1. 1. To add a new default route for vm1,vm2 respectively, we can add following code to step1:

   On vm1: sudo route add default gw 192.168.56.201

   On vm2: sudo route add default gw 192.168.56.201

   2. To Build the VXLAN tunnel and set the remote IP address as 192.168.56.102, add this to vm1 in step2:

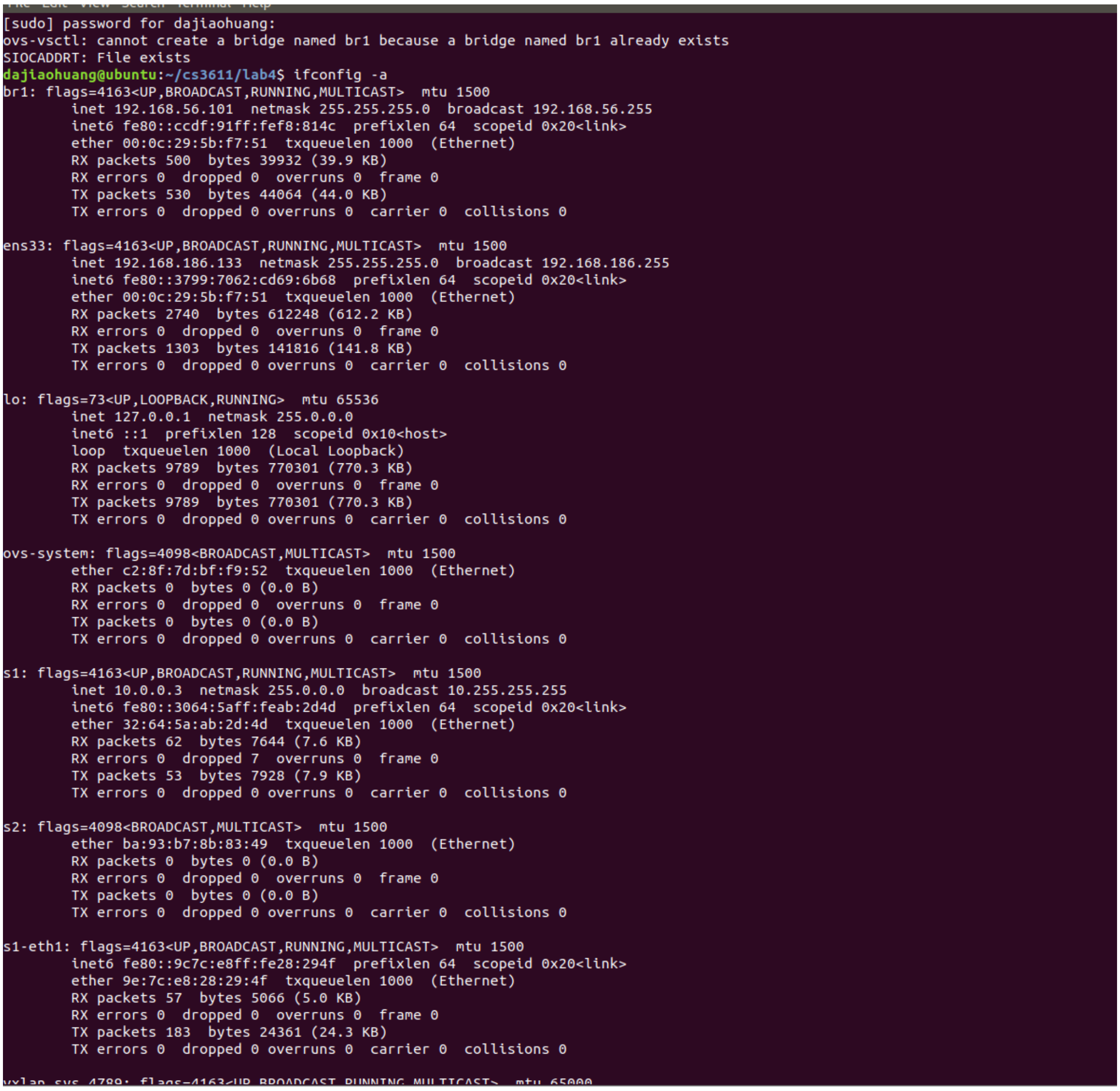
   sudo ovs-vsctl add-port s1 vxlan0 -- set interface vxlan0 type=vxlan options:remote_ip=192.168.56.102

   To Build the VXLAN tunnel and set the remote IP address as 192.168.56.101, add this to vm2 in step2:

   sudo ovs-vsctl add-port s2 vxlan0 -- set interface vxlan0 type=vxlan options:remote_ip=192.168.56.101

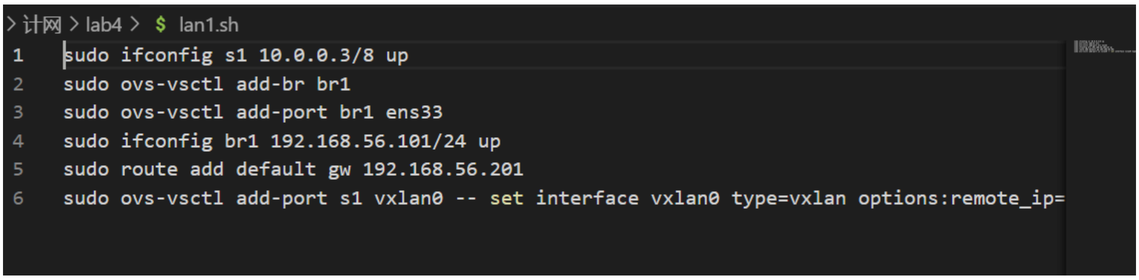
2. First check the config

   use: sudo ifconfig -a



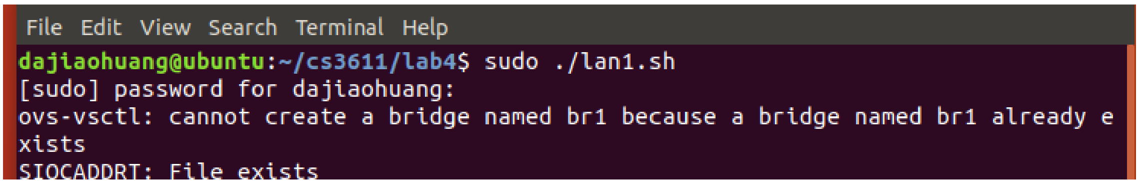
```
[sudo] password for dajiaohuang:
ovs-vsctl: cannot create a bridge named br1 because a bridge named br1 already exists
SIOCADDRT: File exists
dajiaohuang@ubuntu:~/cs3611/lab4$ ifconfig -a
br1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.56.101  netmask 255.255.255.0  broadcast 192.168.56.255
        inet6 fe80::ccdf:91ff:fef8:814c  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:5b:f7:51  txqueuelen 1000  (Ethernet)
        RX packets 500  bytes 39932 (39.9 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 530  bytes 44064 (44.0 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.186.133  netmask 255.255.255.0  broadcast 192.168.186.255
        inet6 fe80::3799:7062:cd69:6b68  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:5b:f7:51  txqueuelen 1000  (Ethernet)
        RX packets 2740  bytes 612248 (612.2 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 1303  bytes 141816 (141.8 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 9789  bytes 770301 (770.3 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 9789  bytes 770301 (770.3 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

ovs-system: flags=4098<BROADCAST,MULTICAST>  mtu 1500
        ether c2:8f:7d:bf:f9:52  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

s1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.0.3  netmask 255.0.0.0  broadcast 10.255.255.255
        inet6 fe80::3064:5aff:feab:2d4d  prefixlen 64  scopeid 0x20<link>
        ether 32:64:5a:ab:2d:4d  txqueuelen 1000  (Ethernet)
        RX packets 62  bytes 7644 (7.6 KB)
        RX errors 0  dropped 7  overruns 0  frame 0
        TX packets 53  bytes 7928 (7.9 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

s2: flags=4098<BROADCAST,MULTICAST>  mtu 1500
        ether ba:93:b7:8b:83:49  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

s1-eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet6 fe80::9c7c:e8ff:fe28:294f  prefixlen 64  scopeid 0x20<link>
        ether 9e:7c:e8:28:29:4f  txqueuelen 1000  (Ethernet)
        RX packets 57  bytes 5066 (5.0 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 183  bytes 24361 (24.3 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

vxlan_sys_4789: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 65000
```

   There's no enp0s8, so I modify "enp0s8" in the given code into "ens33", so that there won't be an error. And put those commands in lan1.sh and lan2.sh



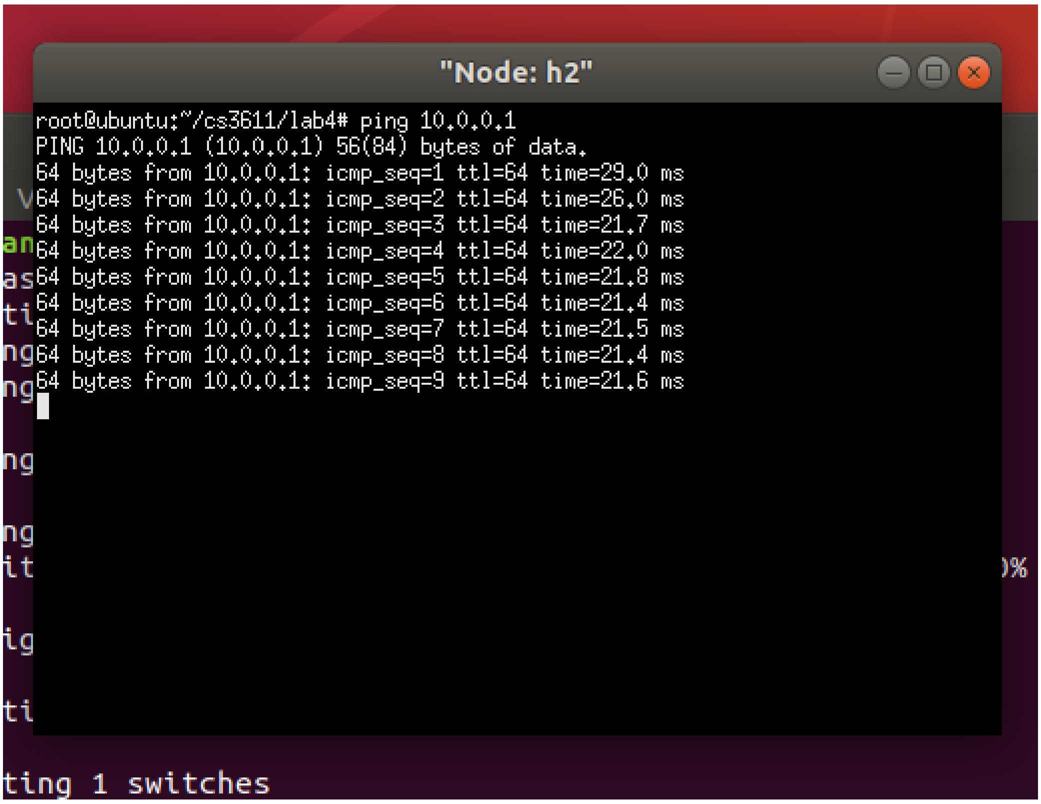
```
sudo ifconfig s1 10.0.0.3/8 up
sudo ovs-vsctl add-br br1
sudo ovs-vsctl add-port br1 ens33
sudo ifconfig br1 192.168.56.101/24 up
sudo route add default gw 192.168.56.201
sudo ovs-vsctl add-port s1 vxlan0 -- set interface vxlan0 type=vxlan options:remote_ip=
```

Run the sh files.



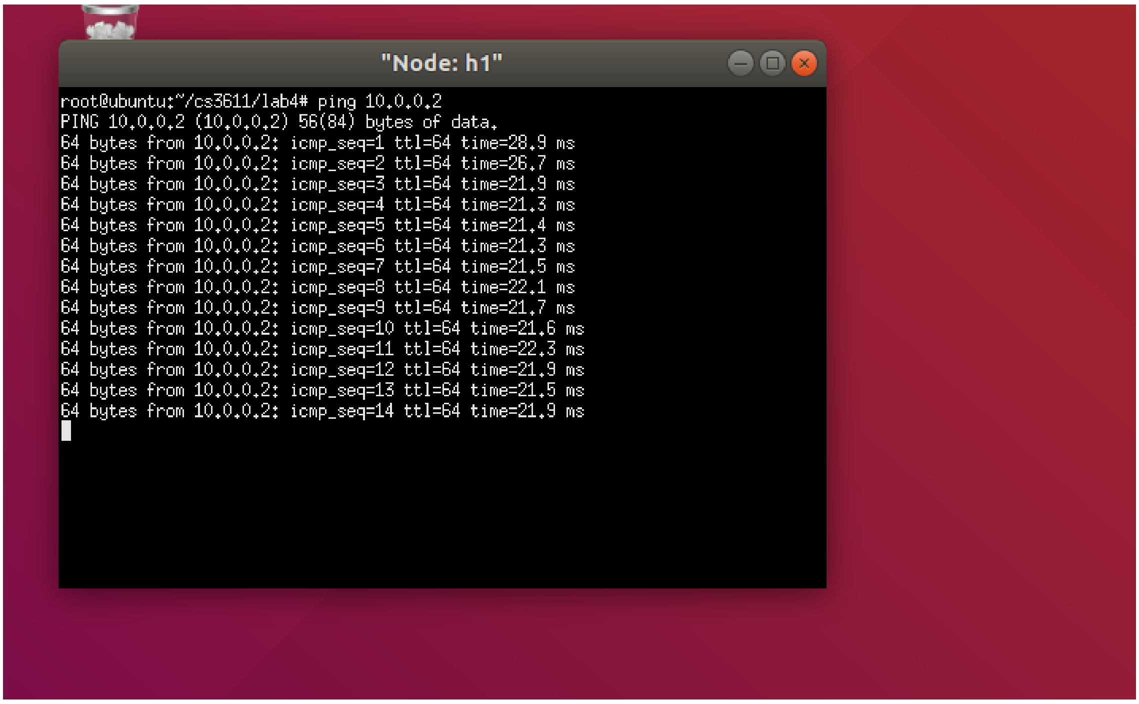
```
dajiaohuang@ubuntu:~/cs3611/lab4$ sudo ./lan1.sh
[sudo] password for dajiaohuang:
ovs-vsctl: cannot create a bridge named br1 because a bridge named br1 already e
xists
SIOCADDRT: File exists
```

Then use xterm and ping 10.0.0.1 from 10.0.0.2.



```
root@ubuntu:~/cs3611/lab4# 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=29.0 ms
64 bytes from 10.0.0.1: icmp_seq=2 ttl=64 time=26.0 ms
64 bytes from 10.0.0.1: icmp_seq=3 ttl=64 time=21.7 ms
64 bytes from 10.0.0.1: icmp_seq=4 ttl=64 time=22.0 ms
64 bytes from 10.0.0.1: icmp_seq=5 ttl=64 time=21.8 ms
64 bytes from 10.0.0.1: icmp_seq=6 ttl=64 time=21.4 ms
64 bytes from 10.0.0.1: icmp_seq=7 ttl=64 time=21.5 ms
64 bytes from 10.0.0.1: icmp_seq=8 ttl=64 time=21.4 ms
64 bytes from 10.0.0.1: icmp_seq=9 ttl=64 time=21.6 ms
```

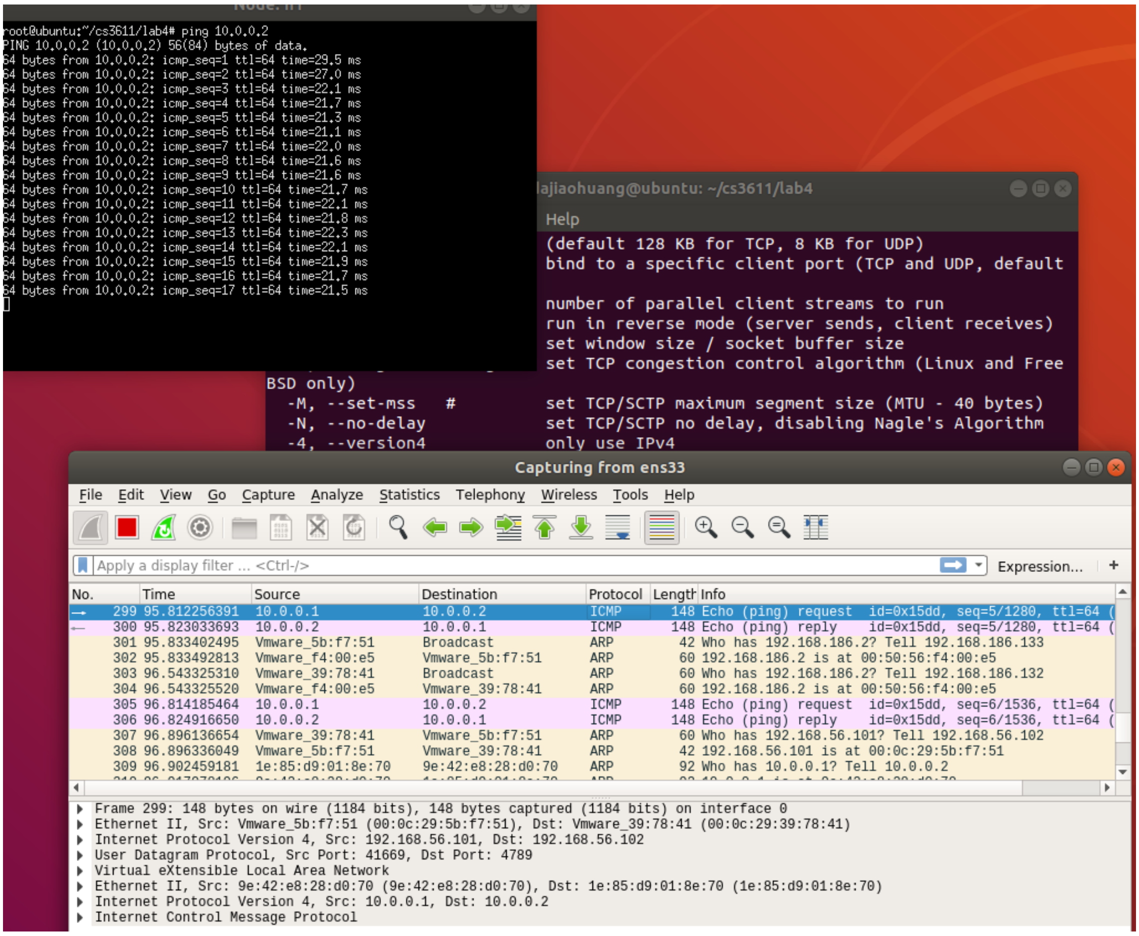
Also ping reversely.



```
root@ubuntu:~/cs3611/lab4# ping 10.0.0.2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=28.9 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=26.7 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=21.9 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=21.3 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=21.4 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=21.3 ms
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=21.5 ms
64 bytes from 10.0.0.2: icmp_seq=8 ttl=64 time=22.1 ms
64 bytes from 10.0.0.2: icmp_seq=9 ttl=64 time=21.7 ms
64 bytes from 10.0.0.2: icmp_seq=10 ttl=64 time=21.6 ms
64 bytes from 10.0.0.2: icmp_seq=11 ttl=64 time=22.3 ms
64 bytes from 10.0.0.2: icmp_seq=12 ttl=64 time=21.9 ms
64 bytes from 10.0.0.2: icmp_seq=13 ttl=64 time=21.5 ms
64 bytes from 10.0.0.2: icmp_seq=14 ttl=64 time=21.9 ms
```

Wireshark when ping 10.0.0.2 from 10.0.0.1:

ens33:



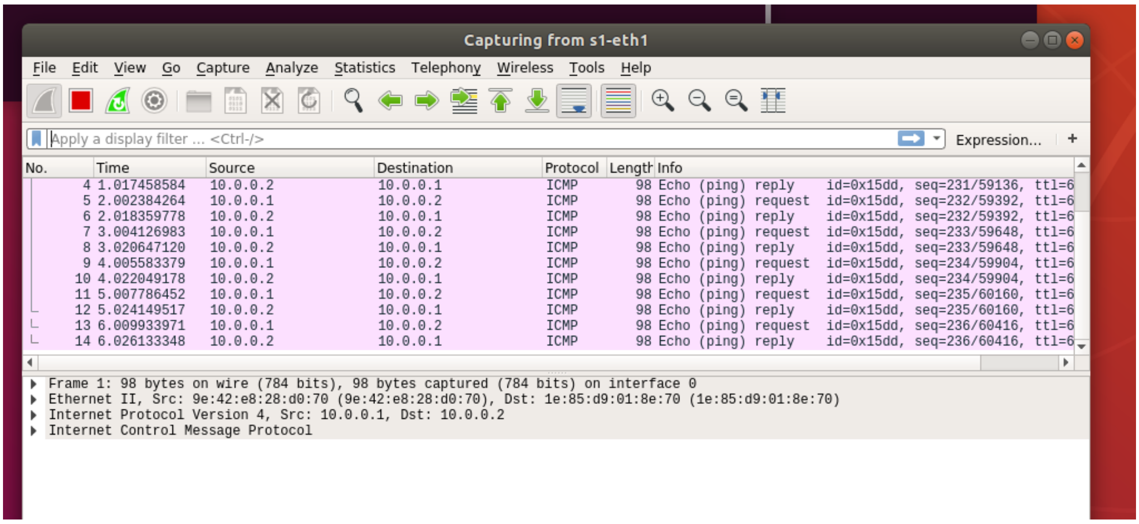
```
root@ubuntu:~/cs3611/lab4# ping 10.0.0.2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=29.5 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=27.0 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=22.1 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=21.7 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=21.3 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=21.1 ms
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=22.0 ms
64 bytes from 10.0.0.2: icmp_seq=8 ttl=64 time=21.6 ms
64 bytes from 10.0.0.2: icmp_seq=9 ttl=64 time=21.6 ms
64 bytes from 10.0.0.2: icmp_seq=10 ttl=64 time=21.7 ms
64 bytes from 10.0.0.2: icmp_seq=11 ttl=64 time=22.1 ms
64 bytes from 10.0.0.2: icmp_seq=12 ttl=64 time=21.8 ms
64 bytes from 10.0.0.2: icmp_seq=13 ttl=64 time=22.3 ms
64 bytes from 10.0.0.2: icmp_seq=14 ttl=64 time=22.1 ms
64 bytes from 10.0.0.2: icmp_seq=15 ttl=64 time=21.9 ms
64 bytes from 10.0.0.2: icmp_seq=16 ttl=64 time=21.7 ms
64 bytes from 10.0.0.2: icmp_seq=17 ttl=64 time=21.5 ms
```

```
                                 (default 128 KB for TCP, 8 KB for UDP)
                                 bind to a specific client port (TCP and UDP, default

                                 number of parallel client streams to run
                                 run in reverse mode (server sends, client receives)
                                 set window size / socket buffer size
                                 set TCP congestion control algorithm (Linux and Free
BSD only)
  -M, --set-mss    #             set TCP/SCTP maximum segment size (MTU - 40 bytes)
  -N, --no-delay                 set TCP/SCTP no delay, disabling Nagle's Algorithm
  -4, --version4                 only use IPv4
```

Capturing from ens33
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
Expression...
+

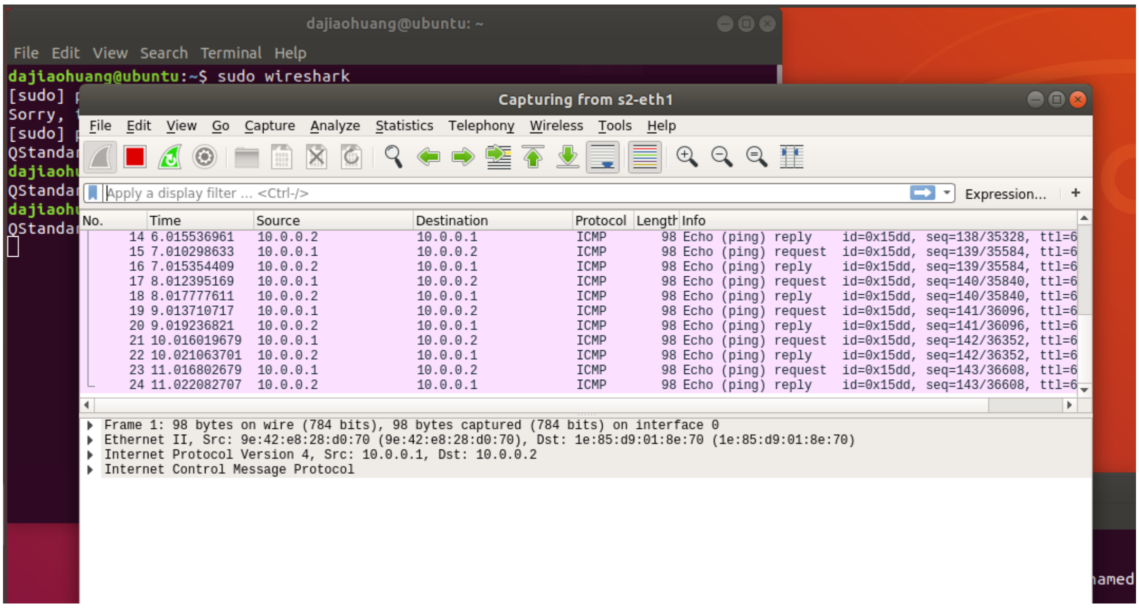
```
No.   Time         Source             Destination        Protocol Length Info
  299 95.812256391 10.0.0.1           10.0.0.2           ICMP     148 Echo (ping) request  id=0x15dd, seq=5/1280, ttl=64 (
  300 95.823033693 10.0.0.2           10.0.0.1           ICMP     148 Echo (ping) reply    id=0x15dd, seq=5/1280, ttl=64 (
  301 95.833402495 Vmware_5b:f7:51    Broadcast          ARP       42 Who has 192.168.186.2? Tell 192.168.186.133
  302 95.833492813 Vmware_f4:00:e5    Vmware_5b:f7:51    ARP       60 192.168.186.2 is at 00:50:56:f4:00:e5
  303 96.543325310 Vmware_39:78:41    Broadcast          ARP       60 Who has 192.168.186.2? Tell 192.168.186.132
  304 96.543325520 Vmware_f4:00:e5    Vmware_39:78:41    ARP       60 192.168.186.2 is at 00:50:56:f4:00:e5
  305 96.814185464 10.0.0.1           10.0.0.2           ICMP     148 Echo (ping) request  id=0x15dd, seq=6/1536, ttl=64 (
  306 96.824916650 10.0.0.2           10.0.0.1           ICMP     148 Echo (ping) reply    id=0x15dd, seq=6/1536, ttl=64 (
  307 96.896136654 Vmware_39:78:41    Vmware_5b:f7:51    ARP       60 Who has 192.168.56.101? Tell 192.168.56.102
  308 96.896336049 Vmware_5b:f7:51    Vmware_39:78:41    ARP       42 192.168.56.101 is at 00:0c:29:5b:f7:51
  309 96.902459181 1e:85:d9:01:8e:70  9e:42:e8:28:d0:70  ARP       92 Who has 10.0.0.1? Tell 10.0.0.2

▸ Frame 299: 148 bytes on wire (1184 bits), 148 bytes captured (1184 bits) on interface 0
▸ Ethernet II, Src: Vmware_5b:f7:51 (00:0c:29:5b:f7:51), Dst: Vmware_39:78:41 (00:0c:29:39:78:41)
▸ Internet Protocol Version 4, Src: 192.168.56.101, Dst: 192.168.56.102
▸ User Datagram Protocol, Src Port: 41669, Dst Port: 4789
▸ Virtual eXtensible Local Area Network
▸ Ethernet II, Src: 9e:42:e8:28:d0:70 (9e:42:e8:28:d0:70), Dst: 1e:85:d9:01:8e:70 (1e:85:d9:01:8e:70)
▸ Internet Protocol Version 4, Src: 10.0.0.1, Dst: 10.0.0.2
▸ Internet Control Message Protocol
```

s1-eth1:


Capturing from s1-eth1
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
Expression...

```
No.   Time          Source      Destination  Protocol Length Info
    4 1.017458584   10.0.0.2    10.0.0.1     ICMP     98 Echo (ping) reply    id=0x15dd, seq=231/59136, ttl=6
    5 2.002384264   10.0.0.1    10.0.0.2     ICMP     98 Echo (ping) request  id=0x15dd, seq=232/59392, ttl=6
    6 2.018359778   10.0.0.2    10.0.0.1     ICMP     98 Echo (ping) reply    id=0x15dd, seq=232/59392, ttl=6
    7 3.004126983   10.0.0.1    10.0.0.2     ICMP     98 Echo (ping) request  id=0x15dd, seq=233/59648, ttl=6
    8 3.020647120   10.0.0.2    10.0.0.1     ICMP     98 Echo (ping) reply    id=0x15dd, seq=233/59648, ttl=6
    9 4.005583379   10.0.0.1    10.0.0.2     ICMP     98 Echo (ping) request  id=0x15dd, seq=234/59904, ttl=6
   10 4.022049178   10.0.0.2    10.0.0.1     ICMP     98 Echo (ping) reply    id=0x15dd, seq=234/59904, ttl=6
   11 5.007786452   10.0.0.1    10.0.0.2     ICMP     98 Echo (ping) request  id=0x15dd, seq=235/60160, ttl=6
   12 5.024149517   10.0.0.2    10.0.0.1     ICMP     98 Echo (ping) reply    id=0x15dd, seq=235/60160, ttl=6
   13 6.009933971   10.0.0.1    10.0.0.2     ICMP     98 Echo (ping) request  id=0x15dd, seq=236/60416, ttl=6
   14 6.026133348   10.0.0.2    10.0.0.1     ICMP     98 Echo (ping) reply    id=0x15dd, seq=236/60416, ttl=6

▸ Frame 1: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface 0
▸ Ethernet II, Src: 9e:42:e8:28:d0:70 (9e:42:e8:28:d0:70), Dst: 1e:85:d9:01:8e:70 (1e:85:d9:01:8e:70)
▸ Internet Protocol Version 4, Src: 10.0.0.1, Dst: 10.0.0.2
▸ Internet Control Message Protocol
```

s2-eth1:

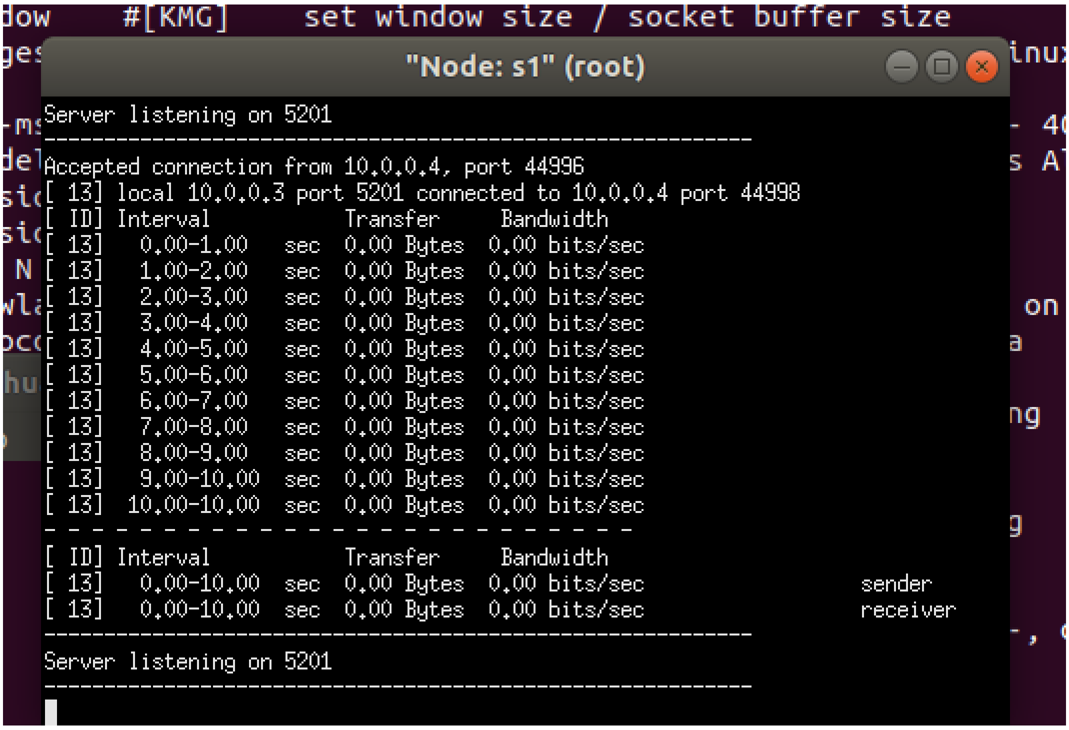

dajiaohuang@ubuntu: ~
File Edit View Search Terminal Help
Capturing from s2-eth1
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
Expression...

```
dajiaohuang@ubuntu:~$ sudo wireshark
[sudo] p
Sorry, 
[sudo] p
QStandar
dajiaohu
QStandar
dajiaohu
QStandar
```

```
No.   Time          Source      Destination  Protocol Length Info
   14 6.015536961   10.0.0.2    10.0.0.1     ICMP     98 Echo (ping) reply    id=0x15dd, seq=138/35328, ttl=6
   15 7.010298633   10.0.0.1    10.0.0.2     ICMP     98 Echo (ping) request  id=0x15dd, seq=139/35584, ttl=6
   16 7.015354409   10.0.0.2    10.0.0.1     ICMP     98 Echo (ping) reply    id=0x15dd, seq=139/35584, ttl=6
   17 8.012395169   10.0.0.1    10.0.0.2     ICMP     98 Echo (ping) request  id=0x15dd, seq=140/35840, ttl=6
   18 8.017777611   10.0.0.2    10.0.0.1     ICMP     98 Echo (ping) reply    id=0x15dd, seq=140/35840, ttl=6
   19 9.013710717   10.0.0.1    10.0.0.2     ICMP     98 Echo (ping) request  id=0x15dd, seq=141/36096, ttl=6
   20 9.019236821   10.0.0.2    10.0.0.1     ICMP     98 Echo (ping) reply    id=0x15dd, seq=141/36096, ttl=6
   21 10.016019679  10.0.0.1    10.0.0.2     ICMP     98 Echo (ping) request  id=0x15dd, seq=142/36352, ttl=6
   22 10.021063701  10.0.0.2    10.0.0.1     ICMP     98 Echo (ping) reply    id=0x15dd, seq=142/36352, ttl=6
   23 11.016802679  10.0.0.1    10.0.0.2     ICMP     98 Echo (ping) request  id=0x15dd, seq=143/36608, ttl=6
   24 11.022082707  10.0.0.2    10.0.0.1     ICMP     98 Echo (ping) reply    id=0x15dd, seq=143/36608, ttl=6

▸ Frame 1: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface 0
▸ Ethernet II, Src: 9e:42:e8:28:d0:70 (9e:42:e8:28:d0:70), Dst: 1e:85:d9:01:8e:70 (1e:85:d9:01:8e:70)
▸ Internet Protocol Version 4, Src: 10.0.0.1, Dst: 10.0.0.2
▸ Internet Control Message Protocol
```

We can see that ICMP,AMP are used.

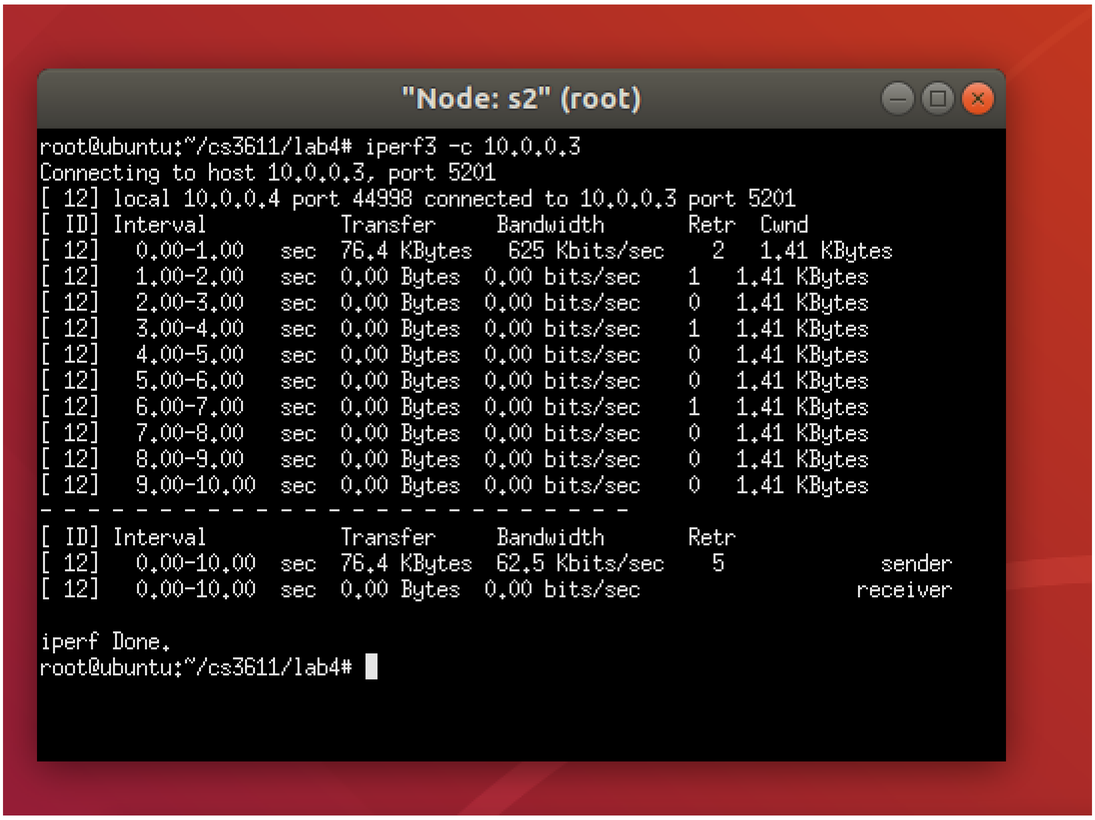
3. sudo iperf3 -s 10.0.0.4 on s1



```
dow     #[KMG]     set window size / socket buffer size
```

```
Server listening on 5201
-----------------------------------------------------------
Accepted connection from 10.0.0.4, port 44996
[ 13] local 10.0.0.3 port 5201 connected to 10.0.0.4 port 44998
[ ID] Interval           Transfer     Bandwidth
[ 13]   0.00-1.00   sec  0.00 Bytes  0.00 bits/sec
[ 13]   1.00-2.00   sec  0.00 Bytes  0.00 bits/sec
[ 13]   2.00-3.00   sec  0.00 Bytes  0.00 bits/sec
[ 13]   3.00-4.00   sec  0.00 Bytes  0.00 bits/sec
[ 13]   4.00-5.00   sec  0.00 Bytes  0.00 bits/sec
[ 13]   5.00-6.00   sec  0.00 Bytes  0.00 bits/sec
[ 13]   6.00-7.00   sec  0.00 Bytes  0.00 bits/sec
[ 13]   7.00-8.00   sec  0.00 Bytes  0.00 bits/sec
[ 13]   8.00-9.00   sec  0.00 Bytes  0.00 bits/sec
[ 13]   9.00-10.00  sec  0.00 Bytes  0.00 bits/sec
[ 13]  10.00-10.00  sec  0.00 Bytes  0.00 bits/sec
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth
[ 13]   0.00-10.00  sec  0.00 Bytes  0.00 bits/sec                  sender
[ 13]   0.00-10.00  sec  0.00 Bytes  0.00 bits/sec                  receiver
-----------------------------------------------------------
Server listening on 5201
-----------------------------------------------------------
```

sudo iperf3 -c 10.0.0.3 on s2

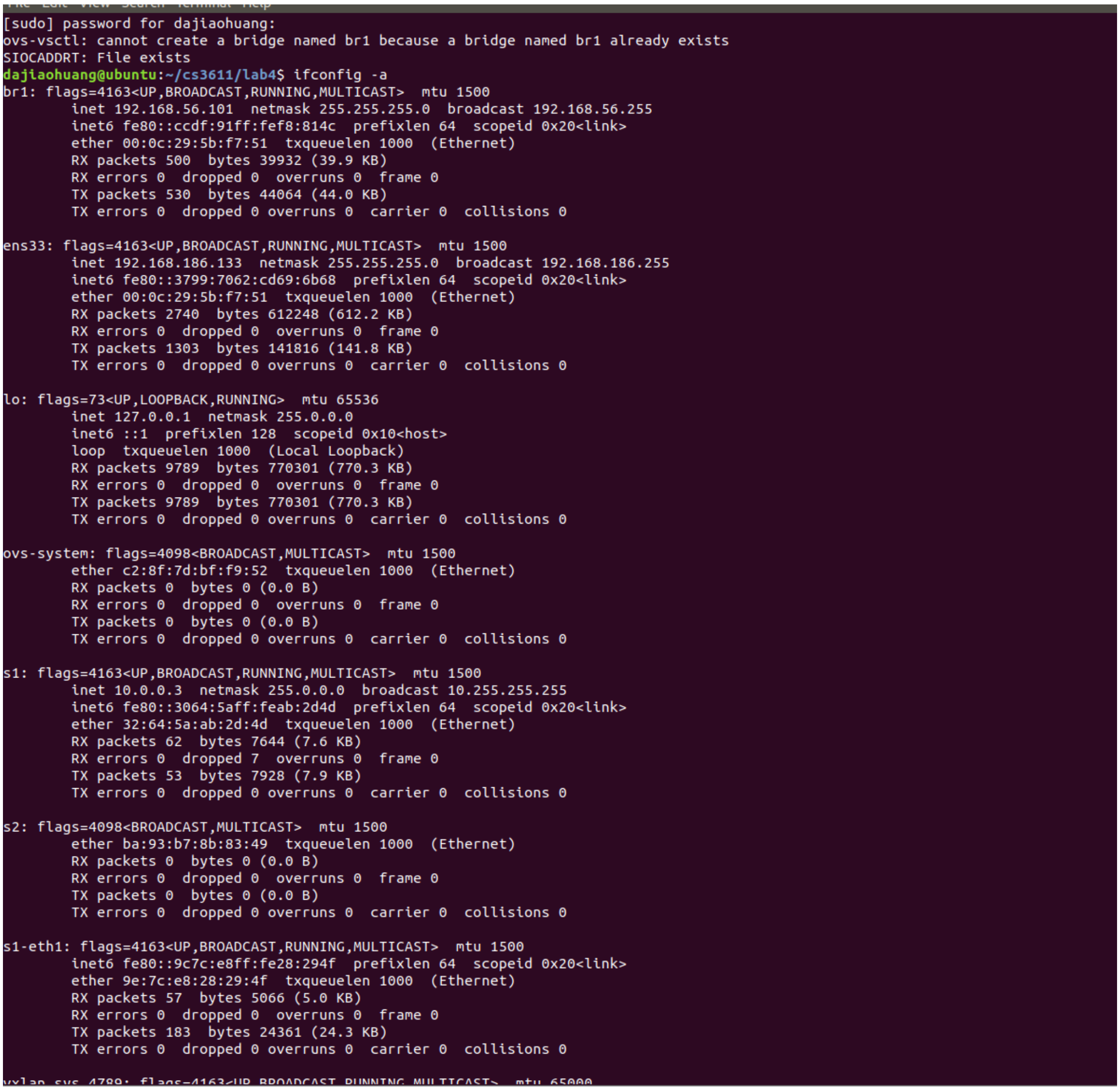


```
root@ubuntu:~/cs3611/lab4# iperf3 -c 10.0.0.3
Connecting to host 10.0.0.3, port 5201
[ 12] local 10.0.0.4 port 44998 connected to 10.0.0.3 port 5201
[ ID] Interval           Transfer     Bandwidth       Retr  Cwnd
[ 12]   0.00-1.00   sec  76.4 KBytes   625 Kbits/sec    2   1.41 KBytes
[ 12]   1.00-2.00   sec  0.00 Bytes  0.00 bits/sec    1   1.41 KBytes
[ 12]   2.00-3.00   sec  0.00 Bytes  0.00 bits/sec    0   1.41 KBytes
[ 12]   3.00-4.00   sec  0.00 Bytes  0.00 bits/sec    1   1.41 KBytes
[ 12]   4.00-5.00   sec  0.00 Bytes  0.00 bits/sec    0   1.41 KBytes
[ 12]   5.00-6.00   sec  0.00 Bytes  0.00 bits/sec    0   1.41 KBytes
[ 12]   6.00-7.00   sec  0.00 Bytes  0.00 bits/sec    1   1.41 KBytes
[ 12]   7.00-8.00   sec  0.00 Bytes  0.00 bits/sec    0   1.41 KBytes
[ 12]   8.00-9.00   sec  0.00 Bytes  0.00 bits/sec    0   1.41 KBytes
[ 12]   9.00-10.00  sec  0.00 Bytes  0.00 bits/sec    0   1.41 KBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth       Retr
[ 12]   0.00-10.00  sec  76.4 KBytes  62.5 Kbits/sec    5             sender
[ 12]   0.00-10.00  sec  0.00 Bytes  0.00 bits/sec                  receiver

iperf Done.
root@ubuntu:~/cs3611/lab4# 
```

Bandwidth is too small and there's no packet received.

This because of the MTU.

From this figure we can get MTU of br1.



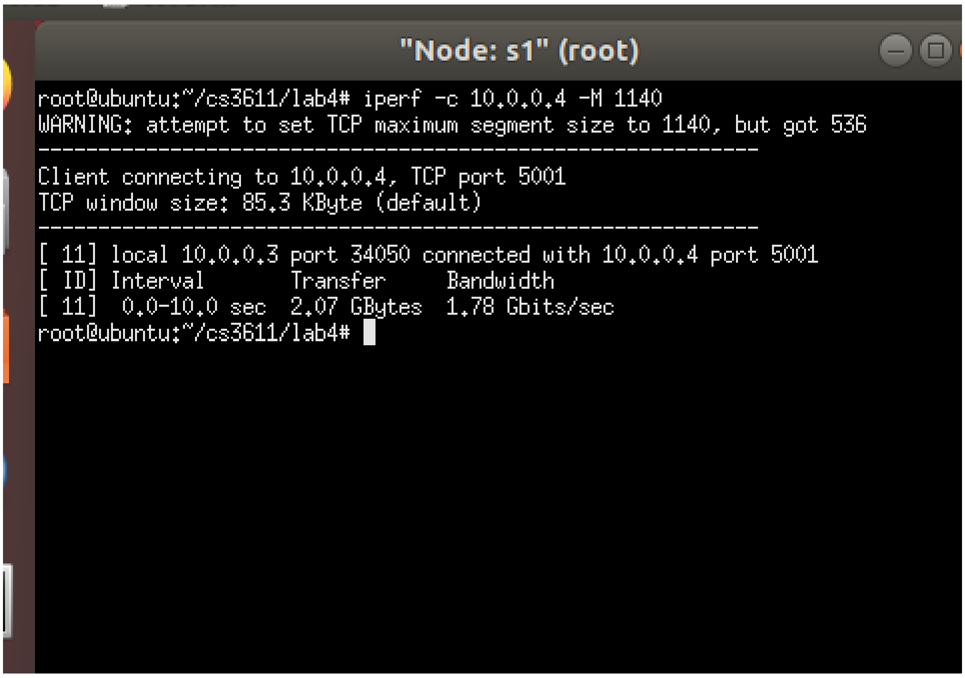
```
[sudo] password for dajiaohuang:
ovs-vsctl: cannot create a bridge named br1 because a bridge named br1 already exists
SIOCADDRT: File exists
dajiaohuang@ubuntu:~/cs3611/lab4$ ifconfig -a
br1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.56.101  netmask 255.255.255.0  broadcast 192.168.56.255
        inet6 fe80::ccdf:91ff:fef8:814c  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:5b:f7:51  txqueuelen 1000  (Ethernet)
        RX packets 500  bytes 39932 (39.9 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 530  bytes 44064 (44.0 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.186.133  netmask 255.255.255.0  broadcast 192.168.186.255
        inet6 fe80::3799:7062:cd69:6b68  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:5b:f7:51  txqueuelen 1000  (Ethernet)
        RX packets 2740  bytes 612248 (612.2 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 1303  bytes 141816 (141.8 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 9789  bytes 770301 (770.3 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 9789  bytes 770301 (770.3 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

ovs-system: flags=4098<BROADCAST,MULTICAST>  mtu 1500
        ether c2:8f:7d:bf:f9:52  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

s1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.0.3  netmask 255.0.0.0  broadcast 10.255.255.255
        inet6 fe80::3064:5aff:feab:2d4d  prefixlen 64  scopeid 0x20<link>
        ether 32:64:5a:ab:2d:4d  txqueuelen 1000  (Ethernet)
        RX packets 62  bytes 7644 (7.6 KB)
        RX errors 0  dropped 7  overruns 0  frame 0
        TX packets 53  bytes 7928 (7.9 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

s2: flags=4098<BROADCAST,MULTICAST>  mtu 1500
        ether ba:93:b7:8b:83:49  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

s1-eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet6 fe80::9c7c:e8ff:fe28:294f  prefixlen 64  scopeid 0x20<link>
        ether 9e:7c:e8:28:29:4f  txqueuelen 1000  (Ethernet)
        RX packets 57  bytes 5066 (5.0 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 183  bytes 24361 (24.3 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

vxlan_sys_4789: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 65000
```

The MTU of br1 is 1500.

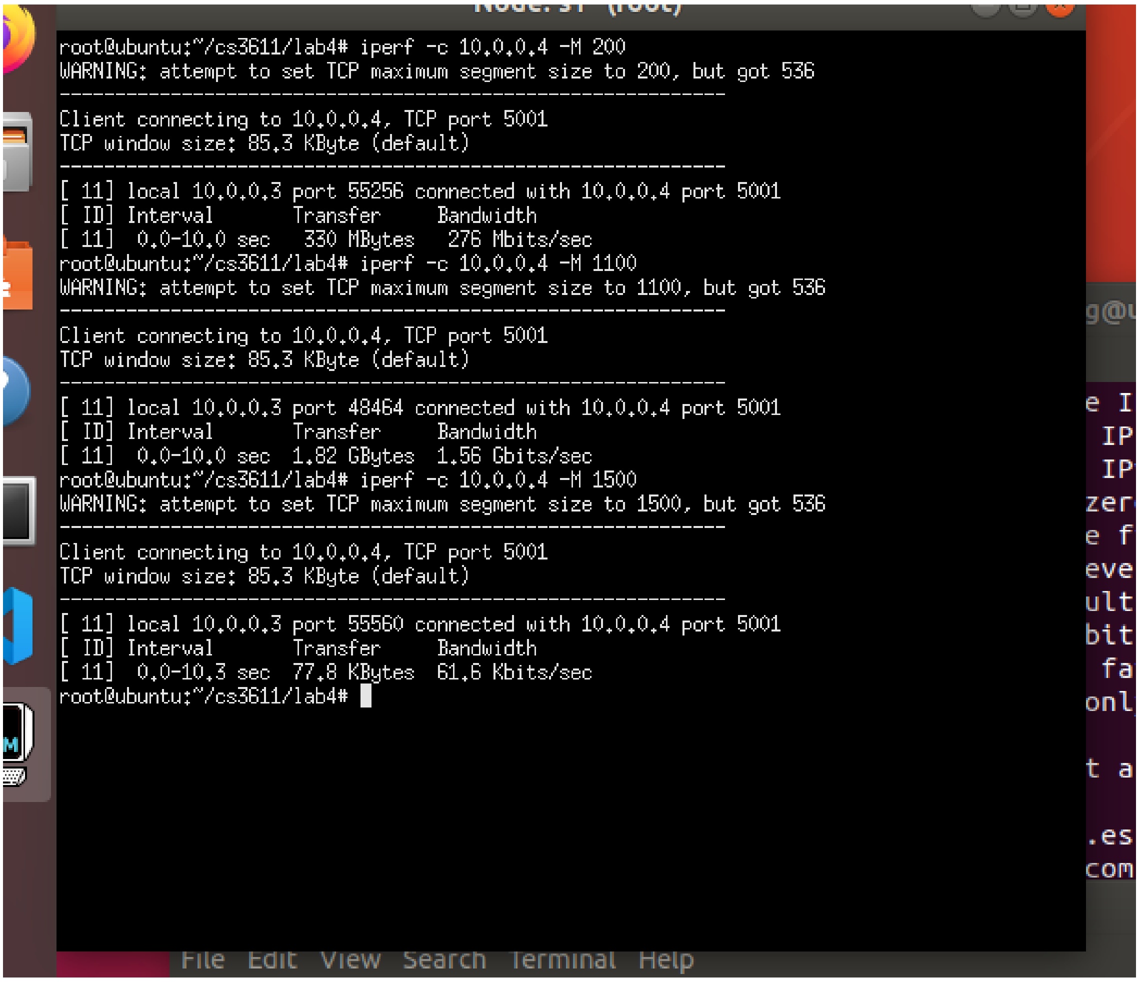
Since ping is an application layer command, we know that a datagram length is consist of 20 bytes datagram header, 20 bytes TCP segment header and application message. And we need 50 bytes for VXLAN in this lab. So the MSS of application message is: 1500−20−20−50 = 1410.

And local mtu is 65536, which is much bigger. So we waste much time packing and unpacking and in congestion.

4. Designating the MTU size of iperf3:



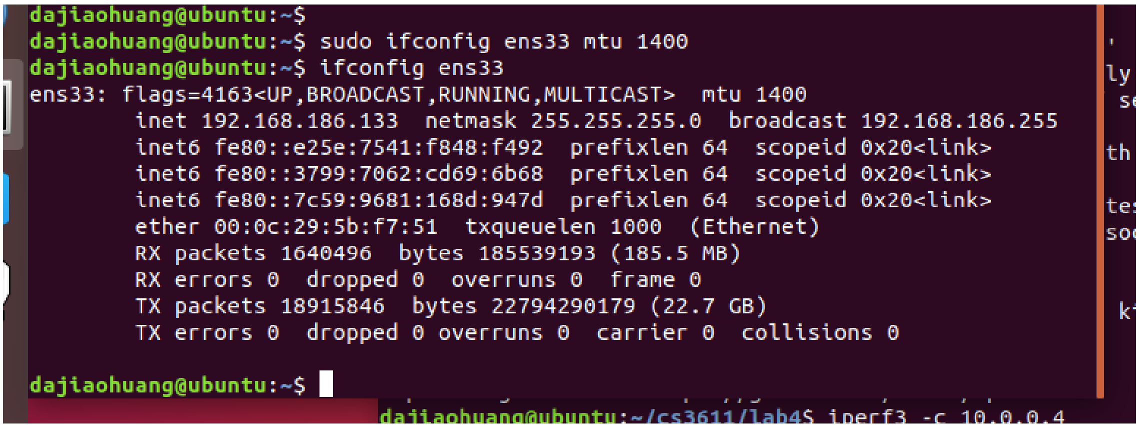
```
root@ubuntu:~/cs3611/lab4# iperf -c 10.0.0.4 -M 1140
WARNING: attempt to set TCP maximum segment size to 1140, but got 536
------------------------------------------------------------
Client connecting to 10.0.0.4, TCP port 5001
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
[ 11] local 10.0.0.3 port 34050 connected with 10.0.0.4 port 5001
[ ID] Interval       Transfer     Bandwidth
[ 11]  0.0-10.0 sec  2.07 GBytes  1.78 Gbits/sec
root@ubuntu:~/cs3611/lab4#
```



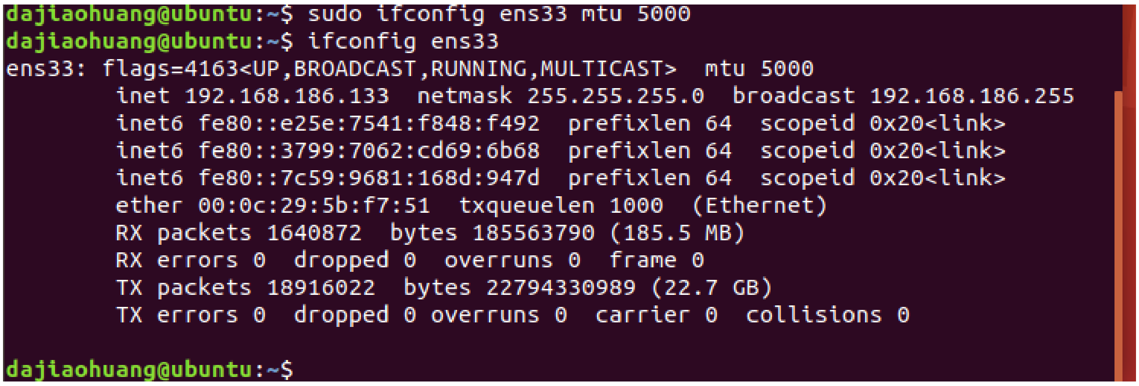
```
root@ubuntu:~/cs3611/lab4# iperf -c 10.0.0.4 -M 200
WARNING: attempt to set TCP maximum segment size to 200, but got 536
------------------------------------------------------------
Client connecting to 10.0.0.4, TCP port 5001
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
[ 11] local 10.0.0.3 port 55256 connected with 10.0.0.4 port 5001
[ ID] Interval       Transfer     Bandwidth
[ 11]  0.0-10.0 sec   330 MBytes   276 Mbits/sec
root@ubuntu:~/cs3611/lab4# iperf -c 10.0.0.4 -M 1100
WARNING: attempt to set TCP maximum segment size to 1100, but got 536
------------------------------------------------------------
Client connecting to 10.0.0.4, TCP port 5001
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
[ 11] local 10.0.0.3 port 48464 connected with 10.0.0.4 port 5001
[ ID] Interval       Transfer     Bandwidth
[ 11]  0.0-10.0 sec  1.82 GBytes  1.56 Gbits/sec
root@ubuntu:~/cs3611/lab4# iperf -c 10.0.0.4 -M 1500
WARNING: attempt to set TCP maximum segment size to 1500, but got 536
------------------------------------------------------------
Client connecting to 10.0.0.4, TCP port 5001
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
[ 11] local 10.0.0.3 port 55560 connected with 10.0.0.4 port 5001
[ ID] Interval       Transfer     Bandwidth
[ 11]  0.0-10.3 sec  77.8 KBytes  61.6 Kbits/sec
root@ubuntu:~/cs3611/lab4#
```

Bandwidth is improved. Because making the MTU of iperf3 smaller than that of br1 and ens33 (1500) avoids congestion, packing and unpacking.

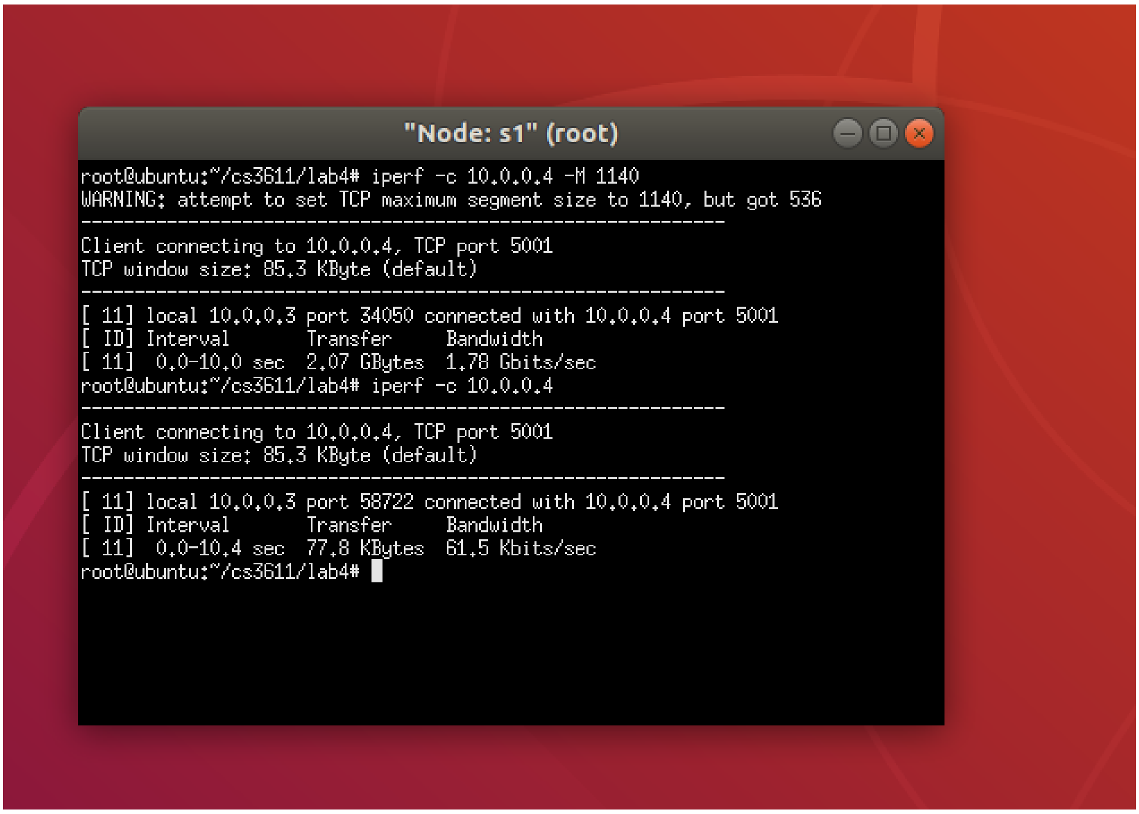
Designating the MTU of the two ens33 interfaces:



```
dajiaohuang@ubuntu:~$
dajiaohuang@ubuntu:~$ sudo ifconfig ens33 mtu 1400
dajiaohuang@ubuntu:~$ ifconfig ens33
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1400
        inet 192.168.186.133  netmask 255.255.255.0  broadcast 192.168.186.255
        inet6 fe80::e25e:7541:f848:f492  prefixlen 64  scopeid 0x20<link>
        inet6 fe80::3799:7062:cd69:6b68  prefixlen 64  scopeid 0x20<link>
        inet6 fe80::7c59:9681:168d:947d  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:5b:f7:51  txqueuelen 1000  (Ethernet)
        RX packets 1640496  bytes 185539193 (185.5 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 18915846  bytes 22794290179 (22.7 GB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

dajiaohuang@ubuntu:~$
                              dajiaohuang@ubuntu:~/cs3611/lab4$ iperf3 -c 10.0.0.4
```



```
dajiaohuang@ubuntu:~$ sudo ifconfig ens33 mtu 5000
dajiaohuang@ubuntu:~$ ifconfig ens33
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 5000
        inet 192.168.186.133  netmask 255.255.255.0  broadcast 192.168.186.255
        inet6 fe80::e25e:7541:f848:f492  prefixlen 64  scopeid 0x20<link>
        inet6 fe80::3799:7062:cd69:6b68  prefixlen 64  scopeid 0x20<link>
        inet6 fe80::7c59:9681:168d:947d  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:5b:f7:51  txqueuelen 1000  (Ethernet)
        RX packets 1640872  bytes 185563790 (185.5 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 18916022  bytes 22794330989 (22.7 GB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

dajiaohuang@ubuntu:~$
```



"Node: s1" (root)

```
root@ubuntu:~/cs3611/lab4# iperf -c 10.0.0.4 -M 1140
WARNING: attempt to set TCP maximum segment size to 1140, but got 536
------------------------------------------------------------
Client connecting to 10.0.0.4, TCP port 5001
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
[ 11] local 10.0.0.3 port 34050 connected with 10.0.0.4 port 5001
[ ID] Interval       Transfer     Bandwidth
[ 11]  0.0-10.0 sec  2.07 GBytes  1.78 Gbits/sec
root@ubuntu:~/cs3611/lab4# iperf -c 10.0.0.4
------------------------------------------------------------
Client connecting to 10.0.0.4, TCP port 5001
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
[ 11] local 10.0.0.3 port 58722 connected with 10.0.0.4 port 5001
[ ID] Interval       Transfer     Bandwidth
[ 11]  0.0-10.4 sec  77.8 KBytes  61.5 Kbits/sec
root@ubuntu:~/cs3611/lab4#
```

Not improved. Because we can never modify this MTU to make it bigger than the local MTU to avoid packing and unpacking. Also MTU of bridge is not changed, so actually there's no real difference.