Lab 4: Overlay Network and VXLAN

518030910283 王航宇

First, I get the two VMs' IP address by the command:

```
ifconfig
```

IP address on VM1: 192.168.164.3

IP address on VM2: 192.168.164.4

Then I construct the network topology on VM1 and VM2 by the files "build_1.py" and "build_2.py". All the links bandwith is 10Mbps, the packet loss rate is 0%, and the time delay is 5ms

```
class MultiSwitchTopo(Topo):
    "Multi switches connected to hosts."

    def build(self):
        s1 = self.addSwitch('s1')
        h1 = self.addHost('h1')
        h2 = self.addHost('h2')
        self.addLink(s1, h1,bw=10,loss=0,delay='5ms')
        self.addLink(s1, h2,bw=10,loss=0,delay='5ms')

def simpleTest():
    "Create and test a simple network"
    topo = MultiSwitchTopo()
    net = Mininet(topo, link=TCLink)
    net.start()
    CLI(net)
    net.stop()
```

Then I follow the steps in the file lab4-questions.pdf. Note that I use 192.168.164.5 as the default route for br1:

```
sudo route add default gw 192.168.164.5
```

Homework 1:

(30 points) Ping 10.0.0.111 from 10.0.0.112 (in your terminal of VM2) and use Wireshark to monitor the interfaces s2 and enp0s8, and describe the protocols used in this procedure and your findings.

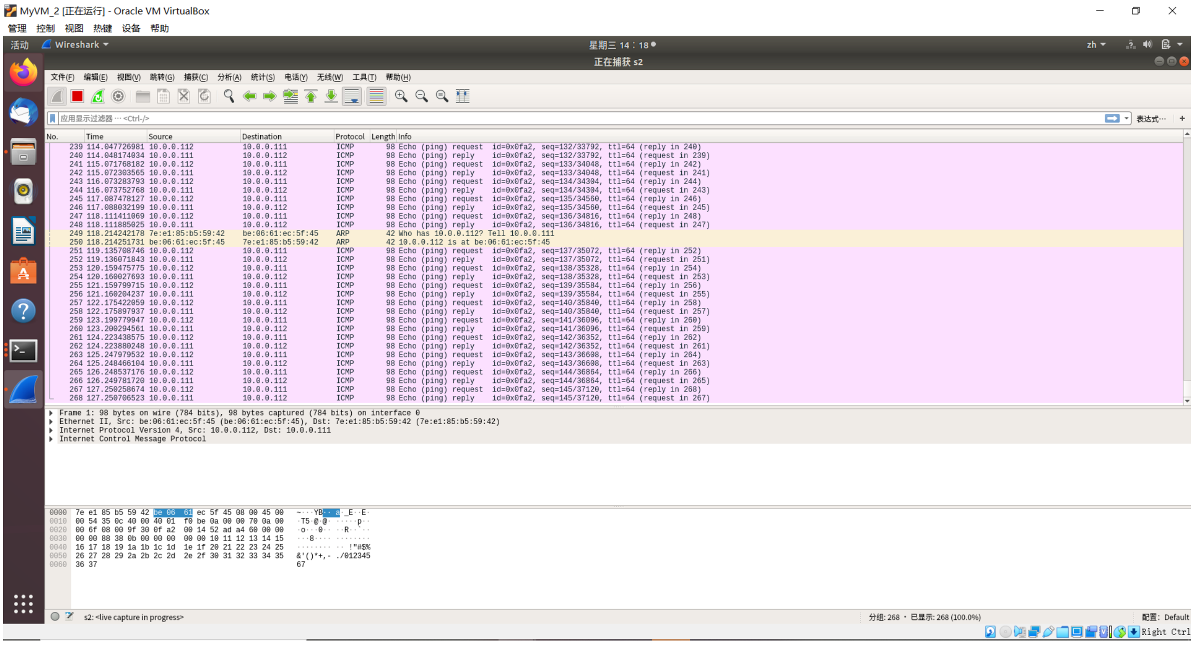
Requirements:

Please provide the screenshot of the wireshark interface in your report. Your description and findings should align with the information in wireshark interface

In the terminal of VM2, I enter the command to ping 10.0.0.111:

```
ping 10.0.0.111
```

The screenshot of Wireshark monitoring the interfaces s2 is:

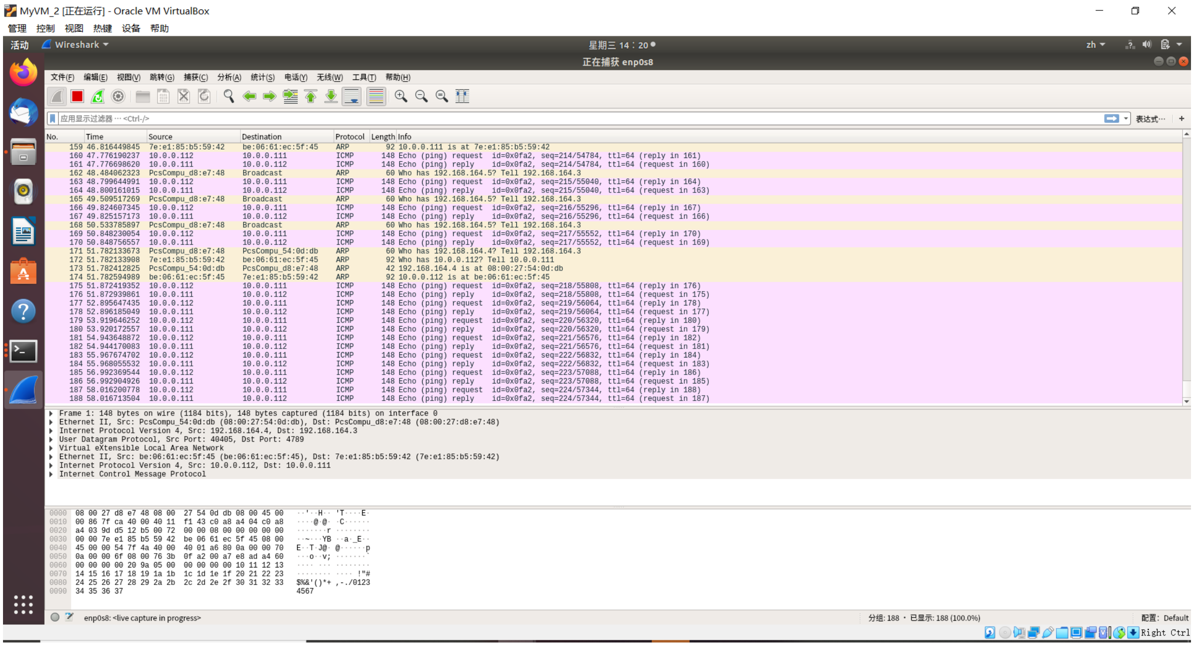


MyVM_2 [正在运行] - Oracle VM VirtualBox
管理 控制 视图 热键 设备 帮助
活动
Wireshark
星期三 14：18
正在捕获 s2
文件(F) 编辑(E) 视图(V) 跳转(G) 捕获(C) 分析(A) 统计(S) 电话(Y) 无线(W) 工具(T) 帮助(H)
应用显示过滤器 … <Ctrl-/>
表达式…

```
No.     Time          Source              Destination         Protocol Length Info
    239 114.047726981 10.0.0.112          10.0.0.111          ICMP       98 Echo (ping) request  id=0x0fa2, seq=132/33792, ttl=64 (reply in 240)
    240 114.048174034 10.0.0.111          10.0.0.112          ICMP       98 Echo (ping) reply    id=0x0fa2, seq=132/33792, ttl=64 (request in 239)
    241 115.071768182 10.0.0.112          10.0.0.111          ICMP       98 Echo (ping) request  id=0x0fa2, seq=133/34048, ttl=64 (reply in 242)
    242 115.072303565 10.0.0.111          10.0.0.112          ICMP       98 Echo (ping) reply    id=0x0fa2, seq=133/34048, ttl=64 (request in 241)
    243 116.073283793 10.0.0.112          10.0.0.111          ICMP       98 Echo (ping) request  id=0x0fa2, seq=134/34304, ttl=64 (reply in 244)
    244 116.073752768 10.0.0.111          10.0.0.112          ICMP       98 Echo (ping) reply    id=0x0fa2, seq=134/34304, ttl=64 (request in 243)
    245 117.087478127 10.0.0.112          10.0.0.111          ICMP       98 Echo (ping) request  id=0x0fa2, seq=135/34560, ttl=64 (reply in 246)
    246 117.088032199 10.0.0.111          10.0.0.112          ICMP       98 Echo (ping) reply    id=0x0fa2, seq=135/34560, ttl=64 (request in 245)
    247 118.111411069 10.0.0.112          10.0.0.111          ICMP       98 Echo (ping) request  id=0x0fa2, seq=136/34816, ttl=64 (reply in 248)
    248 118.111885025 10.0.0.111          10.0.0.112          ICMP       98 Echo (ping) reply    id=0x0fa2, seq=136/34816, ttl=64 (request in 247)
    249 118.214242178 7e:e1:85:b5:59:42   be:06:61:ec:5f:45   ARP        42 Who has 10.0.0.112? Tell 10.0.0.111
    250 118.214251731 be:06:61:ec:5f:45   7e:e1:85:b5:59:42   ARP        42 10.0.0.112 is at be:06:61:ec:5f:45
    251 119.135708746 10.0.0.112          10.0.0.111          ICMP       98 Echo (ping) request  id=0x0fa2, seq=137/35072, ttl=64 (reply in 252)
    252 119.136071843 10.0.0.111          10.0.0.112          ICMP       98 Echo (ping) reply    id=0x0fa2, seq=137/35072, ttl=64 (request in 251)
    253 120.159475775 10.0.0.112          10.0.0.111          ICMP       98 Echo (ping) request  id=0x0fa2, seq=138/35328, ttl=64 (reply in 254)
    254 120.160027693 10.0.0.111          10.0.0.112          ICMP       98 Echo (ping) reply    id=0x0fa2, seq=138/35328, ttl=64 (request in 253)
    255 121.159799715 10.0.0.112          10.0.0.111          ICMP       98 Echo (ping) request  id=0x0fa2, seq=139/35584, ttl=64 (reply in 256)
    256 121.160204237 10.0.0.111          10.0.0.112          ICMP       98 Echo (ping) reply    id=0x0fa2, seq=139/35584, ttl=64 (request in 255)
    257 122.175422059 10.0.0.112          10.0.0.111          ICMP       98 Echo (ping) request  id=0x0fa2, seq=140/35840, ttl=64 (reply in 258)
    258 122.175897937 10.0.0.111          10.0.0.112          ICMP       98 Echo (ping) reply    id=0x0fa2, seq=140/35840, ttl=64 (request in 257)
    259 123.199779947 10.0.0.112          10.0.0.111          ICMP       98 Echo (ping) request  id=0x0fa2, seq=141/36096, ttl=64 (reply in 260)
    260 123.200294561 10.0.0.111          10.0.0.112          ICMP       98 Echo (ping) reply    id=0x0fa2, seq=141/36096, ttl=64 (request in 259)
    261 124.223438575 10.0.0.112          10.0.0.111          ICMP       98 Echo (ping) request  id=0x0fa2, seq=142/36352, ttl=64 (reply in 262)
    262 124.223880248 10.0.0.111          10.0.0.112          ICMP       98 Echo (ping) reply    id=0x0fa2, seq=142/36352, ttl=64 (request in 261)
    263 125.247979532 10.0.0.112          10.0.0.111          ICMP       98 Echo (ping) request  id=0x0fa2, seq=143/36608, ttl=64 (reply in 264)
    264 125.248466104 10.0.0.111          10.0.0.112          ICMP       98 Echo (ping) reply    id=0x0fa2, seq=143/36608, ttl=64 (request in 263)
    265 126.248537176 10.0.0.112          10.0.0.111          ICMP       98 Echo (ping) request  id=0x0fa2, seq=144/36864, ttl=64 (reply in 266)
    266 126.249781720 10.0.0.111          10.0.0.112          ICMP       98 Echo (ping) reply    id=0x0fa2, seq=144/36864, ttl=64 (request in 265)
    267 127.250258674 10.0.0.112          10.0.0.111          ICMP       98 Echo (ping) request  id=0x0fa2, seq=145/37120, ttl=64 (reply in 268)
    268 127.250706523 10.0.0.111          10.0.0.112          ICMP       98 Echo (ping) reply    id=0x0fa2, seq=145/37120, ttl=64 (request in 267)

▸ Frame 1: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface 0
▸ Ethernet II, Src: be:06:61:ec:5f:45 (be:06:61:ec:5f:45), Dst: 7e:e1:85:b5:59:42 (7e:e1:85:b5:59:42)
▸ Internet Protocol Version 4, Src: 10.0.0.112, Dst: 10.0.0.111
▸ Internet Control Message Protocol

0000  7e e1 85 b5 59 42 be 06  61 ec 5f 45 08 00 45 00   ~···YB·· a·_E··E·
0010  00 54 35 0c 40 00 40 01  f0 be 0a 00 00 70 0a 00   ·T5·@·@· ·····p··
0020  00 6f 08 00 9f 30 0f a2  00 14 52 ad a4 60 00 00   ·o···0·· ··R··`··
0030  00 00 88 38 0b 00 00 00  00 00 10 11 12 13 14 15   ···8···· ········
0040  16 17 18 19 1a 1b 1c 1d  1e 1f 20 21 22 23 24 25   ········ ·· !"#$%
0050  26 27 28 29 2a 2b 2c 2d  2e 2f 30 31 32 33 34 35   &'()*+,- ./012345
0060  36 37                                              67

s2: <live capture in progress>        分组: 268 · 已显示: 268 (100.0%)        配置: Default
```

From this screenshot, we can see that Internet Control Message Protocol(ICMP), Internet Protocol Version 4(IPv4) and Address Resolution Protocol(ARP) are used in this procedure.

The screenshot of Wireshark monitoring the interfaces enp0s8 is:



MyVM_2 [正在运行] - Oracle VM VirtualBox
管理 控制 视图 热键 设备 帮助
活动
Wireshark
星期三 14：20
正在捕获 enp0s8
文件(F) 编辑(E) 视图(V) 跳转(G) 捕获(C) 分析(A) 统计(S) 电话(Y) 无线(W) 工具(T) 帮助(H)
应用显示过滤器 … <Ctrl-/>
表达式…

```
No.     Time          Source              Destination         Protocol Length Info
    159 46.816449845  7e:e1:85:b5:59:42   be:06:61:ec:5f:45   ARP        92 10.0.0.111 is at 7e:e1:85:b5:59:42
    160 47.776190237  10.0.0.112          10.0.0.111          ICMP      148 Echo (ping) request  id=0x0fa2, seq=214/54784, ttl=64 (reply in 161)
    161 47.776698620  10.0.0.111          10.0.0.112          ICMP      148 Echo (ping) reply    id=0x0fa2, seq=214/54784, ttl=64 (request in 160)
    162 48.484062323  PcsCompu_d8:e7:48   Broadcast           ARP        60 Who has 192.168.164.5? Tell 192.168.164.3
    163 48.799644991  10.0.0.112          10.0.0.111          ICMP      148 Echo (ping) request  id=0x0fa2, seq=215/55040, ttl=64 (reply in 164)
    164 48.800161015  10.0.0.111          10.0.0.112          ICMP      148 Echo (ping) reply    id=0x0fa2, seq=215/55040, ttl=64 (request in 163)
    165 49.509517269  PcsCompu_d8:e7:48   Broadcast           ARP        60 Who has 192.168.164.5? Tell 192.168.164.3
    166 49.824607345  10.0.0.112          10.0.0.111          ICMP      148 Echo (ping) request  id=0x0fa2, seq=216/55296, ttl=64 (reply in 167)
    167 49.825157173  10.0.0.111          10.0.0.112          ICMP      148 Echo (ping) reply    id=0x0fa2, seq=216/55296, ttl=64 (request in 166)
    168 50.533785897  PcsCompu_d8:e7:48   Broadcast           ARP        60 Who has 192.168.164.5? Tell 192.168.164.3
    169 50.848230054  10.0.0.112          10.0.0.111          ICMP      148 Echo (ping) request  id=0x0fa2, seq=217/55552, ttl=64 (reply in 170)
    170 50.848756557  10.0.0.111          10.0.0.112          ICMP      148 Echo (ping) reply    id=0x0fa2, seq=217/55552, ttl=64 (request in 169)
    171 51.782133673  PcsCompu_d8:e7:48   PcsCompu_54:0d:db   ARP        60 Who has 192.168.164.4? Tell 192.168.164.3
    172 51.782133908  7e:e1:85:b5:59:42   be:06:61:ec:5f:45   ARP        92 Who has 10.0.0.112? Tell 10.0.0.111
    173 51.782412825  PcsCompu_54:0d:db   PcsCompu_d8:e7:48   ARP        42 192.168.164.4 is at 08:00:27:54:0d:db
    174 51.782594989  be:06:61:ec:5f:45   7e:e1:85:b5:59:42   ARP        92 10.0.0.112 is at be:06:61:ec:5f:45
    175 51.872419352  10.0.0.112          10.0.0.111          ICMP      148 Echo (ping) request  id=0x0fa2, seq=218/55808, ttl=64 (reply in 176)
    176 51.872939861  10.0.0.111          10.0.0.112          ICMP      148 Echo (ping) reply    id=0x0fa2, seq=218/55808, ttl=64 (request in 175)
    177 52.895647435  10.0.0.112          10.0.0.111          ICMP      148 Echo (ping) request  id=0x0fa2, seq=219/56064, ttl=64 (reply in 178)
    178 52.896185049  10.0.0.111          10.0.0.112          ICMP      148 Echo (ping) reply    id=0x0fa2, seq=219/56064, ttl=64 (request in 177)
    179 53.919646252  10.0.0.112          10.0.0.111          ICMP      148 Echo (ping) request  id=0x0fa2, seq=220/56320, ttl=64 (reply in 180)
    180 53.920172557  10.0.0.111          10.0.0.112          ICMP      148 Echo (ping) reply    id=0x0fa2, seq=220/56320, ttl=64 (request in 179)
    181 54.943648872  10.0.0.112          10.0.0.111          ICMP      148 Echo (ping) request  id=0x0fa2, seq=221/56576, ttl=64 (reply in 182)
    182 54.944170083  10.0.0.111          10.0.0.112          ICMP      148 Echo (ping) reply    id=0x0fa2, seq=221/56576, ttl=64 (request in 181)
    183 55.967674702  10.0.0.112          10.0.0.111          ICMP      148 Echo (ping) request  id=0x0fa2, seq=222/56832, ttl=64 (reply in 184)
    184 55.968055532  10.0.0.111          10.0.0.112          ICMP      148 Echo (ping) reply    id=0x0fa2, seq=222/56832, ttl=64 (request in 183)
    185 56.992369544  10.0.0.112          10.0.0.111          ICMP      148 Echo (ping) request  id=0x0fa2, seq=223/57088, ttl=64 (reply in 186)
    186 56.992904926  10.0.0.111          10.0.0.112          ICMP      148 Echo (ping) reply    id=0x0fa2, seq=223/57088, ttl=64 (request in 185)
    187 58.016200778  10.0.0.112          10.0.0.111          ICMP      148 Echo (ping) request  id=0x0fa2, seq=224/57344, ttl=64 (reply in 188)
    188 58.016713504  10.0.0.111          10.0.0.112          ICMP      148 Echo (ping) reply    id=0x0fa2, seq=224/57344, ttl=64 (request in 187)

▸ Frame 1: 148 bytes on wire (1184 bits), 148 bytes captured (1184 bits) on interface 0
▸ Ethernet II, Src: PcsCompu_54:0d:db (08:00:27:54:0d:db), Dst: PcsCompu_d8:e7:48 (08:00:27:d8:e7:48)
▸ Internet Protocol Version 4, Src: 192.168.164.4, Dst: 192.168.164.3
▸ User Datagram Protocol, Src Port: 40405, Dst Port: 4789
▸ Virtual eXtensible Local Area Network
▸ Ethernet II, Src: be:06:61:ec:5f:45 (be:06:61:ec:5f:45), Dst: 7e:e1:85:b5:59:42 (7e:e1:85:b5:59:42)
▸ Internet Protocol Version 4, Src: 10.0.0.112, Dst: 10.0.0.111
▸ Internet Control Message Protocol

0000  08 00 27 d8 e7 48 08 00  27 54 0d db 08 00 45 00   ··'··H·· 'T····E·
0010  00 86 7f ca 40 00 40 11  f1 43 c0 a8 a4 04 c0 a8   ····@·@· ·C······
0020  a4 03 9d d5 12 b5 00 72  00 00 08 00 00 00 00 00   ·······r ········
0030  00 00 7e e1 85 b5 59 42  be 06 61 ec 5f 45 08 00   ··~···YB ··a·_E··
0040  45 00 00 54 7f 4a 40 00  40 01 a6 80 0a 00 00 70   E··T·J@· @······p
0050  0a 00 00 6f 08 00 76 3b  0f a2 00 a7 e8 ad a4 60   ···o··v; ·······`
0060  00 00 00 00 20 9a 05 00  00 00 00 00 10 11 12 13   ···· ··· ········
0070  14 15 16 17 18 19 1a 1b  1c 1d 1e 1f 20 21 22 23   ········ ···· !"#
0080  24 25 26 27 28 29 2a 2b  2c 2d 2e 2f 30 31 32 33   $%&'()*+ ,-./0123
0090  34 35 36 37                                        4567

enp0s8: <live capture in progress>        分组: 188 · 已显示: 188 (100.0%)        配置: Default
```

From this screenshot, we can see that Internet Control Message Protocol(ICMP), Internet Protocol Version 4(IPv4), Address Resolution Protocol(ARP) and User Datagram Protocol(UDP) are used in this procedure.

Decriptions:

ICMP is a network layer protocol used by network devices to diagnose network communication issues. ICMP is mainly used to determine whether or not data is reaching its intended destination in a timely manner.

IPv4 is a connectionless protocol, and operates on a best-effort delivery model, in that it does not guarantee delivery, nor does it assure proper sequencing or avoidance of duplicate delivery.

ARP is a protocol used by the Internet Protocol, specifically IPv4, to map IP network addresses to the hardware addresses used by a data link protocol.

UDP is a lightweight data transport protocol that works on top of IP. UDP provides a mechanism to detect corrupt data in packets, but it does not attempt to solve other problems that arise with packets, such as lost or out of order packets.

Findings:

I find that enp0s8 and s2 both use ICMP, IPv4 and ARP, but enp0s8 uses one more protocol than s2, that's UDP. Moreover, the frame length in enp0s8 is 148 bytes, which is longer than 98 bytes, the frame length in s2. The extra 50 bytes are added by VXLAN.

Homework 2:

(50 points) Use iperf to test the network bandwidth between the two virtual machine

• Test the bandwidth between 192.168.56.101 and 192.168.56.102

• Test the bandwidth between 10.0.0.1/10.0.0.2/10.0.0.111 and 10.0.0.3

Compare the above results and explain the reason. (Hint: you may need to specify a reasonable MTU size in order for your iperf to work in this case. Please also think about why.)

Requirements:

Please provide the screenshot of your terminals in your report.

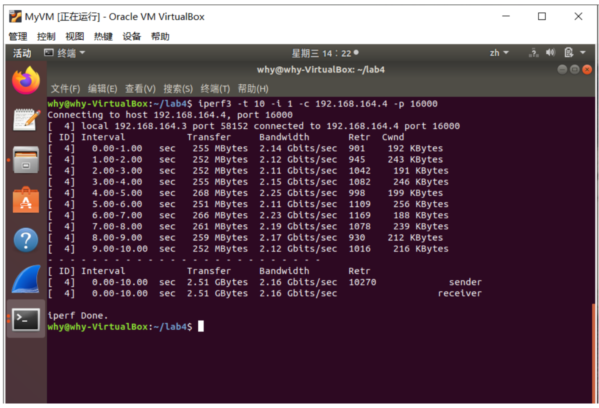
In the terminal of VM1, I enter the command to test the bandwidth between 192.168.164.4 and 192.168.164.3:

```
iperf3 -t 10 -i 1 -c 192.168.164.4 -p 16000
```

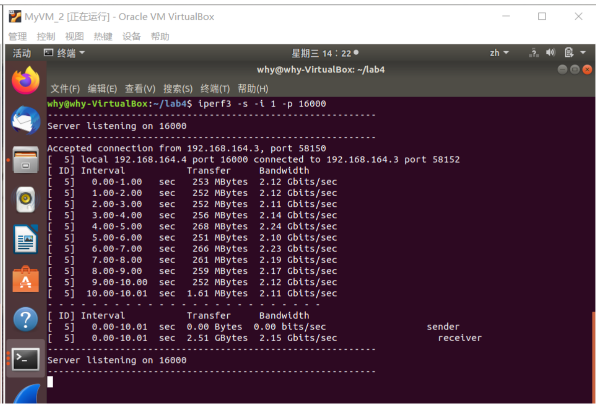
and in VM2, I enter:

```
iperf3 -s -i 1 -p 16000
```

The bandwidth between 192.168.164.3 and 192.168.164.4 is 2.16 Gbits/sec


MyVM [正在运行] - Oracle VM VirtualBox
管理 控制 视图 热键 设备 帮助
活动 终端
星期三 14：22
zh
why@why-VirtualBox: ~/lab4
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)

```
why@why-VirtualBox:~/lab4$ iperf3 -t 10 -i 1 -c 192.168.164.4 -p 16000
Connecting to host 192.168.164.4, port 16000
[  4] local 192.168.164.3 port 58152 connected to 192.168.164.4 port 16000
[ ID] Interval           Transfer     Bandwidth       Retr  Cwnd
[  4]   0.00-1.00   sec   255 MBytes  2.14 Gbits/sec  901    192 KBytes
[  4]   1.00-2.00   sec   252 MBytes  2.12 Gbits/sec  945    243 KBytes
[  4]   2.00-3.00   sec   252 MBytes  2.11 Gbits/sec  1042    191 KBytes
[  4]   3.00-4.00   sec   255 MBytes  2.15 Gbits/sec  1082    246 KBytes
[  4]   4.00-5.00   sec   268 MBytes  2.25 Gbits/sec  998    199 KBytes
[  4]   5.00-6.00   sec   251 MBytes  2.11 Gbits/sec  1109    256 KBytes
[  4]   6.00-7.00   sec   266 MBytes  2.23 Gbits/sec  1169    188 KBytes
[  4]   7.00-8.00   sec   261 MBytes  2.19 Gbits/sec  1078    239 KBytes
[  4]   8.00-9.00   sec   259 MBytes  2.17 Gbits/sec  930    212 KBytes
[  4]   9.00-10.00  sec   252 MBytes  2.12 Gbits/sec  1016    216 KBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth       Retr
[  4]   0.00-10.00  sec  2.51 GBytes  2.16 Gbits/sec  10270             sender
[  4]   0.00-10.00  sec  2.51 GBytes  2.16 Gbits/sec                  receiver

iperf Done.
why@why-VirtualBox:~/lab4$ 
```


MyVM_2 [正在运行] - Oracle VM VirtualBox
管理 控制 视图 热键 设备 帮助
活动 终端
星期三 14：22
zh
why@why-VirtualBox: ~/lab4
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)

```
why@why-VirtualBox:~/lab4$ iperf3 -s -i 1 -p 16000
-----------------------------------------------------------
Server listening on 16000
-----------------------------------------------------------
Accepted connection from 192.168.164.3, port 58150
[  5] local 192.168.164.4 port 16000 connected to 192.168.164.3 port 58152
[ ID] Interval           Transfer     Bandwidth
[  5]   0.00-1.00   sec   253 MBytes  2.12 Gbits/sec
[  5]   1.00-2.00   sec   252 MBytes  2.12 Gbits/sec
[  5]   2.00-3.00   sec   252 MBytes  2.11 Gbits/sec
[  5]   3.00-4.00   sec   256 MBytes  2.14 Gbits/sec
[  5]   4.00-5.00   sec   268 MBytes  2.24 Gbits/sec
[  5]   5.00-6.00   sec   251 MBytes  2.10 Gbits/sec
[  5]   6.00-7.00   sec   266 MBytes  2.23 Gbits/sec
[  5]   7.00-8.00   sec   261 MBytes  2.19 Gbits/sec
[  5]   8.00-9.00   sec   259 MBytes  2.17 Gbits/sec
[  5]   9.00-10.00  sec   252 MBytes  2.12 Gbits/sec
[  5]  10.00-10.01  sec  1.61 MBytes  2.11 Gbits/sec
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth
[  5]   0.00-10.01  sec  0.00 Bytes  0.00 bits/sec                  sender
[  5]   0.00-10.01  sec  2.51 GBytes  2.15 Gbits/sec                  receiver
-----------------------------------------------------------
Server listening on 16000
-----------------------------------------------------------
```

In the xterm of 10.0.0.1/10.0.0.2/10.0.0.111, I all enter:

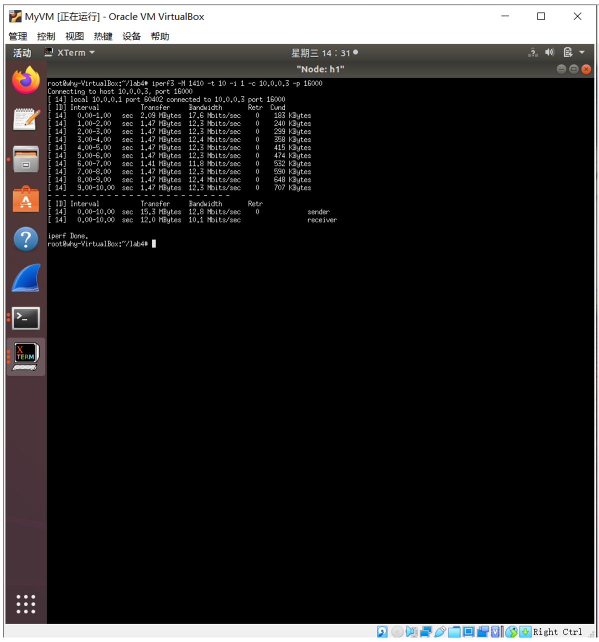
```
iperf3 -M 1410 -t 10 -i 1 -c 10.0.0.3 -p 16000
```

and in the xterm of 10.0.0.3, I enter:

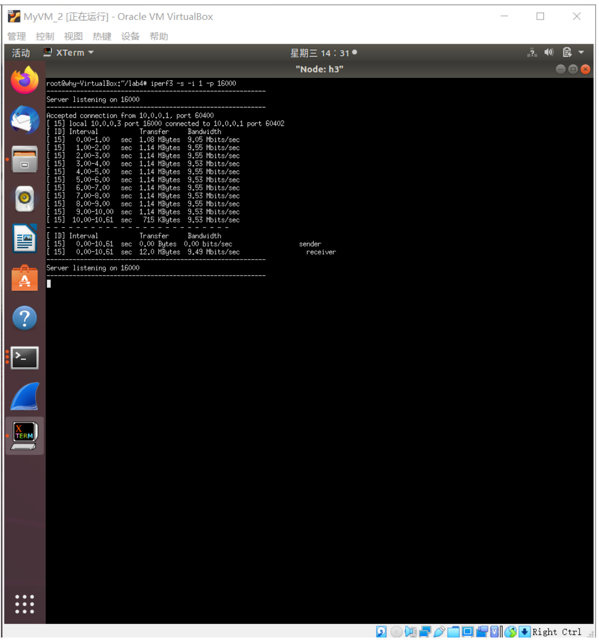
```
iperf3 -s -i 1 -p 16000
```

-M set the maximum segment size(= $MTU - 40$ bytes), I find the maximum value of -M is 1410 (so MTU$= 1450$) and if -M is 1411, the iperf won't work. That's because the default MTU of the virtual machine is 1500 Bytes, and when this message passes through the VTEP (VXLAN Tunnel End Point), a new header of 50 bytes will be encapsulated. so we need to lower MTU size(maximum MTU=1500-50=1450) in order to avoid the segment size exceeding the limit.

The bandwidth between 10.0.0.1 and 10.0.0.3 are 9.49 Mbits/sec.

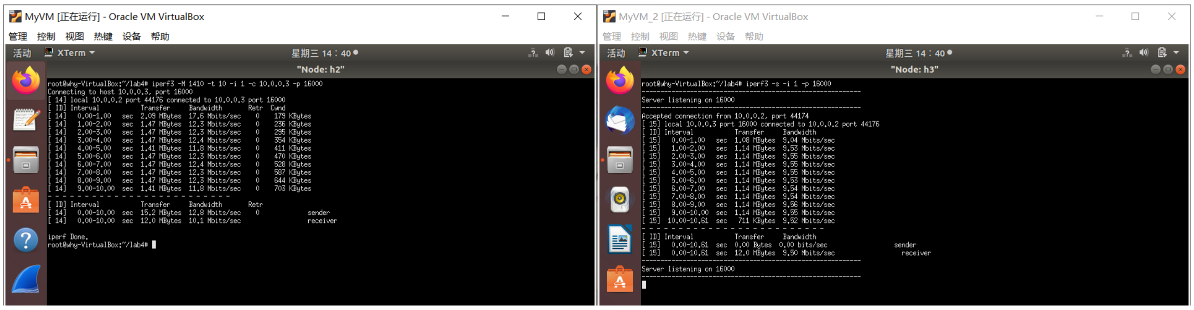

MyVM [正在运行] - Oracle VM VirtualBox
管理 控制 视图 热键 设备 帮助
活动 XTerm
星期三 14：31
"Node: h1"
Right Ctrl

```
root@why-VirtualBox:~/lab4# iperf3 -M 1410 -t 10 -i 1 -c 10.0.0.3 -p 16000
Connecting to host 10.0.0.3, port 16000
[ 14] local 10.0.0.1 port 60402 connected to 10.0.0.3 port 16000
[ ID] Interval           Transfer     Bandwidth       Retr  Cwnd
[ 14]   0.00-1.00   sec  2.09 MBytes  17.6 Mbits/sec    0    183 KBytes
[ 14]   1.00-2.00   sec  1.47 MBytes  12.3 Mbits/sec    0    240 KBytes
[ 14]   2.00-3.00   sec  1.47 MBytes  12.3 Mbits/sec    0    299 KBytes
[ 14]   3.00-4.00   sec  1.47 MBytes  12.4 Mbits/sec    0    358 KBytes
[ 14]   4.00-5.00   sec  1.47 MBytes  12.3 Mbits/sec    0    415 KBytes
[ 14]   5.00-6.00   sec  1.47 MBytes  12.3 Mbits/sec    0    474 KBytes
[ 14]   6.00-7.00   sec  1.41 MBytes  11.8 Mbits/sec    0    532 KBytes
[ 14]   7.00-8.00   sec  1.47 MBytes  12.3 Mbits/sec    0    590 KBytes
[ 14]   8.00-9.00   sec  1.47 MBytes  12.4 Mbits/sec    0    648 KBytes
[ 14]   9.00-10.00  sec  1.47 MBytes  12.3 Mbits/sec    0    707 KBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth       Retr
[ 14]   0.00-10.00  sec  15.3 MBytes  12.8 Mbits/sec    0             sender
[ 14]   0.00-10.00  sec  12.0 MBytes  10.1 Mbits/sec                  receiver

iperf Done.
root@why-VirtualBox:~/lab4# 
```


MyVM_2 [正在运行] - Oracle VM VirtualBox
管理 控制 视图 热键 设备 帮助
活动 XTerm
星期三 14：31
"Node: h3"
Right Ctrl

```
root@why-VirtualBox:~/lab4# iperf3 -s -i 1 -p 16000
-----------------------------------------------------------
Server listening on 16000
-----------------------------------------------------------
Accepted connection from 10.0.0.1, port 60400
[ 15] local 10.0.0.3 port 16000 connected to 10.0.0.1 port 60402
[ ID] Interval           Transfer     Bandwidth
[ 15]   0.00-1.00   sec  1.08 MBytes  9.05 Mbits/sec
[ 15]   1.00-2.00   sec  1.14 MBytes  9.55 Mbits/sec
[ 15]   2.00-3.00   sec  1.14 MBytes  9.55 Mbits/sec
[ 15]   3.00-4.00   sec  1.14 MBytes  9.53 Mbits/sec
[ 15]   4.00-5.00   sec  1.14 MBytes  9.55 Mbits/sec
[ 15]   5.00-6.00   sec  1.14 MBytes  9.53 Mbits/sec
[ 15]   6.00-7.00   sec  1.14 MBytes  9.55 Mbits/sec
[ 15]   7.00-8.00   sec  1.14 MBytes  9.53 Mbits/sec
[ 15]   8.00-9.00   sec  1.14 MBytes  9.55 Mbits/sec
[ 15]   9.00-10.00  sec  1.14 MBytes  9.53 Mbits/sec
[ 15]  10.00-10.61  sec   715 KBytes  9.53 Mbits/sec
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth
[ 15]   0.00-10.61  sec  0.00 Bytes  0.00 bits/sec                  sender
[ 15]   0.00-10.61  sec  12.0 MBytes  9.49 Mbits/sec                  receiver
-----------------------------------------------------------
Server listening on 16000
-----------------------------------------------------------
```

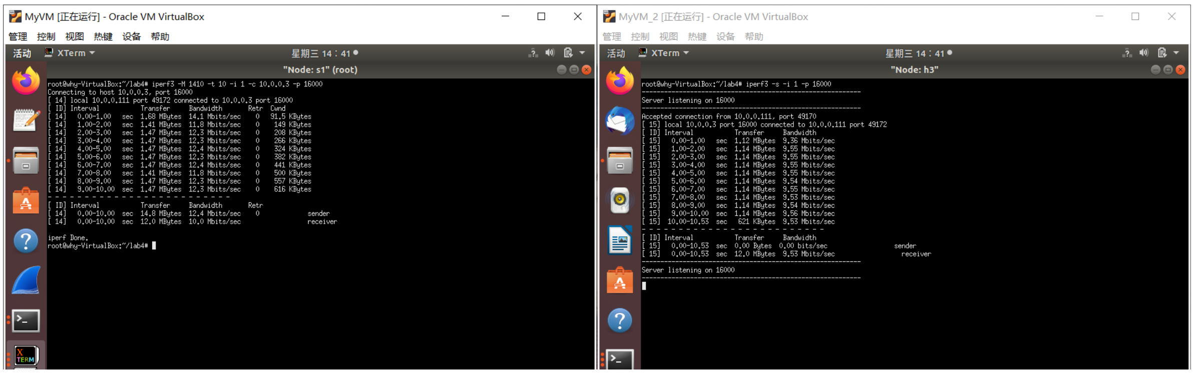
The bandwidth between 10.0.0.2 and 10.0.0.3 are 9.50 Mbits/sec.


MyVM [正在运行] - Oracle VM VirtualBox
管理 控制 视图 热键 设备 帮助
活动 XTerm
星期三 14：40
"Node: h2"
MyVM_2 [正在运行] - Oracle VM VirtualBox
管理 控制 视图 热键 设备 帮助
活动 XTerm
星期三 14：40
"Node: h3"

```
root@why-VirtualBox:~/lab4# iperf3 -M 1410 -t 10 -i 1 -c 10.0.0.3 -p 16000
Connecting to host 10.0.0.3, port 16000
[ 14] local 10.0.0.2 port 44176 connected to 10.0.0.3 port 16000
[ ID] Interval           Transfer     Bandwidth       Retr  Cwnd
[ 14]   0.00-1.00   sec  2.09 MBytes  17.6 Mbits/sec    0    179 KBytes
[ 14]   1.00-2.00   sec  1.47 MBytes  12.3 Mbits/sec    0    236 KBytes
[ 14]   2.00-3.00   sec  1.47 MBytes  12.3 Mbits/sec    0    295 KBytes
[ 14]   3.00-4.00   sec  1.47 MBytes  12.4 Mbits/sec    0    354 KBytes
[ 14]   4.00-5.00   sec  1.41 MBytes  11.8 Mbits/sec    0    411 KBytes
[ 14]   5.00-6.00   sec  1.47 MBytes  12.3 Mbits/sec    0    470 KBytes
[ 14]   6.00-7.00   sec  1.47 MBytes  12.4 Mbits/sec    0    528 KBytes
[ 14]   7.00-8.00   sec  1.47 MBytes  12.3 Mbits/sec    0    587 KBytes
[ 14]   8.00-9.00   sec  1.47 MBytes  12.3 Mbits/sec    0    644 KBytes
[ 14]   9.00-10.00  sec  1.41 MBytes  11.8 Mbits/sec    0    703 KBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth       Retr
[ 14]   0.00-10.00  sec  15.2 MBytes  12.8 Mbits/sec    0             sender
[ 14]   0.00-10.00  sec  12.0 MBytes  10.1 Mbits/sec                  receiver

iperf Done.
root@why-VirtualBox:~/lab4# 
```

```
root@why-VirtualBox:~/lab4# iperf3 -s -i 1 -p 16000
-----------------------------------------------------------
Server listening on 16000
-----------------------------------------------------------
Accepted connection from 10.0.0.2, port 44174
[ 15] local 10.0.0.3 port 16000 connected to 10.0.0.2 port 44176
[ ID] Interval           Transfer     Bandwidth
[ 15]   0.00-1.00   sec  1.08 MBytes  9.04 Mbits/sec
[ 15]   1.00-2.00   sec  1.14 MBytes  9.53 Mbits/sec
[ 15]   2.00-3.00   sec  1.14 MBytes  9.55 Mbits/sec
[ 15]   3.00-4.00   sec  1.14 MBytes  9.55 Mbits/sec
[ 15]   4.00-5.00   sec  1.14 MBytes  9.55 Mbits/sec
[ 15]   5.00-6.00   sec  1.14 MBytes  9.53 Mbits/sec
[ 15]   6.00-7.00   sec  1.14 MBytes  9.54 Mbits/sec
[ 15]   7.00-8.00   sec  1.14 MBytes  9.54 Mbits/sec
[ 15]   8.00-9.00   sec  1.14 MBytes  9.56 Mbits/sec
[ 15]   9.00-10.00  sec  1.14 MBytes  9.55 Mbits/sec
[ 15]  10.00-10.61  sec   711 KBytes  9.52 Mbits/sec
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth
[ 15]   0.00-10.61  sec  0.00 Bytes  0.00 bits/sec                  sender
[ 15]   0.00-10.61  sec  12.0 MBytes  9.50 Mbits/sec                  receiver
-----------------------------------------------------------
Server listening on 16000
-----------------------------------------------------------
```

The bandwidth between 10.0.0.111 and 10.0.0.3 are 9.53 Mbits/sec.


MyVM [正在运行] - Oracle VM VirtualBox
管理 控制 视图 热键 设备 帮助
活动 XTerm
星期三 14：41
"Node: s1" (root)
MyVM_2 [正在运行] - Oracle VM VirtualBox
管理 控制 视图 热键 设备 帮助
活动 XTerm
星期三 14：41
"Node: h3"

```
root@why-VirtualBox:~/lab4# iperf3 -M 1410 -t 10 -i 1 -c 10.0.0.3 -p 16000
Connecting to host 10.0.0.3, port 16000
[ 14] local 10.0.0.111 port 49172 connected to 10.0.0.3 port 16000
[ ID] Interval           Transfer     Bandwidth       Retr  Cwnd
[ 14]   0.00-1.00   sec  1.68 MBytes  14.1 Mbits/sec    0   91.5 KBytes
[ 14]   1.00-2.00   sec  1.41 MBytes  11.8 Mbits/sec    0    149 KBytes
[ 14]   2.00-3.00   sec  1.47 MBytes  12.3 Mbits/sec    0    208 KBytes
[ 14]   3.00-4.00   sec  1.47 MBytes  12.3 Mbits/sec    0    266 KBytes
[ 14]   4.00-5.00   sec  1.47 MBytes  12.4 Mbits/sec    0    324 KBytes
[ 14]   5.00-6.00   sec  1.47 MBytes  12.3 Mbits/sec    0    382 KBytes
[ 14]   6.00-7.00   sec  1.47 MBytes  12.4 Mbits/sec    0    441 KBytes
[ 14]   7.00-8.00   sec  1.41 MBytes  11.8 Mbits/sec    0    500 KBytes
[ 14]   8.00-9.00   sec  1.47 MBytes  12.3 Mbits/sec    0    557 KBytes
[ 14]   9.00-10.00  sec  1.47 MBytes  12.3 Mbits/sec    0    616 KBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth       Retr
[ 14]   0.00-10.00  sec  14.8 MBytes  12.4 Mbits/sec    0             sender
[ 14]   0.00-10.00  sec  12.0 MBytes  10.0 Mbits/sec                  receiver

iperf Done.
root@why-VirtualBox:~/lab4# 
```

```
root@why-VirtualBox:~/lab4# iperf3 -s -i 1 -p 16000
-----------------------------------------------------------
Server listening on 16000
-----------------------------------------------------------
Accepted connection from 10.0.0.111, port 49170
[ 15] local 10.0.0.3 port 16000 connected to 10.0.0.111 port 49172
[ ID] Interval           Transfer     Bandwidth
[ 15]   0.00-1.00   sec  1.12 MBytes  9.36 Mbits/sec
[ 15]   1.00-2.00   sec  1.14 MBytes  9.55 Mbits/sec
[ 15]   2.00-3.00   sec  1.14 MBytes  9.55 Mbits/sec
[ 15]   3.00-4.00   sec  1.14 MBytes  9.55 Mbits/sec
[ 15]   4.00-5.00   sec  1.14 MBytes  9.55 Mbits/sec
[ 15]   5.00-6.00   sec  1.14 MBytes  9.54 Mbits/sec
[ 15]   6.00-7.00   sec  1.14 MBytes  9.55 Mbits/sec
[ 15]   7.00-8.00   sec  1.14 MBytes  9.53 Mbits/sec
[ 15]   8.00-9.00   sec  1.14 MBytes  9.54 Mbits/sec
[ 15]   9.00-10.00  sec  1.14 MBytes  9.56 Mbits/sec
[ 15]  10.00-10.53  sec   621 KBytes  9.53 Mbits/sec
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth
[ 15]   0.00-10.53  sec  0.00 Bytes  0.00 bits/sec                  sender
[ 15]   0.00-10.53  sec  12.0 MBytes  9.53 Mbits/sec                  receiver
-----------------------------------------------------------
Server listening on 16000
-----------------------------------------------------------
```

We can find that

- the bandwidths between 10.0.0.1/10.0.0.2/10.0.0.111 and 10.0.0.3 are almost the same
- the bandwidth between 192.168.164.4 and 192.168.164.3 is larger than the bandwidths between 10.0.0.1/10.0.0.2/10.0.0.111 and 10.0.0.3.

That's because we set all the links bandwidth as 10 Mbps, so even the real bandwidth can reach 2.16 Gbits/sec, the bandwidth in the network topology can't exceed 10 Mbps.

Homework 3:

(20 points) Similar to Q2, use ping to test the network latency and analyze your results.

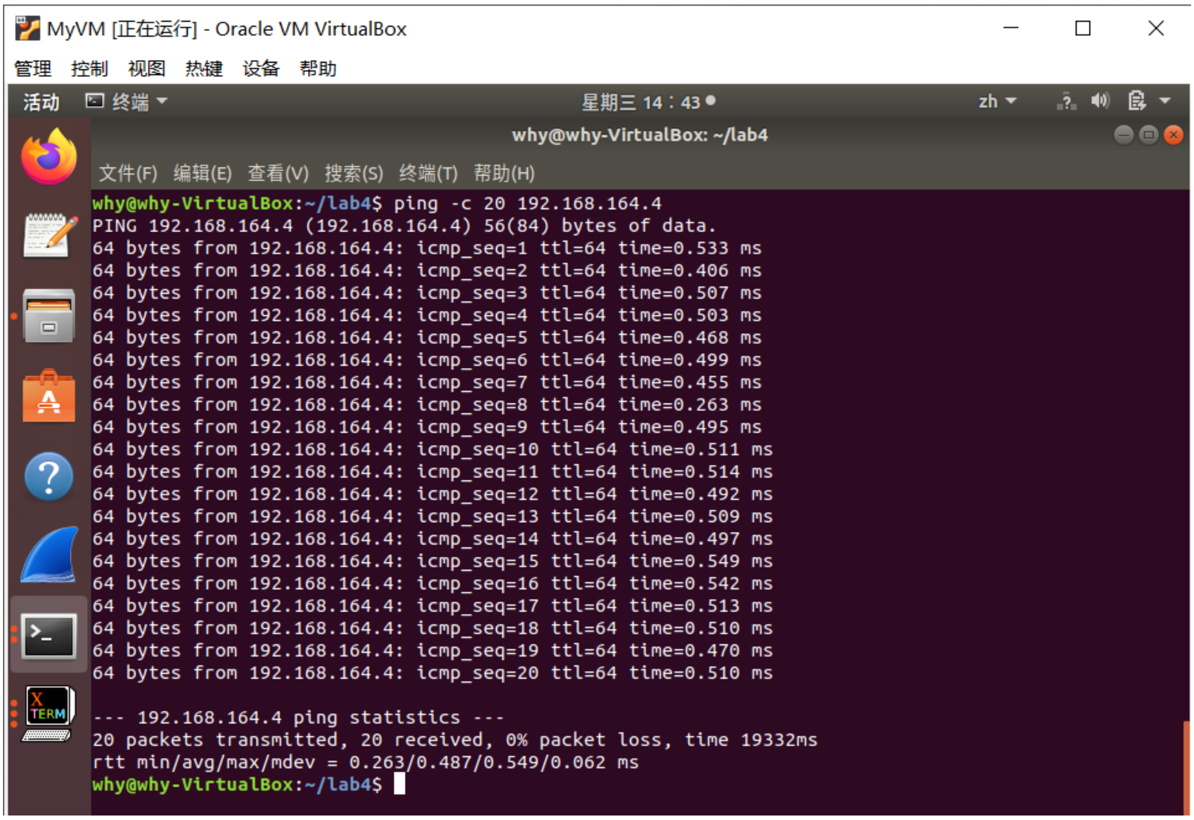
Requirements:

Please provide the screenshot of your terminals in your report.

In the terminal of VM1, I enter the command to test the RTT between 192.168.164.4 and 192.168.164.3:

```
ping -c 20 192.168.164.4
```

The average RTT between 192.168.164.4 and 192.168.164.3 is 0.487 ms.

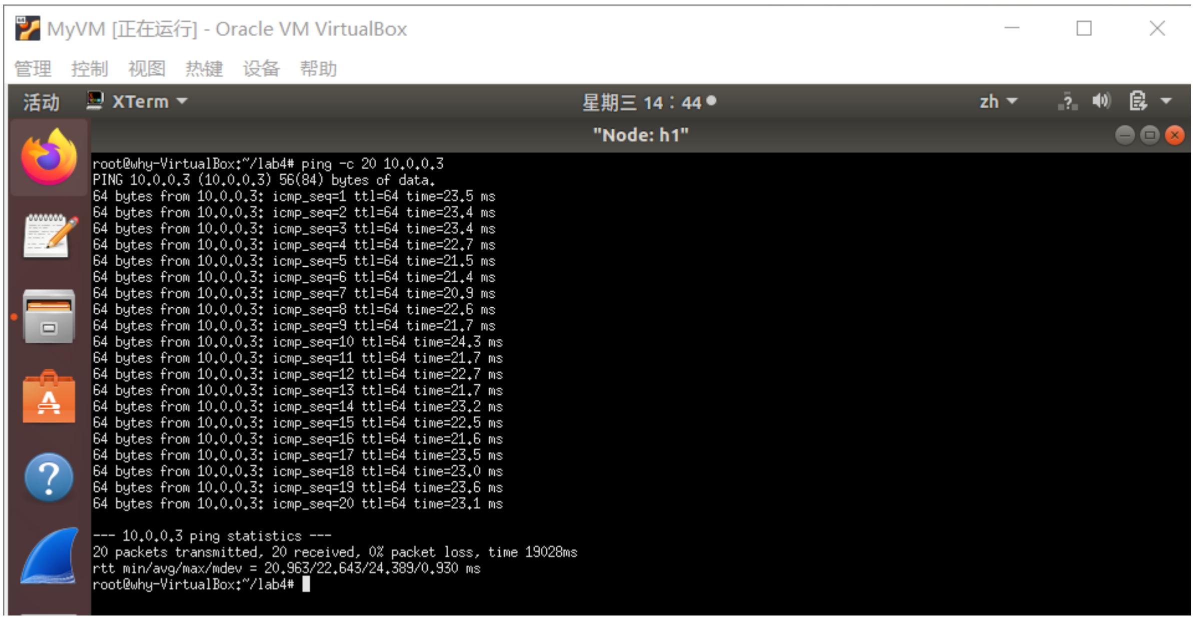

MyVM [正在运行] - Oracle VM VirtualBox
管理 控制 视图 热键 设备 帮助
活动 终端
星期三 14：43
zh
why@why-VirtualBox: ~/lab4
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)

```
why@why-VirtualBox:~/lab4$ ping -c 20 192.168.164.4
PING 192.168.164.4 (192.168.164.4) 56(84) bytes of data.
64 bytes from 192.168.164.4: icmp_seq=1 ttl=64 time=0.533 ms
64 bytes from 192.168.164.4: icmp_seq=2 ttl=64 time=0.406 ms
64 bytes from 192.168.164.4: icmp_seq=3 ttl=64 time=0.507 ms
64 bytes from 192.168.164.4: icmp_seq=4 ttl=64 time=0.503 ms
64 bytes from 192.168.164.4: icmp_seq=5 ttl=64 time=0.468 ms
64 bytes from 192.168.164.4: icmp_seq=6 ttl=64 time=0.499 ms
64 bytes from 192.168.164.4: icmp_seq=7 ttl=64 time=0.455 ms
64 bytes from 192.168.164.4: icmp_seq=8 ttl=64 time=0.263 ms
64 bytes from 192.168.164.4: icmp_seq=9 ttl=64 time=0.495 ms
64 bytes from 192.168.164.4: icmp_seq=10 ttl=64 time=0.511 ms
64 bytes from 192.168.164.4: icmp_seq=11 ttl=64 time=0.514 ms
64 bytes from 192.168.164.4: icmp_seq=12 ttl=64 time=0.492 ms
64 bytes from 192.168.164.4: icmp_seq=13 ttl=64 time=0.509 ms
64 bytes from 192.168.164.4: icmp_seq=14 ttl=64 time=0.497 ms
64 bytes from 192.168.164.4: icmp_seq=15 ttl=64 time=0.549 ms
64 bytes from 192.168.164.4: icmp_seq=16 ttl=64 time=0.542 ms
64 bytes from 192.168.164.4: icmp_seq=17 ttl=64 time=0.513 ms
64 bytes from 192.168.164.4: icmp_seq=18 ttl=64 time=0.510 ms
64 bytes from 192.168.164.4: icmp_seq=19 ttl=64 time=0.470 ms
64 bytes from 192.168.164.4: icmp_seq=20 ttl=64 time=0.510 ms

--- 192.168.164.4 ping statistics ---
20 packets transmitted, 20 received, 0% packet loss, time 19332ms
rtt min/avg/max/mdev = 0.263/0.487/0.549/0.062 ms
why@why-VirtualBox:~/lab4$
```

Then I test the latency between 10.0.0.1/10.0.0.2/10.0.0.111 and 10.0.0.3 respectively:

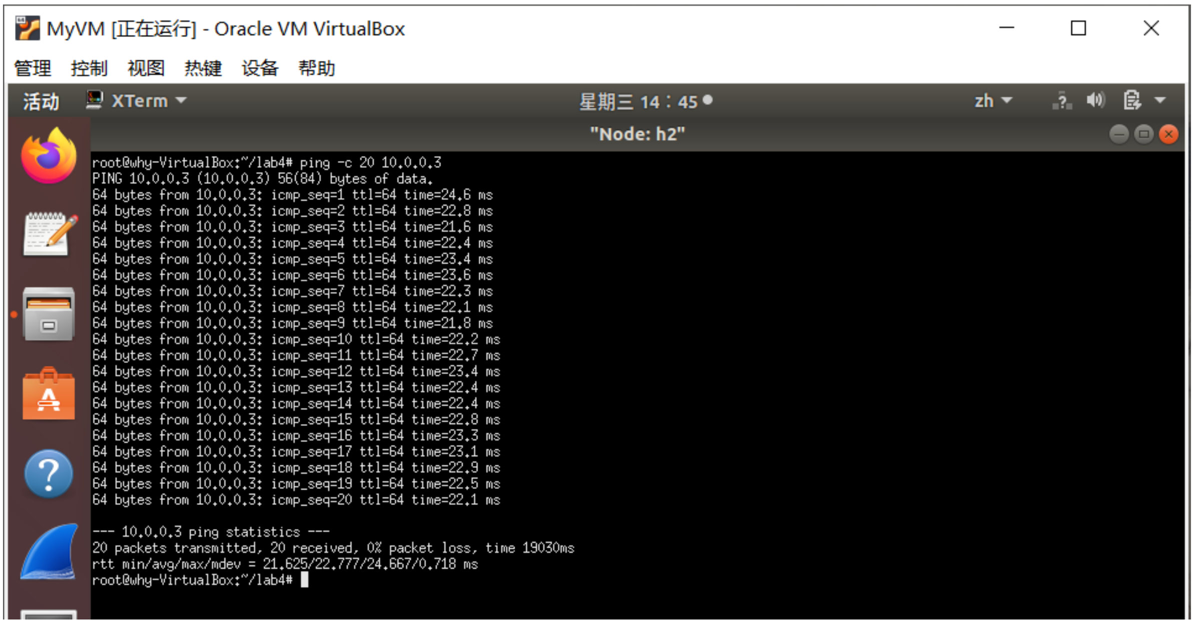
```
ping -c 20 10.0.0.3
```

The average RTT between 10.0.0.1 and 10.0.0.3 is 22.643 ms.


MyVM [正在运行] - Oracle VM VirtualBox
管理 控制 视图 热键 设备 帮助
活动 XTerm
星期三 14：44
zh
"Node: h1"

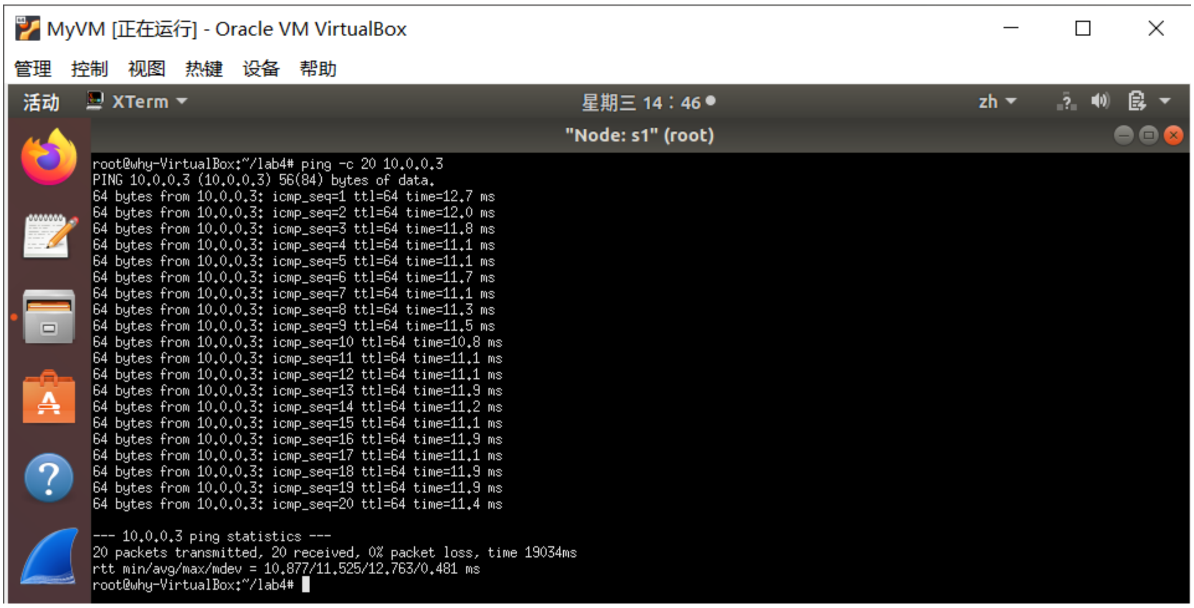
```
root@why-VirtualBox:~/lab4# ping -c 20 10.0.0.3
PING 10.0.0.3 (10.0.0.3) 56(84) bytes of data.
64 bytes from 10.0.0.3: icmp_seq=1 ttl=64 time=23.5 ms
64 bytes from 10.0.0.3: icmp_seq=2 ttl=64 time=23.4 ms
64 bytes from 10.0.0.3: icmp_seq=3 ttl=64 time=23.4 ms
64 bytes from 10.0.0.3: icmp_seq=4 ttl=64 time=22.7 ms
64 bytes from 10.0.0.3: icmp_seq=5 ttl=64 time=21.5 ms
64 bytes from 10.0.0.3: icmp_seq=6 ttl=64 time=21.4 ms
64 bytes from 10.0.0.3: icmp_seq=7 ttl=64 time=20.9 ms
64 bytes from 10.0.0.3: icmp_seq=8 ttl=64 time=22.6 ms
64 bytes from 10.0.0.3: icmp_seq=9 ttl=64 time=21.7 ms
64 bytes from 10.0.0.3: icmp_seq=10 ttl=64 time=24.3 ms
64 bytes from 10.0.0.3: icmp_seq=11 ttl=64 time=21.7 ms
64 bytes from 10.0.0.3: icmp_seq=12 ttl=64 time=22.7 ms
64 bytes from 10.0.0.3: icmp_seq=13 ttl=64 time=21.7 ms
64 bytes from 10.0.0.3: icmp_seq=14 ttl=64 time=23.2 ms
64 bytes from 10.0.0.3: icmp_seq=15 ttl=64 time=22.5 ms
64 bytes from 10.0.0.3: icmp_seq=16 ttl=64 time=21.6 ms
64 bytes from 10.0.0.3: icmp_seq=17 ttl=64 time=23.5 ms
64 bytes from 10.0.0.3: icmp_seq=18 ttl=64 time=23.0 ms
64 bytes from 10.0.0.3: icmp_seq=19 ttl=64 time=23.6 ms
64 bytes from 10.0.0.3: icmp_seq=20 ttl=64 time=23.1 ms

--- 10.0.0.3 ping statistics ---
20 packets transmitted, 20 received, 0% packet loss, time 19028ms
rtt min/avg/max/mdev = 20.963/22.643/24.389/0.930 ms
root@why-VirtualBox:~/lab4#
```

The average RTT between 10.0.0.2 and 10.0.0.3 is 22.777 ms.


MyVM [正在运行] - Oracle VM VirtualBox
管理 控制 视图 热键 设备 帮助
活动 XTerm
星期三 14：45
zh
"Node: h2"

```
root@why-VirtualBox:~/lab4# ping -c 20 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=24.6 ms
64 bytes from 10.0.0.3: icmp_seq=2 ttl=64 time=22.8 ms
64 bytes from 10.0.0.3: icmp_seq=3 ttl=64 time=21.6 ms
64 bytes from 10.0.0.3: icmp_seq=4 ttl=64 time=22.4 ms
64 bytes from 10.0.0.3: icmp_seq=5 ttl=64 time=23.4 ms
64 bytes from 10.0.0.3: icmp_seq=6 ttl=64 time=23.6 ms
64 bytes from 10.0.0.3: icmp_seq=7 ttl=64 time=22.3 ms
64 bytes from 10.0.0.3: icmp_seq=8 ttl=64 time=22.1 ms
64 bytes from 10.0.0.3: icmp_seq=9 ttl=64 time=21.8 ms
64 bytes from 10.0.0.3: icmp_seq=10 ttl=64 time=22.2 ms
64 bytes from 10.0.0.3: icmp_seq=11 ttl=64 time=22.7 ms
64 bytes from 10.0.0.3: icmp_seq=12 ttl=64 time=23.4 ms
64 bytes from 10.0.0.3: icmp_seq=13 ttl=64 time=22.4 ms
64 bytes from 10.0.0.3: icmp_seq=14 ttl=64 time=22.4 ms
64 bytes from 10.0.0.3: icmp_seq=15 ttl=64 time=22.8 ms
64 bytes from 10.0.0.3: icmp_seq=16 ttl=64 time=23.3 ms
64 bytes from 10.0.0.3: icmp_seq=17 ttl=64 time=23.1 ms
64 bytes from 10.0.0.3: icmp_seq=18 ttl=64 time=22.9 ms
64 bytes from 10.0.0.3: icmp_seq=19 ttl=64 time=22.5 ms
64 bytes from 10.0.0.3: icmp_seq=20 ttl=64 time=22.1 ms

--- 10.0.0.3 ping statistics ---
20 packets transmitted, 20 received, 0% packet loss, time 19030ms
rtt min/avg/max/mdev = 21.625/22.777/24.667/0.718 ms
root@why-VirtualBox:~/lab4# 
```

The average RTT between 10.0.0.111 and 10.0.0.3 is 11.525 ms.


MyVM [正在运行] - Oracle VM VirtualBox
管理 控制 视图 热键 设备 帮助
活动 XTerm
星期三 14：46
zh
"Node: s1" (root)

```
root@why-VirtualBox:~/lab4# ping -c 20 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=12.7 ms
64 bytes from 10.0.0.3: icmp_seq=2 ttl=64 time=12.0 ms
64 bytes from 10.0.0.3: icmp_seq=3 ttl=64 time=11.8 ms
64 bytes from 10.0.0.3: icmp_seq=4 ttl=64 time=11.1 ms
64 bytes from 10.0.0.3: icmp_seq=5 ttl=64 time=11.1 ms
64 bytes from 10.0.0.3: icmp_seq=6 ttl=64 time=11.7 ms
64 bytes from 10.0.0.3: icmp_seq=7 ttl=64 time=11.1 ms
64 bytes from 10.0.0.3: icmp_seq=8 ttl=64 time=11.3 ms
64 bytes from 10.0.0.3: icmp_seq=9 ttl=64 time=11.5 ms
64 bytes from 10.0.0.3: icmp_seq=10 ttl=64 time=10.8 ms
64 bytes from 10.0.0.3: icmp_seq=11 ttl=64 time=11.1 ms
64 bytes from 10.0.0.3: icmp_seq=12 ttl=64 time=11.1 ms
64 bytes from 10.0.0.3: icmp_seq=13 ttl=64 time=11.9 ms
64 bytes from 10.0.0.3: icmp_seq=14 ttl=64 time=11.2 ms
64 bytes from 10.0.0.3: icmp_seq=15 ttl=64 time=11.1 ms
64 bytes from 10.0.0.3: icmp_seq=16 ttl=64 time=11.9 ms
64 bytes from 10.0.0.3: icmp_seq=17 ttl=64 time=11.1 ms
64 bytes from 10.0.0.3: icmp_seq=18 ttl=64 time=11.9 ms
64 bytes from 10.0.0.3: icmp_seq=19 ttl=64 time=11.9 ms
64 bytes from 10.0.0.3: icmp_seq=20 ttl=64 time=11.4 ms

--- 10.0.0.3 ping statistics ---
20 packets transmitted, 20 received, 0% packet loss, time 19034ms
rtt min/avg/max/mdev = 10.877/11.525/12.763/0.481 ms
root@why-VirtualBox:~/lab4# 
```

We can find that

1. The average RTTs between 10.0.0.1/10.0.0.2 and 10.0.0.3 are almost the same (22 ms)
2. The average RTTs between 10.0.0.1/10.0.0.2 and 10.0.0.3 are larger than the average RTT between 10.0.0.111 and 10.0.0.3 (11 ms)
3. The average RTTs between 192.168.164.4 and 192.168.164.3 is smallest (0.487ms)

(1) is because the topological status of h1 and h2 are equivalent, so their latency are almost the same

(2) is because we set all the links time delay is 5 ms, and a round trip from h1 (or h2) to h3 passes links 4 times (4*5=20ms), a round trip from s1 to h3 passes links 2 times (2*5=10ms)

(3) 's reason is obvious, it shows that the real latency is very small.