report

tangpeng: 517020910038

The main purpose of the lab is to learn how to use **mininet** to simulate topology by finishing three problems.

Problem 1

Analysis

In this problem, I use the **mininet** to simulate the requested topology and test throughput between every host-pairs by using **Iperf**. By reading the codes in the examples file, I learned how to construct the topology and test the throughput. The code of this problem is showing as follows:

```
#!/usr/bin/env python
import sys

from functools import partial

from mininet.net import Mininet
from mininet.node import UserSwitch, OVSKernelSwitch, Controller, CPULimitedHost
from mininet.topo import Topo
from mininet.log import lg, info, setLogLevel
from mininet.util import irange, quietRun
from mininet.util import dumpNodeConnections
from mininet.link import TCLink


class networkTopo(Topo):
    def build(self,**params):
        # define the host
        h1  = self.addHost('h1')
        h2  = self.addHost('h2')
        h3  = self.addHost('h3')
        #define the switch
        s1 = self.addSwitch('s1')
        s2 = self.addSwitch('s2')
        s3 = self.addSwitch('s3')

        #define the link
        self.addLink(h1, s1)
        self.addLink(s1, s2, bw=10)
        self.addLink(s1, s3, bw=10)
        self.addLink(s3, h3)
        self.addLink(h2, s2)

def throughputTest():
    #create the network
    topo = networkTopo()
    net = Mininet(topo=topo,
                  host=CPULimitedHost, link=TCLink,
                  autoStaticArp=True )
```

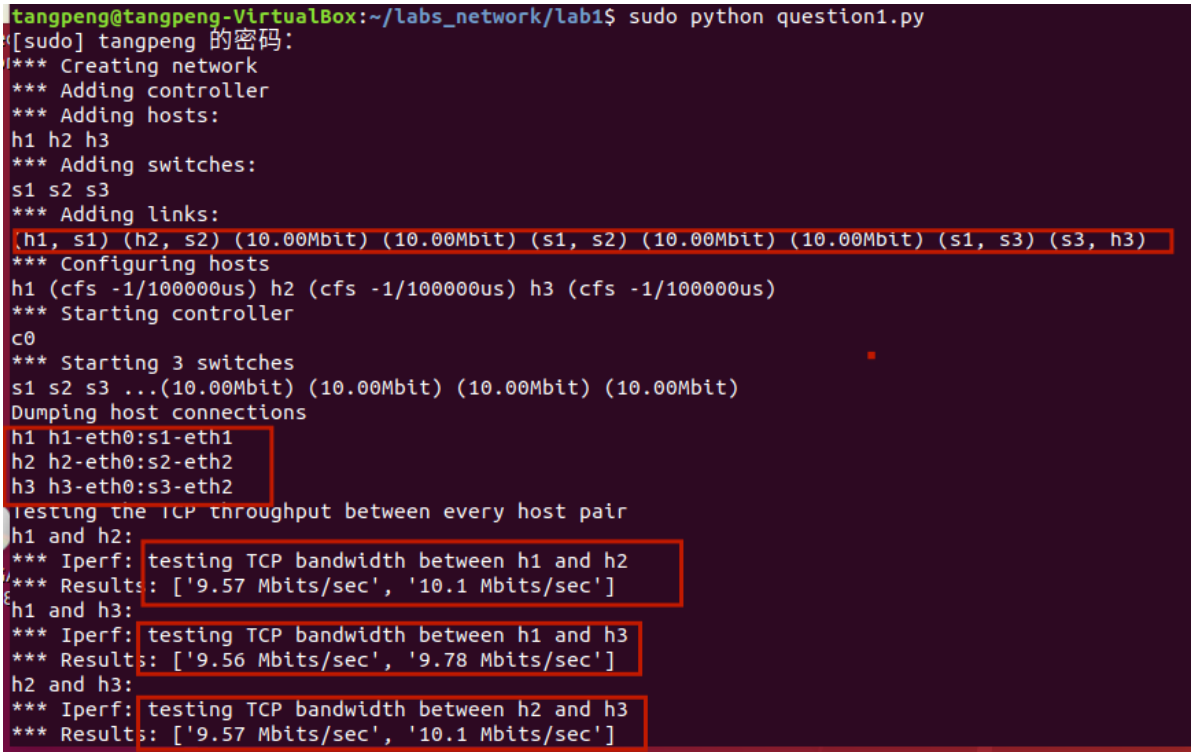
```
    net.start()
    info( "Dumping host connections\n" )
        dumpNodeConnections(net.hosts)
    info("Testing the TCP throughput between every host pair\n")
    h1, h2, h3 = net.getNodeByName('h1','h2','h3')
    info("h1 and h2:\n")
    net.iperf((h1,h2), l4Type = 'TCP')
    info("h1 and h3:\n")
    net.iperf((h1,h3), l4Type = 'TCP')
    info("h2 and h3:\n")
    net.iperf((h2,h3), l4Type = 'TCP')
    net.stop()


if __name__ == '__main__':
    setLogLevel( 'info' )
    # testing
    throughputTest()
```

In above codes, **networkTopo(Topo)** and **throughputTest()** are the main parts. **networkTopo(Topo)** is used to define a topology class and **throughputTest()** is used to test the throughput between every host pair.

- In **networkTopo(Topo)** , we can add needed hosts by using the **addHost()** function, add needed switches by using the **addSwitch()** function and add needed links by using the **addLink()** function. When adding links, we can set the bandwidth with requirement. By using the three functions, we can construct a basic network topology.
- In **throughputTest()**, we need construct a topology object by **networkTopo(Topo)** class at first. Then we use the **Mininet()** function to construct the network object. after setting up the network by calling the **start()** function, we can use the **iperf()** function to test the throughput and print the results. At the end, we close the network by calling the **stop()** function.

Results

By running the above code, we get the following result.



From the result, we can see the bandwidth is set to $10Mbit/sec$ and throughputs of three host-pairs are all about $10Mbit/sec$.

Problem 2

Analysis

In this problem, we need to set the packet loss rate of the link (s1,s2) and (s1,s3) as 5%. We can modify the parameters of **addLink()** function in the code of problem 1. Detailedly, we modify two lines codes as follows:

```
self.addLink(s1, s2, bw=10, loss=5)
self.addLink(s1, s3, bw=10, loss=5)
```

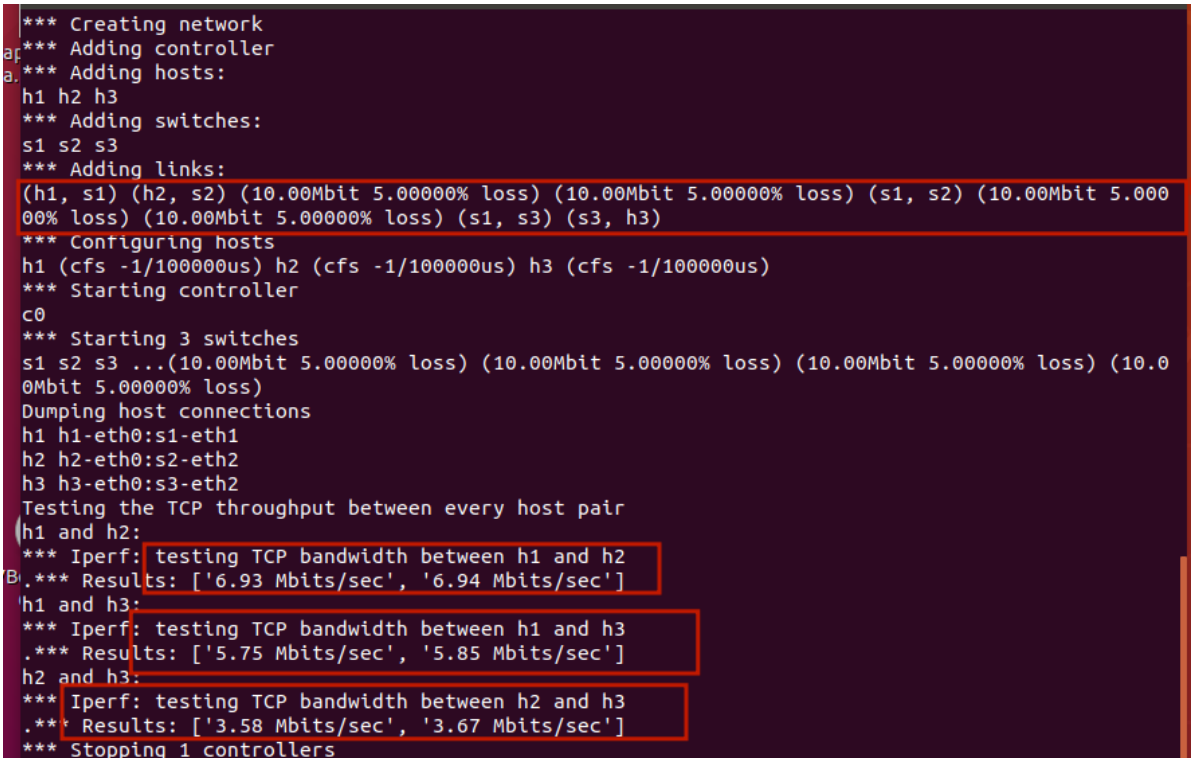
The whole codes of this problem is shown as follows:

```
#!/usr/bin/env python
import sys

from functools import partial

from mininet.net import Mininet
from mininet.node import UserSwitch, OVSKernelSwitch, Controller, CPULimitedHost
from mininet.topo import Topo
from mininet.log import lg, info, setLogLevel
from mininet.util import irange, quietRun
from mininet.util import dumpNodeConnections
from mininet.link import TCLink


class networkTopo(Topo):
    def build(self,**params):
        # define the host
        h1  = self.addHost('h1')
        h2  = self.addHost('h2')
```

```
        h3  = self.addHost('h3')
        #define the switch
        s1 = self.addSwitch('s1')
        s2 = self.addSwitch('s2')
        s3 = self.addSwitch('s3')

        #define the link
        self.addLink(h1, s1)
        self.addLink(s1, s2, bw=10, loss=5)
        self.addLink(s1, s3, bw=10, loss=5)
        self.addLink(s3, h3)
        self.addLink(h2, s2)

def throughputTest():
    #create the network
    topo = networkTopo()
    net = Mininet(topo=topo,
                  host=CPULimitedHost, link=TCLink,
                  autoStaticArp=True )

    net.start()
    info( "Dumping host connections\n" )
        dumpNodeConnections(net.hosts)
    info("Testing the TCP throughput between every host pair\n")
    h1, h2, h3 = net.getNodeByName('h1','h2','h3')
    info("h1 and h2:\n")
    net.iperf((h1,h2), l4Type = 'TCP')
    info("h1 and h3:\n")
    net.iperf((h1,h3), l4Type = 'TCP')
    info("h2 and h3:\n")
    net.iperf((h2,h3), l4Type = 'TCP')
    net.stop()


if __name__ == '__main__':
    setLogLevel( 'info' )
    # testing
    throughputTest()
```

Result

By running the above code, we get the following result.



From the result, we can see the loss rate is set to 5%. What's more, we can see the throughputs become much smaller than they in problem 1, which means the loss rate has a big influence on the actual throughput.

Problem 3

Analysis

In this problem, we should add another link between s2 and s3, then try pinging h2 from h1. In order to do this, we add the additional link by using the **addLink()** function. What's, we need to use the **CLI** of the **mininet**. Thus, we add two lines in the code.

```
self.addLink(s2, s3)
CLI( net )
```

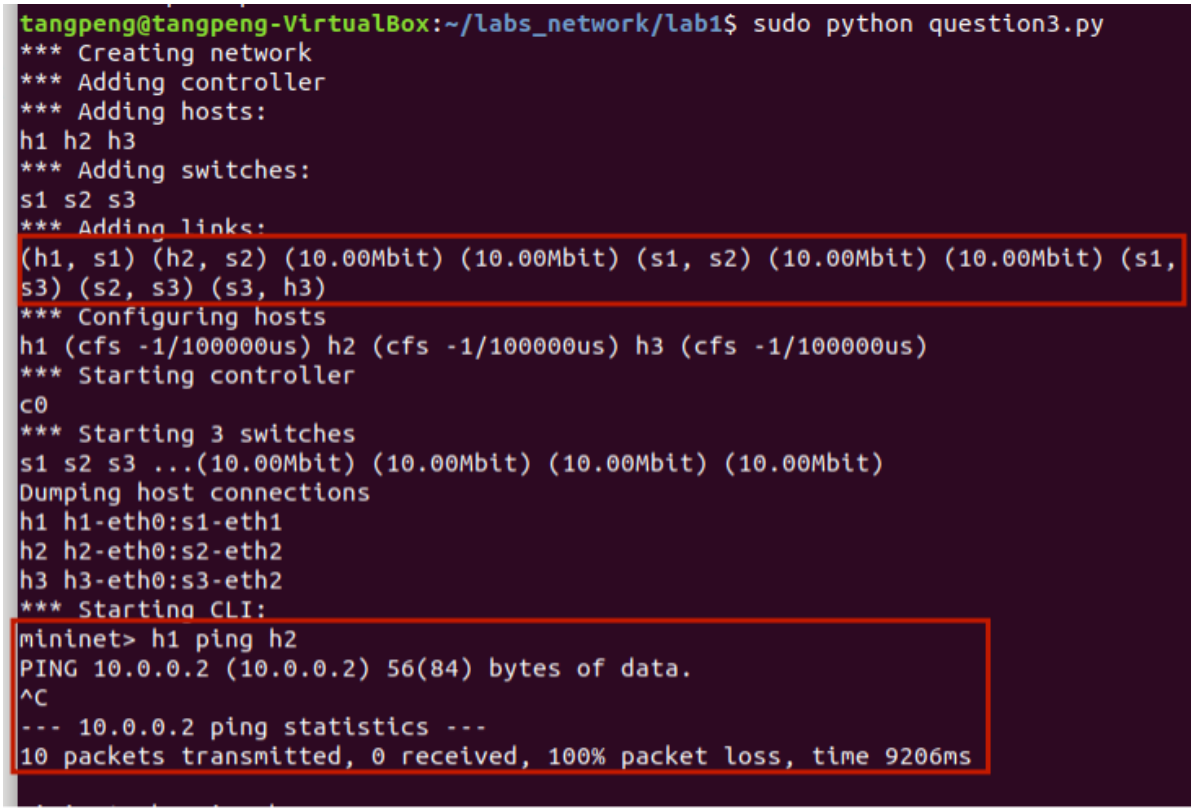
Then whole code is shown as follows:

```
#!/usr/bin/env python

import sys

from functools import partial

from mininet.net import Mininet
from mininet.node import UserSwitch, OVSKernelSwitch, Controller, CPULimitedHost
from mininet.topo import Topo
from mininet.log import lg, info, setLogLevel
from mininet.util import irange, quietRun
from mininet.util import dumpNodeConnections
from mininet.link import TCLink
from mininet.cli import CLI

class networkTopo(Topo):
    def build(self,**params):
        # define the host
```

```
        h1  = self.addHost('h1')
        h2  = self.addHost('h2')
        h3  = self.addHost('h3')
        #define the switch
        s1 = self.addSwitch('s1')
        s2 = self.addSwitch('s2')
        s3 = self.addSwitch('s3')

        #define the link
        self.addLink(h1, s1)
        self.addLink(s1, s2, bw=10)
        self.addLink(s1, s3, bw=10)
        self.addLink(s3, h3)
        self.addLink(h2, s2)
        self.addLink(s2, s3)

def throughputTest():
    #create the network
    topo = networkTopo()
    net = Mininet(topo=topo,
                  host=CPULimitedHost, link=TCLink,
                  autoStaticArp=True )

    net.start()
    info( "Dumping host connections\n" )
        dumpNodeConnections(net.hosts)
    CLI( net )
    net.stop()


if __name__ == '__main__':
    setLogLevel( 'info' )
    # testing
    throughputTest()
```

Result

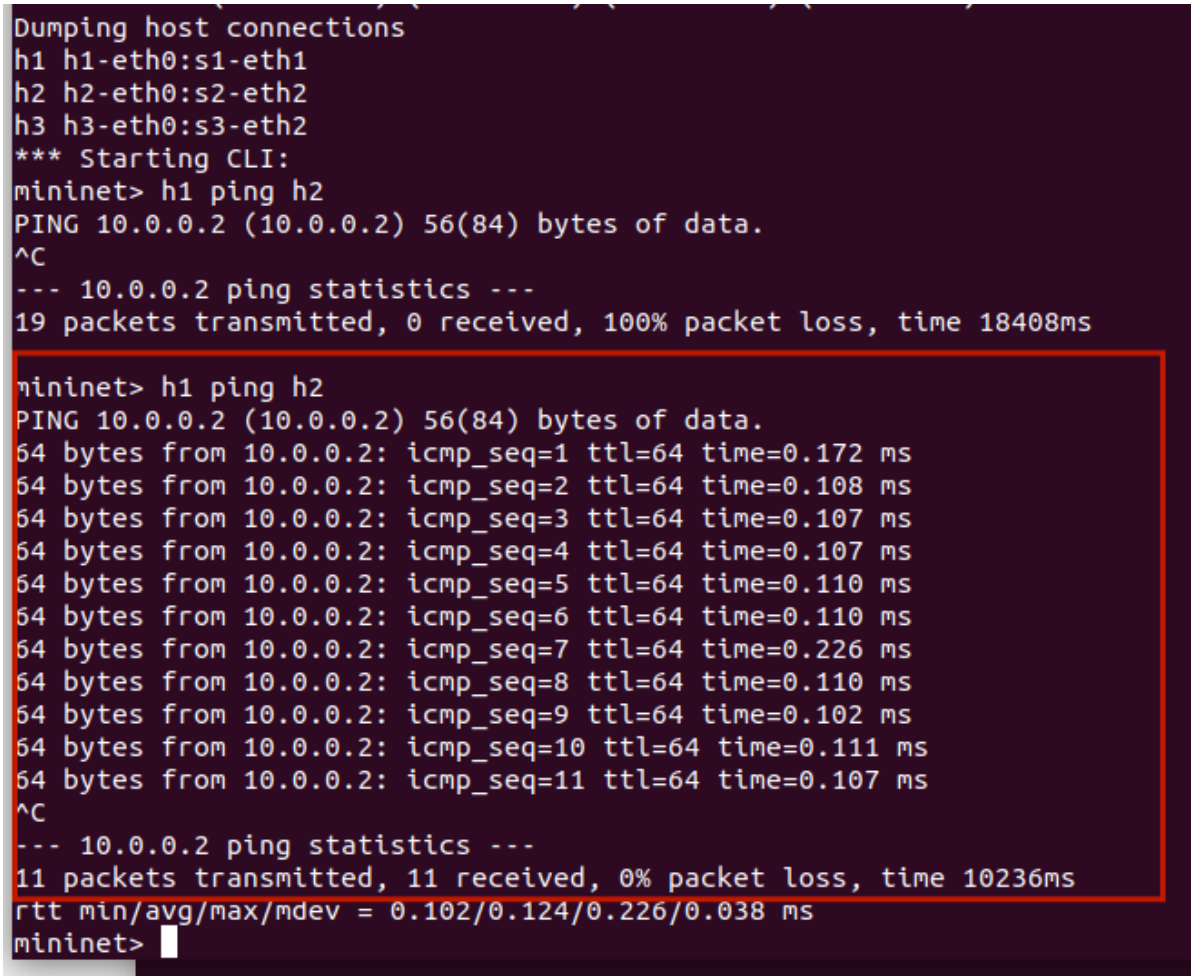
By running the above code, we try pinging h2 from h1, then we find this can't work and all the packets get lost. The following figure shows this phenomenon.



This is because when there are more than one path from h1 to h2, the mininet can't chose which one to transfer the packets, then it won't set up the connection between h1 and h2. Thus, we need set up it by ourselves. We can use the **sudo ovs-ofctl add-flow** command to set up one path. In this problem, we using the following four commands to connect h1 and h2.

```
sudo ovs-ofctl add-flow s1 "in_port=1 actions=output:2"
sudo ovs-ofctl add-flow s1 "in_port=2 actions=output:1"
sudo ovs-ofctl add-flow s2 "in_port=1 actions=output:2"
sudo ovs-ofctl add-flow s2 "in_port=2 actions=output:1"
```

The first two lines means we connect port1 and port2 of s1, and the second two lines means we connect port1 and port2 of s2. Because port1 of s1 is connected with h1 and port2 of s1 is connected with port1 of s2 and port2 of s2 is connected with h2, we set up a path $h1 - s1 - s2 - h2$ between h1 and h2. After doing this, we try pinging h2 from h1 again, we get the following result.



the figure shows that h2 can receive the packets from h1, which means we connect them successfully.

conclusion

By finishing this lab, I learned how to simulate a network in **mininet** and knew some basic functions about it.