vml:/mininet/examples/h1#
```

```
Host: h2
root@mininet-vml:/mininet/examples# ls
baresshd.py      controlnet.py      linuxrouter.py      popen.py
bind.py          cpup.py           miniedit.py         README.md
clustercli.py    emptynet.py        mobility.py          scratchnet.py
clustercli.pyc   h1                multilink.py        scratchnetuser.py
clusterdemo.py   h2                multiping.py         simpleperf.py
cluster.py        hwintf.py          multipoll.py         sshd.py
cluster.pyc       __init__.py        multitest.py         test
clusterSanity.py __init__.pyc        natnet.py            tree1024.py
consoles.py       intfOptions.py     nat.py              treeping64.py
controllers2.py   limit.py           numberedports.py    vlanhost.py
controllers.py    linearbandwidth.py popenpoll.py
root@mininet-vml:/mininet/examples# cd h2
root@mininet-vml:/mininet/examples/h2# ls
practicetcp1.py practiceudpsrver.py
root@mininet-vml:/mininet/examples/h2# python3 practicetcp1.py
('10.0.0.1', 40451)
```

Result:

- Created an network topology
- Performed time delay and packet loss between h1 and h2
- Tcp protocol is implemented between hosts h1 and h2