计算机网络网络层实验2报告

马成 20307130112

定制化拓扑

1. 代码

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

from mininet.node import CPULimitedHost
from mininet.link import TCLink
from mininet.util import dumpNodeConnections
from mininet.log import setLogLevel
from mininet.node import OVSController
import time
import threading

class labtopo(Topo):
    def __init__(self):
        Topo.__init__(self)

        Host1 =
self.addHost("H1",ip_address="10.0.0.1",mac="00:00:00:00:ff:01")
        Host2 =
self.addHost("H2",ip_address="10.0.0.2",mac="00:00:00:00:ff:02")
        Host3 =
self.addHost("H3",ip_address="10.0.0.3",mac="00:00:00:00:ff:03")
        Host4 =
self.addHost("H4",ip_address="10.0.0.4",mac="00:00:00:00:ff:04")
        Switch1 = self.addSwitch("S1")
        Switch2 = self.addSwitch("S2")

        self.addLink(Host1,Switch1,bw=10,delay="2ms")
        self.addLink(Host2,Switch1,bw=20,delay="10ms")
        self.addLink(Host3,Switch2,bw=10,delay="2ms")
        self.addLink(Host4,Switch2,bw=20,delay="10ms")
        self.addLink(Switch1,Switch2,bw=20,delay="2ms",loss=10)

def perfTest():
    topo = labtopo()
    net = Mininet( topo=topo, link=TCLink, controller=OVSController)
    net.start()
    net.pingAll()
    h1,h2,h3,h4 = net.get( 'H1','H2','H3','H4')
    net.iperf( (h1, h2))
    net.iperf( (h2, h4))
    net.iperf( (h3, h4))
    net.stop()

if __name__ == '__main__':
```

```
    setLogLevel( 'info' )
    perfTest()
```

2. 输出

```
*** Creating network
*** Adding controller
*** Adding hosts:
H1 H2 H3 H4
*** Adding switches:
S1 S2
*** Adding links:
(10.00Mbit 2ms delay) (10.00Mbit 2ms delay) (H1, S1) (20.00Mbit 10ms delay)
(20.00Mbit 10ms delay) (H2, S1) (10.00Mbit 2ms delay) (10.00Mbit 2ms delay)
(H3, S2) (20.00Mbit 10ms delay) (20.00Mbit 10ms delay) (H4, S2) (20.00Mbit
2ms delay 10% loss) (20.00Mbit 2ms delay 10% loss) (S1, S2)
*** Configuring hosts
H1 H2 H3 H4
*** Starting controller
c0
*** Starting 2 switches
S1 S2 ...(10.00Mbit 2ms delay) (20.00Mbit 10ms delay) (20.00Mbit 2ms delay
10% loss) (10.00Mbit 2ms delay) (20.00Mbit 10ms delay) (20.00Mbit 2ms delay
10% loss)
*** Ping: testing ping reachability
H1 -> H2 H3 H4
H2 -> H1 H3 H4
H3 -> H1 X H4
H4 -> H1 H2 H3
*** Results: 8% dropped (11/12 received)
*** Iperf: testing TCP bandwidth between H1 and H2
*** Results: ['7.87 Mbits/sec', '9.13 Mbits/sec']
*** Iperf: testing TCP bandwidth between H2 and H4
*** Results: ['629 Kbits/sec', '724 Kbits/sec']
*** Iperf: testing TCP bandwidth between H3 and H4
*** Results: ['8.59 Mbits/sec', '9.93 Mbits/sec']
*** Stopping 1 controllers
c0
*** Stopping 5 links
.....
*** Stopping 2 switches
S1 S2
*** Stopping 4 hosts
H1 H2 H3 H4
*** Done
```

3. 结果

- H1和H2之间的带宽是['7.87 Mbits/sec', '9.13 Mbits/sec']与理论值接近
- H2和H4之间的带宽是['629 Kbits/sec', '724 Kbits/sec'] 与理论值相差较多
- H3和H4之间的带宽是['8.59 Mbits/sec', '9.93 Mbits/sec']与理论值接近
- 丢包率大约为8%和理论值较为接近

利用iperf生成TCP流

1. 代码

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

from mininet.node import CPULimitedHost
from mininet.link import TCLink
from mininet.util import dumpNodeConnections
from mininet.log import setLogLevel
from mininet.node import OVSController
import time
import threading

class labtopo(Topo):
    def __init__(self):
        Topo.__init__(self)

        Host1 =
self.addHost("H1",ip_address="10.0.0.1",mac="00:00:00:00:ff:01")
        Host2 =
self.addHost("H2",ip_address="10.0.0.2",mac="00:00:00:00:ff:02")
        Host3 =
self.addHost("H3",ip_address="10.0.0.3",mac="00:00:00:00:ff:03")
        Host4 =
self.addHost("H4",ip_address="10.0.0.4",mac="00:00:00:00:ff:04")
        Switch1 = self.addSwitch("S1")
        Switch2 = self.addSwitch("S2")

        self.addLink(Host1,Switch1,bw=10,delay="2ms")
        self.addLink(Host2,Switch1,bw=20,delay="10ms")
        self.addLink(Host3,Switch2,bw=10,delay="2ms")
        self.addLink(Host4,Switch2,bw=20,delay="10ms")
        self.addLink(Switch1,Switch2,bw=20,delay="2ms",loss=10)

def test1(h1,h3):
    result3 = h3.cmd('iperf -s -p 5001 & ')
    print(result3)
    result1 = h1.cmd('iperf -c 10.0.0.3 -p 5001 -t 20 -i 0.5')
    print(result1)
    # print(h1,h3)

def test2(h2,h4):
    time.sleep(10)
    result4 = h4.cmd('iperf -s -p 5002 & ')
    print(result4)
    result2 = h2.cmd('iperf -c 10.0.0.4 -p 5002 -t 20 -i 0.5 ')
    print(result2)

def perfTest2():
    "Create network and run simple performance test"
    topo = labtopo()
    net = Mininet( topo=topo, link=TCLink, controller=OVSController)
    net.start()
    print( "Dumping host connections" )
```

```
    dumpNodeConnections( net.hosts )

    print( "Testing bandwidth between h1 and h4" )
    h1,h2,h3,h4 = net.get( 'H1','H2','H3','H4')
    thread1=threading.Thread(group=None,target=test1,args=(h1,h3))
    thread2=threading.Thread(group=None,target=test2,args=(h2,h4))
    thread1.start()
    thread2.start()
    time.sleep(35)
    net.stop()

if __name__ == '__main__':
    setLogLevel( 'info' )
    # perfTest()
    perfTest2()
```

2. 输出

```
*** Creating network
*** Adding controller
*** Adding hosts:
H1 H2 H3 H4
*** Adding switches:
S1 S2
*** Adding links:
(10.00Mbit 2ms delay) (10.00Mbit 2ms delay) (H1, S1) (20.00Mbit 10ms delay)
(20.00Mbit 10ms delay) (H2, S1) (10.00Mbit 2ms delay) (10.00Mbit 2ms delay)
(H3, S2) (20.00Mbit 10ms delay) (20.00Mbit 10ms delay) (H4, S2) (20.00Mbit
2ms delay 10% loss) (20.00Mbit 2ms delay 10% loss) (S1, S2)
*** Configuring hosts
H1 H2 H3 H4
*** Starting controller
c0
*** Starting 2 switches
S1 S2 ...(10.00Mbit 2ms delay) (20.00Mbit 10ms delay) (20.00Mbit 2ms delay
10% loss) (10.00Mbit 2ms delay) (20.00Mbit 10ms delay) (20.00Mbit 2ms delay
10% loss)
Dumping host connections
H1 H1-eth0:S1-eth1
H2 H2-eth0:S1-eth2
H3 H3-eth0:S2-eth1
H4 H4-eth0:S2-eth2
Testing bandwidth between h1 and h4
[1] 22659

[1] 22741

------------------------------------------------------------
Client connecting to 10.0.0.3, TCP port 5001
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
[  3] local 10.0.0.1 port 52138 connected with 10.0.0.3 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0- 0.5 sec   638 KBytes  10.4 Mbits/sec
```

```
[  3]  0.5- 1.0 sec   191 KBytes  3.13 Mbits/sec
[  3]  1.0- 1.5 sec   127 KBytes  2.09 Mbits/sec
[  3]  1.5- 2.0 sec  63.6 KBytes  1.04 Mbits/sec
[  3]  2.0- 2.5 sec  0.00 Bytes  0.00 bits/sec
[  3]  2.5- 3.0 sec   127 KBytes  2.09 Mbits/sec
[  3]  3.0- 3.5 sec   191 KBytes  3.13 Mbits/sec
[  3]  3.5- 4.0 sec  63.6 KBytes  1.04 Mbits/sec
[  3]  4.0- 4.5 sec  63.6 KBytes  1.04 Mbits/sec
[  3]  4.5- 5.0 sec  0.00 Bytes  0.00 bits/sec
[  3]  5.0- 5.5 sec  0.00 Bytes  0.00 bits/sec
[  3]  5.5- 6.0 sec  0.00 Bytes  0.00 bits/sec
[  3]  6.0- 6.5 sec  63.6 KBytes  1.04 Mbits/sec
[  3]  6.5- 7.0 sec  63.6 KBytes  1.04 Mbits/sec
[  3]  7.0- 7.5 sec  0.00 Bytes  0.00 bits/sec
[  3]  7.5- 8.0 sec   127 KBytes  2.09 Mbits/sec
[  3]  8.0- 8.5 sec  63.6 KBytes  1.04 Mbits/sec
[  3]  8.5- 9.0 sec  63.6 KBytes  1.04 Mbits/sec
[  3]  9.0- 9.5 sec  0.00 Bytes  0.00 bits/sec
[  3]  9.5-10.0 sec   127 KBytes  2.09 Mbits/sec
[  3] 10.0-10.5 sec  63.6 KBytes  1.04 Mbits/sec
[  3] 10.5-11.0 sec  63.6 KBytes  1.04 Mbits/sec
[  3] 11.0-11.5 sec   127 KBytes  2.09 Mbits/sec
[  3] 11.5-12.0 sec  63.6 KBytes  1.04 Mbits/sec
[  3] 12.0-12.5 sec  63.6 KBytes  1.04 Mbits/sec
[  3] 12.5-13.0 sec   191 KBytes  3.13 Mbits/sec
[  3] 13.0-13.5 sec   127 KBytes  2.09 Mbits/sec
[  3] 13.5-14.0 sec  63.6 KBytes  1.04 Mbits/sec
[  3] 14.0-14.5 sec  63.6 KBytes  1.04 Mbits/sec
[  3] 14.5-15.0 sec   191 KBytes  3.13 Mbits/sec
[  3] 15.0-15.5 sec  63.6 KBytes  1.04 Mbits/sec
[  3] 15.5-16.0 sec   127 KBytes  2.09 Mbits/sec
[  3] 16.0-16.5 sec  63.6 KBytes  1.04 Mbits/sec
[  3] 16.5-17.0 sec  63.6 KBytes  1.04 Mbits/sec
[  3] 17.0-17.5 sec  63.6 KBytes  1.04 Mbits/sec
[  3] 17.5-18.0 sec   127 KBytes  2.09 Mbits/sec
[  3] 18.0-18.5 sec  63.6 KBytes  1.04 Mbits/sec
[  3] 18.5-19.0 sec  63.6 KBytes  1.04 Mbits/sec
[  3] 19.0-19.5 sec  63.6 KBytes  1.04 Mbits/sec
[  3] 19.5-20.0 sec  63.6 KBytes  1.04 Mbits/sec
[  3]  0.0-20.1 sec  3.67 MBytes  1.53 Mbits/sec

------------------------------------------------------------
Client connecting to 10.0.0.4, TCP port 5002
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
[  3] local 10.0.0.2 port 39092 connected with 10.0.0.4 port 5002
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0- 0.5 sec   230 KBytes  3.78 Mbits/sec
[  3]  0.5- 1.0 sec  93.3 KBytes  1.53 Mbits/sec
[  3]  1.0- 1.5 sec  63.6 KBytes  1.04 Mbits/sec
[  3]  1.5- 2.0 sec  0.00 Bytes  0.00 bits/sec
[  3]  2.0- 2.5 sec  63.6 KBytes  1.04 Mbits/sec
[  3]  2.5- 3.0 sec  63.6 KBytes  1.04 Mbits/sec
[  3]  3.0- 3.5 sec  0.00 Bytes  0.00 bits/sec
[  3]  3.5- 4.0 sec  63.6 KBytes  1.04 Mbits/sec
```

```
[  3]  4.0- 4.5 sec  63.6 KBytes  1.04 Mbits/sec
[  3]  4.5- 5.0 sec  0.00 Bytes  0.00 bits/sec
[  3]  5.0- 5.5 sec  0.00 Bytes  0.00 bits/sec
[  3]  5.5- 6.0 sec  63.6 KBytes  1.04 Mbits/sec
[  3]  6.0- 6.5 sec  0.00 Bytes  0.00 bits/sec
[  3]  6.5- 7.0 sec   127 KBytes  2.09 Mbits/sec
[  3]  7.0- 7.5 sec  0.00 Bytes  0.00 bits/sec
[  3]  7.5- 8.0 sec  63.6 KBytes  1.04 Mbits/sec
[  3]  8.0- 8.5 sec  63.6 KBytes  1.04 Mbits/sec
[  3]  8.5- 9.0 sec  63.6 KBytes  1.04 Mbits/sec
[  3]  9.0- 9.5 sec  0.00 Bytes  0.00 bits/sec
[  3]  9.5-10.0 sec  0.00 Bytes  0.00 bits/sec
[  3] 10.0-10.5 sec  0.00 Bytes  0.00 bits/sec
[  3] 10.5-11.0 sec  0.00 Bytes  0.00 bits/sec
[  3] 11.0-11.5 sec  0.00 Bytes  0.00 bits/sec
[  3] 11.5-12.0 sec  63.6 KBytes  1.04 Mbits/sec
[  3] 12.0-12.5 sec  0.00 Bytes  0.00 bits/sec
[  3] 12.5-13.0 sec  0.00 Bytes  0.00 bits/sec
[  3] 13.0-13.5 sec  63.6 KBytes  1.04 Mbits/sec
[  3] 13.5-14.0 sec  0.00 Bytes  0.00 bits/sec
[  3] 14.0-14.5 sec  63.6 KBytes  1.04 Mbits/sec
[  3] 14.5-15.0 sec  0.00 Bytes  0.00 bits/sec
[  3] 15.0-15.5 sec  63.6 KBytes  1.04 Mbits/sec
[  3] 15.5-16.0 sec  63.6 KBytes  1.04 Mbits/sec
[  3] 16.0-16.5 sec  0.00 Bytes  0.00 bits/sec
[  3] 16.5-17.0 sec  63.6 KBytes  1.04 Mbits/sec
[  3] 17.0-17.5 sec  0.00 Bytes  0.00 bits/sec
[  3] 17.5-18.0 sec  63.6 KBytes  1.04 Mbits/sec
[  3] 18.0-18.5 sec  0.00 Bytes  0.00 bits/sec
[  3] 18.5-19.0 sec  63.6 KBytes  1.04 Mbits/sec
[  3] 19.0-19.5 sec  0.00 Bytes  0.00 bits/sec
[  3] 19.5-20.0 sec  0.00 Bytes  0.00 bits/sec
[  3]  0.0-20.2 sec  1.50 MBytes   620 Kbits/sec

*** Stopping 1 controllers
c0
*** Stopping 5 links
.....
*** Stopping 2 switches
S1 S2
*** Stopping 4 hosts
H1 H2 H3 H4
*** Done
```

3. 分析

    1. 把H3、H4作为服务器，H1、H2作为客户端分别与这两个服务器建立TCP连接，连接时长为20s，但是H2和H4的TCP建立时间晚10s
    2. 可以看到两边TCP的连接带宽都远小于理论值有的时候甚至会降为0，猜测可能拥塞有关