

Computer Comm & Networks - ITCS 8166

(Assignment – 2)

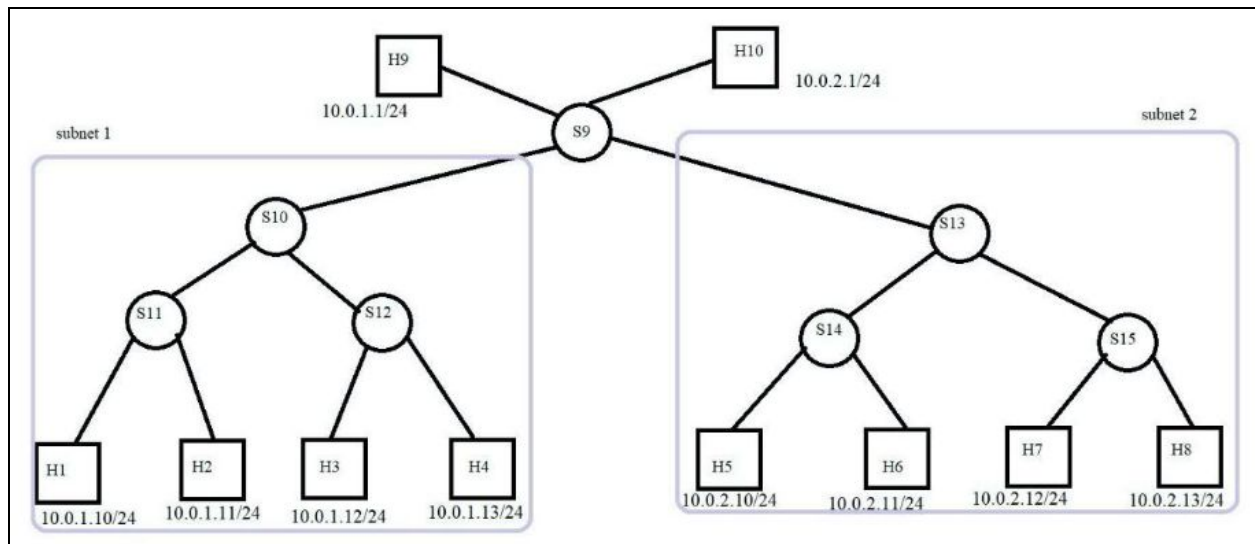
Abdullah Al Raqibul Islam (UNCC ID# 801151189)

Software Details

1. Host Operating System and version: macOS (Version: 10.15.7)
2. Virtualization tool name and version: VirtualBox (Version: 6.1.16)

Answers

1. Create mininet topology script for the network topology given below (5 points)



Answer:

Code attached code/topology.py

```
#!/usr/bin/python

from mininet.topo import Topo
from mininet.net import Mininet
from mininet.util import dumpNodeConnections
from mininet.log import setLogLevel
```

```

class MyTopo(Topo):
    "Simple topology example."

    def __init__(self):
        "Create custom topo."

        # Initialize topology
        Topo.__init__(self)

        # Add hosts and switches
        h1 = [self.addHost('h1', ip='10.0.1.10/24')]
        h2 = [self.addHost('h2', ip='10.0.1.11/24')]
        h3 = [self.addHost('h3', ip='10.0.1.12/24')]
        h4 = [self.addHost('h4', ip='10.0.1.13/24')]
        h5 = [self.addHost('h5', ip='10.0.2.10/24')]
        h6 = [self.addHost('h6', ip='10.0.2.11/24')]
        h7 = [self.addHost('h7', ip='10.0.2.12/24')]
        h8 = [self.addHost('h8', ip='10.0.2.13/24')]
        h9 = [self.addHost('h9', ip='10.0.1.1/24')]
        h10 = [self.addHost('h10', ip='10.0.2.1/24')]

        s9 = [self.addSwitch('s9')]
        s10 = [self.addSwitch('s10')]
        s11 = [self.addSwitch('s11')]
        s12 = [self.addSwitch('s12')]
        s13 = [self.addSwitch('s13')]
        s14 = [self.addSwitch('s14')]
        s15 = [self.addSwitch('s15')]

        # host to switch links
        self.addLink('s11', 'h1')
        self.addLink('s11', 'h2')
        self.addLink('s12', 'h3')
        self.addLink('s12', 'h4')
        self.addLink('s14', 'h5')
        self.addLink('s14', 'h6')
        self.addLink('s15', 'h7')
        self.addLink('s15', 'h8')
        self.addLink('s9', 'h9')
        self.addLink('s9', 'h10')

        # switch to switch links
        self.addLink('s10', 's11')
        self.addLink('s10', 's12')
        self.addLink('s13', 's14')
        self.addLink('s13', 's15')
        self.addLink('s9', 's10')
        self.addLink('s9', 's13')

def basicTopoTest():
    "Create and test a custom network"
    topo = MyTopo()
    net = Mininet(topo)

```

```

net.start()
print("Dumping host connections")
dumpNodeConnections(net.hosts)
net.stop()

if __name__ == '__main__':
    # Tell mininet to print useful information
    setLogLevel('info')
    basicTopoTest()

```

2. Run the topology script in the provided VM as shown below (5 points)

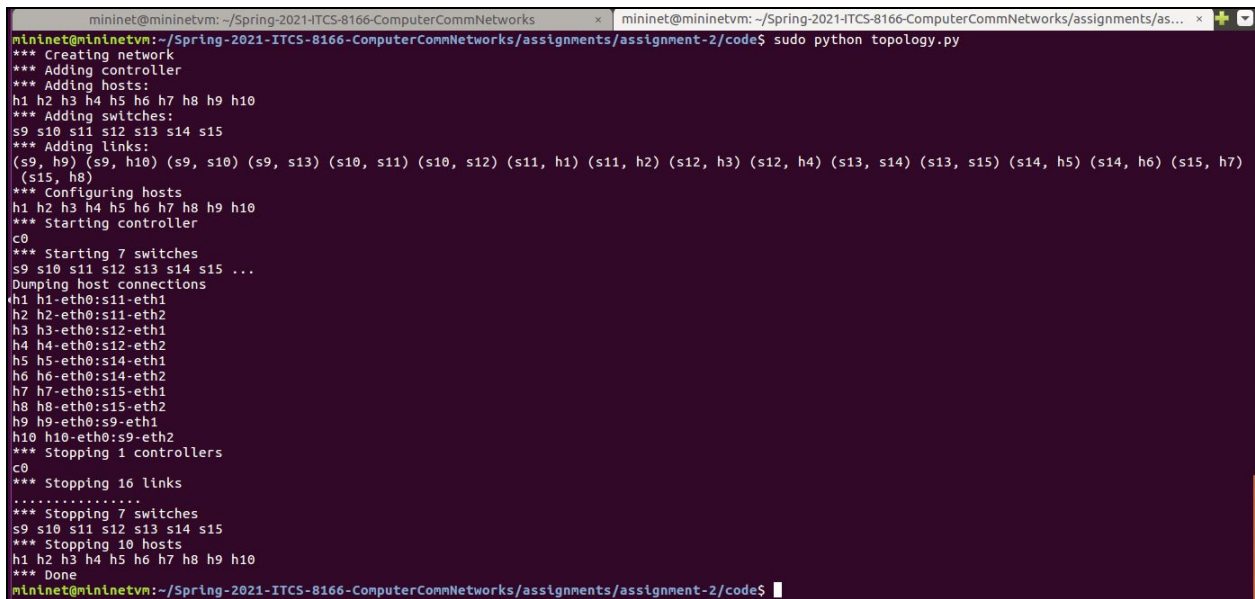
Answer:

Run the code (attached in Q1) using the following command:

```
> sudo python topology.py
```

Program code: Code attached code/topology.py

Verification and required information/output: Screenshot attached bellow



```

mininet@mininetvm: ~/Spring-2021-ITCS-8166-ComputerCommNetworks
mininet@mininetvm: ~/Spring-2021-ITCS-8166-ComputerCommNetworks/assignments/as...
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3 h4 h5 h6 h7 h8 h9 h10
*** Adding switches:
s9 s10 s11 s12 s13 s14 s15
*** Adding links:
(s9, h9) (s9, h10) (s9, s10) (s9, s13) (s10, s11) (s10, s12) (s11, h1) (s11, h2) (s12, h3) (s12, h4) (s13, s14) (s13, s15) (s14, h5) (s14, h6) (s15, h7)
(s15, h8)
*** Configuring hosts
h1 h2 h3 h4 h5 h6 h7 h8 h9 h10
*** Starting controller
c0
*** Starting 7 switches
s9 s10 s11 s12 s13 s14 s15 ...
Dumping host connections
h1 h1-eth0:s11-eth1
h2 h2-eth0:s11-eth2
h3 h3-eth0:s12-eth1
h4 h4-eth0:s12-eth2
h5 h5-eth0:s14-eth1
h6 h6-eth0:s14-eth2
h7 h7-eth0:s15-eth1
h8 h8-eth0:s15-eth2
h9 h9-eth0:s9-eth1
h10 h10-eth0:s9-eth2
*** Stopping 1 controllers
c0
*** Stopping 16 links
.....
*** Stopping 7 switches
s9 s10 s11 s12 s13 s14 s15
*** Stopping 10 hosts
h1 h2 h3 h4 h5 h6 h7 h8 h9 h10
*** Done
mininet@mininetvm: ~/Spring-2021-ITCS-8166-ComputerCommNetworks/assignments/assignment-2/code$

```

3. Use the pingall command in the Mininet terminal and check the connectivity (5 points). If you see a packet drop, please identify the reason for packet drop.

Answer:

Code: Code attached code/topology.py

```

#!/usr/bin/python

from mininet.topo import Topo
from mininet.net import Mininet
from mininet.util import dumpNodeConnections
from mininet.log import setLogLevel

class MyTopo(Topo):
    "Simple topology example."

    def __init__(self):
        "Create custom topo."

        # Initialize topology
        Topo.__init__(self)

        # Add hosts and switches
        h1 = [self.addHost('h1', ip='10.0.1.10/24')]
        h2 = [self.addHost('h2', ip='10.0.1.11/24')]
        h3 = [self.addHost('h3', ip='10.0.1.12/24')]
        h4 = [self.addHost('h4', ip='10.0.1.13/24')]
        h5 = [self.addHost('h5', ip='10.0.2.10/24')]
        h6 = [self.addHost('h6', ip='10.0.2.11/24')]
        h7 = [self.addHost('h7', ip='10.0.2.12/24')]
        h8 = [self.addHost('h8', ip='10.0.2.13/24')]
        h9 = [self.addHost('h9', ip='10.0.1.1/24')]
        h10 = [self.addHost('h10', ip='10.0.2.1/24')]

        s9 = [self.addSwitch('s9')]
        s10 = [self.addSwitch('s10')]
        s11 = [self.addSwitch('s11')]
        s12 = [self.addSwitch('s12')]
        s13 = [self.addSwitch('s13')]
        s14 = [self.addSwitch('s14')]
        s15 = [self.addSwitch('s15')]

        # host to switch links
        self.addLink('s11', 'h1')
        self.addLink('s11', 'h2')
        self.addLink('s12', 'h3')
        self.addLink('s12', 'h4')
        self.addLink('s14', 'h5')
        self.addLink('s14', 'h6')
        self.addLink('s15', 'h7')
        self.addLink('s15', 'h8')
        self.addLink('s9', 'h9')
        self.addLink('s9', 'h10')

        # switch to switch links
        self.addLink('s10', 's11')
        self.addLink('s10', 's12')
        self.addLink('s13', 's14')
        self.addLink('s13', 's15')

```

```

        self.addLink('s9', 's10')
        self.addLink('s9', 's13')

def pingAllTest():
    "Create and test connectivity of a custom network"
    topo = MyTopo()
    net = Mininet(topo)
    net.start()
    print("Testing network connectivity")
    net.pingAll()
    net.stop()

if __name__ == '__main__':
    # Tell mininet to print useful information
    setLogLevel('info')
    pingAllTest()

```

Output: Screenshot attached below

```

mininet@mininetvm: ~/Spring-2021-ITCS-8166-ComputerCommNetworks
mininet@mininetvm: ~/Spring-2021-ITCS-8166-ComputerCommNetworks/assignments/as...
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3 h4 h5 h6 h7 h8 h9 h10
*** Adding switches:
s9 s10 s11 s12 s13 s14 s15
*** Adding links:
(s9, h9) (s9, h10) (s9, s10) (s9, s13) (s10, s11) (s10, s12) (s11, h1) (s11, h2) (s12, h3) (s12, h4) (s13, s14) (s13, s15) (s14, h5) (s14, h6) (s15, h7)
(s15, h8)
*** Configuring hosts
h1 h2 h3 h4 h5 h6 h7 h8 h9 h10
*** Starting controller
c0
*** Starting 7 switches
s9 s10 s11 s12 s13 s14 s15 ...
Testing network connectivity
*** Ping: testing ping reachability
h1 -> h2 h3 h4 X X X X h9 X
h2 -> h1 h3 h4 X X X X h9 X
h3 -> h1 h2 h4 X X X X h9 X
h4 -> h1 h2 h3 X X X X h9 X
h5 -> X X X X h6 h7 h8 X h10
h6 -> X X X X h5 h7 h8 X h10
h7 -> X X X X h5 h6 h8 X h10
h8 -> X X X X h5 h6 h7 X h10
h9 -> h1 h2 h3 h4 X X X X
h10 -> X X X X h5 h6 h7 h8 X
*** Results: 55% dropped (40/90 received)
*** Stopping 1 controllers
c0
*** Stopping 16 links
.....
*** Stopping 7 switches
s9 s10 s11 s12 s13 s14 s15
*** Stopping 10 hosts
h1 h2 h3 h4 h5 h6 h7 h8 h9 h10
*** Done
mininet@mininetvm: ~/Spring-2021-ITCS-8166-ComputerCommNetworks/assignments/assignment-2/code$

```

Note on packet drop: From the program output it is clearly visible that 55% packet dropped. This is because we have two subnets in our network topology. Subnet mask is 24 (i.e., 255.255.255.0), hence ips “10.0.1.*” and “10.0.2.*” are in two different subnets. I further test this by changing the subnet mask setting as 16 (meaning 255.255.0.0, and hence all the hosts belong to the same subnet), where I get 0% packet drop. The test run output attached below:

```

mininet@mininetvm:~/Spring-2021-ITCS-8166-ComputerCommNetworks/assignments/assignment-2/code$ sudo python topology.py
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3 h4 h5 h6 h7 h8 h9 h10
*** Adding switches:
s9 s10 s11 s12 s13 s14 s15
*** Adding links:
(s9, h9) (s9, h10) (s9, s10) (s9, s13) (s10, s11) (s10, s12) (s11, h1) (s11, h2) (s12, h3) (s12, h4) (s13, s14) (s13, s15) (s14, h5) (s14, h6) (s15, h7)
(s15, h8)
*** Configuring hosts
h1 h2 h3 h4 h5 h6 h7 h8 h9 h10
*** Starting controller
c0
*** Starting 7 switches
s9 s10 s11 s12 s13 s14 s15 ...
Testing network connectivity
*** Ping: testing ping reachability
h1 -> h2 h3 h4 h5 h6 h7 h8 h9 h10
h2 -> h1 h3 h4 h5 h6 h7 h8 h9 h10
h3 -> h1 h2 h4 h5 h6 h7 h8 h9 h10
h4 -> h1 h2 h3 h5 h6 h7 h8 h9 h10
h5 -> h1 h2 h3 h4 h6 h7 h8 h9 h10
h6 -> h1 h2 h3 h4 h5 h7 h8 h9 h10
h7 -> h1 h2 h3 h4 h5 h6 h8 h9 h10
h8 -> h1 h2 h3 h4 h5 h6 h7 h9 h10
h9 -> h1 h2 h3 h4 h5 h6 h7 h8 h10
h10 -> h1 h2 h3 h4 h5 h6 h7 h8 h9
*** Results: 0% dropped (90/90 received)
*** Stopping 1 controllers
c0
*** Stopping 16 links
.....
*** Stopping 7 switches
s9 s10 s11 s12 s13 s14 s15
*** Stopping 10 hosts
h1 h2 h3 h4 h5 h6 h7 h8 h9 h10
*** Done
mininet@mininetvm:~/Spring-2021-ITCS-8166-ComputerCommNetworks/assignments/assignment-2/code$

```

4. You must think about interconnection between each node and create a tabular format (30 points) below to analyze the network topology

- Scenario: 1
 - Set link bandwidth between S9<-> S10 as bw=10Mbps
 - Set link bandwidth between S10<-> S12 as bw=15Mbps, delay = 10ms
- Scenario: 2
 - Set link bandwidth between S9<-> S13 as bw=20Mbps
 - Set link bandwidth between S13<-> S14 as bw=20Mbps, delay = 5ms

Answer:

Host	IP address	Is this host pingable from h9?	Is this host pingable from h10?	Scenario: 1 Measure bw and delay using iperf on H9	Scenario: 2 Measure bw and delay using iperf on H10
H1	10.0.1.10/24	Yes	No	9.24 Mbits/sec	x
H2	10.0.1.11/24	Yes	No	7.51 Mbits/sec	x
H3	10.0.1.12/24	Yes	No	9.17 Mbits/sec	x
H4	10.0.1.13/24	Yes	No	9.37 Mbits/sec	x
H5	10.0.2.10/24	No	Yes	x	18.2 Mbits/sec
H6	10.0.2.11/24	No	Yes	x	18.3 Mbits/sec

H7	10.0.2.12/24	No	Yes	x	18.6 Mbits/sec
H8	10.0.2.13/24	No	Yes	x	18.3 Mbits/sec
H9	10.0.1.1/24	Yes	No	x	x
H10	10.0.2.1/24	No	Yes	x	x

Scenario-1 test output:

```

mininet@mininetvm: ~/Spring-2021-ITCS-8166-ComputerCommNetworks/assignments/assignment-2/code
*** Configuring hosts
h1 h2 h3 h4 h5 h6 h7 h8 h9 h10
*** Starting controller
c0
*** Starting 7 switches
s9 s10 s11 s12 s13 s14 s15 ...(10.00Mbit) (15.00Mbit 10ms delay) (10.00Mbit) (15.00Mbit 10ms delay)
Dumping host connections
h1 h1-eth0:s11-eth1
h2 h2-eth0:s11-eth2
h3 h3-eth0:s12-eth1
h4 h4-eth0:s12-eth2
h5 h5-eth0:s14-eth1
h6 h6-eth0:s14-eth2
h7 h7-eth0:s15-eth1
h8 h8-eth0:s15-eth2
h9 h9-eth0:s9-eth1
h10 h10-eth0:s9-eth2
Testing network connectivity
*** Ping: testing ping reachability
h1 -> h2 h3 h4 X X X X h9 X
h2 -> h1 h3 h4 X X X X h9 X
h3 -> h1 h2 h4 X X X X h9 X
h4 -> h1 h2 h3 X X X X h9 X
h5 -> X X X X h6 h7 h8 X h10
h6 -> X X X X h5 h7 h8 X h10
h7 -> X X X X h5 h6 h8 X h10
h8 -> X X X X h5 h6 h7 X h10
h9 -> h1 h2 h3 h4 X X X X
h10 -> X X X X h5 h6 h7 h8 X
*** Results: 55% dropped (40/90 received)
Testing bandwidth between h1 and h9
*** Iperf: testing TCP bandwidth between h1 and h9
*** Results: ['9.24 Mbits/sec', '11.3 Mbits/sec']
Testing bandwidth between h2 and h9
*** Iperf: testing TCP bandwidth between h2 and h9
*** Results: ['7.51 Mbits/sec', '9.81 Mbits/sec']
Testing bandwidth between h3 and h9
*** Iperf: testing TCP bandwidth between h3 and h9
*** Results: ['9.17 Mbits/sec', '11.6 Mbits/sec']
Testing bandwidth between h4 and h9
*** Iperf: testing TCP bandwidth between h4 and h9
*** Results: ['9.37 Mbits/sec', '11.8 Mbits/sec']
*** Stopping 1 controllers
c0
*** Stopping 16 links
.....
*** Stopping 7 switches
s9 s10 s11 s12 s13 s14 s15
*** Stopping 10 hosts
h1 h2 h3 h4 h5 h6 h7 h8 h9 h10
*** Done

```

Scenario-2 test output:


```
mininet@mininetvm: ~/Spring-2021-ITCS-8166-ComputerCommNetworks/assignments/assignment-2/code x mini
*** Configuring hosts
h1 h2 h3 h4 h5 h6 h7 h8 h9 h10
*** Starting controller
c0
*** Starting 7 switches
s9 s10 s11 s12 s13 s14 s15 ...(20.00Mbit) (20.00Mbit 5ms delay) (20.00Mbit) (20.00Mbit 5ms delay)
Dumping host connections
h1 h1-eth0:s11-eth1
h2 h2-eth0:s11-eth2
h3 h3-eth0:s12-eth1
h4 h4-eth0:s12-eth2
h5 h5-eth0:s14-eth1
h6 h6-eth0:s14-eth2
h7 h7-eth0:s15-eth1
h8 h8-eth0:s15-eth2
h9 h9-eth0:s9-eth1
h10 h10-eth0:s9-eth2
Testing network connectivity
*** Ping: testing ping reachability
h1 -> h2 h3 h4 X X X h9 X
h2 -> h1 h3 h4 X X X h9 X
h3 -> h1 h2 h4 X X X h9 X
h4 -> h1 h2 h3 X X X h9 X
h5 -> X X X X h6 h7 h8 X h10
h6 -> X X X X h5 h7 h8 X h10
h7 -> X X X X h5 h6 h8 X h10
h8 -> X X X X h5 h6 h7 X h10
h9 -> h1 h2 h3 h4 X X X X
h10 -> X X X X h5 h6 h7 h8 X
*** Results: 55% dropped (40/90 received)
Testing bandwidth between h5 and h10
*** Iperf: testing TCP bandwidth between h5 and h10
*** Results: ['18.2 Mbits/sec', '21.7 Mbits/sec']
Testing bandwidth between h6 and h10
*** Iperf: testing TCP bandwidth between h6 and h10
*** Results: ['18.3 Mbits/sec', '22.0 Mbits/sec']
Testing bandwidth between h7 and h10
*** Iperf: testing TCP bandwidth between h7 and h10
*** Results: ['18.6 Mbits/sec', '22.1 Mbits/sec']
Testing bandwidth between h8 and h10
*** Iperf: testing TCP bandwidth between h8 and h10
*** Results: ['18.3 Mbits/sec', '22.1 Mbits/sec']
*** Stopping 1 controllers
c0
*** Stopping 16 links
.....
*** Stopping 7 switches
s9 s10 s11 s12 s13 s14 s15
*** Stopping 10 hosts
h1 h2 h3 h4 h5 h6 h7 h8 h9 h10
*** Done
```

5. In Short, explain about why some nodes are not reachable from h9 and h10 in the given topology.

Answer:

From the table provided in Q4, we can see that h5, h6, h7, h8, h10 are not reachable from h9. On the other hand, h1, h2, h3, h4, h9 are not reachable from h10. This is because they belong to different subnets. I give an explanation on this in the answer of Q3.