LAPORAN

TUGAS BESAR

Mata Kuliah: Jaringan Komputer

Oleh:

MUHAMMAD RAFI ANDEO PRAJA (1301200278)

KELAS IF-44-01
JURUSAN S1 INFORMATIKA
FAKULTAS INFORMATIKA
UNIVERSITAS TELKOM

DAFTAR ISI

BAB I

PENDAHULUAN

A. Deskripsi Tugas Besar

Tugas Besar Jaringan Komputer dibagi menjadi 4 CLO yang masing - masing memiliki tujuan atau goalnya masing - masing. CLO 1, mahasiswa diharapkan untuk dapat membuat topologi jaringan yang sesuai dengan soal diberikan, serta topologi tersebut harus bisa dijalankan dengan melakukan subnetting dan assign IP sesuai subnet. CLO 1 juga diharapkan dapat mengecek konektivitas antar host dengan router menggunakan ping. CLO 2, mahasiswa diharapkan untuk dapat melakukan mekanisme routing pada topologi yang sudah dibuat dan menganalisis routing yang digunakan menggunakan traceroute. CLO 3, mahasiswa diharapkan menggunakan iperf sebagai generate traffic dan capture trafic menggunakan wireshark. CLO 4, mahasiswa dapat mengatur buffer dan capture pengaruh buffer terhadap delay.

B. Rumusan Masalah

Mahasiswa diharapkan dapat menyelesaikan seluruh komponen CLO tersebut dengan baik menggunakan aplikasi yang diminta. Tugas besar kali ini mahasiswa akan menggunakan mininet dan wireshark sebagai aplikasi untuk menuntaskan tugas besar. Inti dari keempat CLO tersebut adalah bagaimana mahasiswa harus bisa membuat sebuah jaringan menggunakan mininet dengan melakukan subnetting terlebih dahulu.

C. Tujuan

Tujuan dari tugas besar ini yaitu mahasiswa diharapkan dapat memahami bagaimana cara menggunakan aplikasi mininet dan wireshark dalam membuat dan menganalisis jaringan komputer yang dibuat. Mahasiswa juga diharapkan agar dapat menguasai bahasa python dan penggunaan sistem operasi linux sebagai tugas opsional dari tugas besar ini.

BAB II

PEMBAHASAN

A. CLO 1 (Konfigurasi Jaringan)

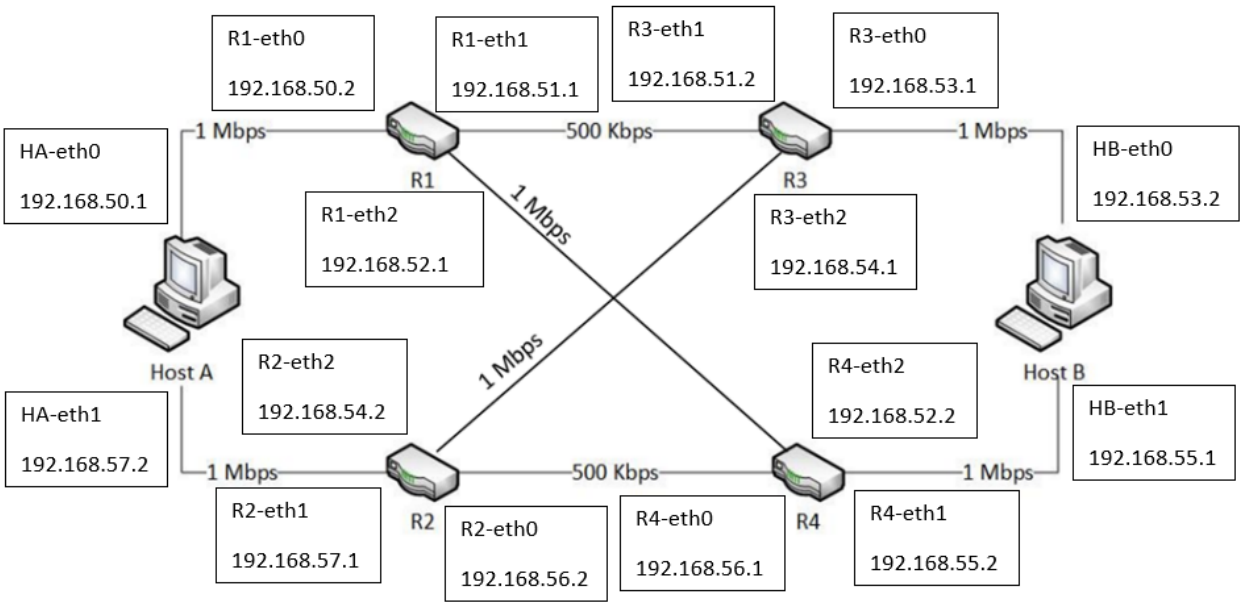
CLO 1 memiliki tujuan yang dimana membuat mahasiswa harus bisa membuat topologi jaringan soal dan melakukan konfigurasi jaringan berupa memasang IP address dan melakukan subnetting. Berikut adalah hasil dari CLO 1 yang saya kerjakan.

Tabel Subnetting:

IP yang dipakai	192.168.50.0

Nama	Needs	Alokasi	Network ID	Host Range	Broadcast	Prefix	Subnet Mask
Net 1	2	254	192.168.50.0	192.168.50.1 - 192.168.50.254	192.168.50.255	32 - 8 = /24	255.255.255.0
Net 2	2	254	192.168.51.0	192.168.51.1 - 192.168.51.254	192.168.51.255	32 - 8 = /24	255.255.255.0
Net 3	2	254	192.168.52.0	192.168.52.1 - 192.168.52.254	192.168.52.255	32 - 8 = /24	255.255.255.0
Net 4	2	254	192.168.53.0	192.168.53.1 - 192.168.53.254	192.168.53.255	32 - 8 = /24	255.255.255.0
Net 5	2	254	192.168.54.0	192.168.54.1 - 192.168.54.254	192.168.54.255	32 - 8 = /24	255.255.255.0
Net 6	2	254	192.168.55.0	192.168.55.1 - 192.168.55.254	192.168.55.255	32 - 8 = /24	255.255.255.0
Net 7	2	254	192.168.56.0	192.168.56.1 - 192.168.56.254	192.168.56.255	32 - 8 = /24	255.255.255.0
Net 8	2	254	192.168.57.0	192.168.57.1 - 192.168.57.254	192.168.57.255	32 - 8 = /24	255.255.255.0

Topologi yang sudah di pasang IP address:



Konfigurasi Topologi Jaringan:

```
mininet> net
r1 r1-eth0:ha-eth0 r1-eth1:r3-eth1 r1-eth2:r4-eth2
r2 r2-eth1:ha-eth1 r2-eth0:r4-eth0 r2-eth2:r3-eth2
r3 r3-eth0:hb-eth0 r3-eth1:r1-eth1 r3-eth2:r2-eth2
r4 r4-eth1:hb-eth1 r4-eth2:r1-eth2 r4-eth0:r2-eth0
ha ha-eth0:r1-eth0 ha-eth1:r2-eth1
hb hb-eth0:r3-eth0 hb-eth1:r4-eth1
mininet>
```

Uji Konektivitas:

1. Dalam satu area

```
mininet> ha ping r1
PING 192.168.50.2 (192.168.50.2) 56(84) bytes of data.
64 bytes from 192.168.50.2: icmp_seq=1 ttl=64 time=0.043 ms
^C
--- 192.168.50.2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.043/0.043/0.043/0.000 ms
mininet> ha ping r2
PING 192.168.57.1 (192.168.57.1) 56(84) bytes of data.
64 bytes from 192.168.57.1: icmp_seq=1 ttl=64 time=0.061 ms
^C
--- 192.168.57.1 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.061/0.061/0.061/0.000 ms
mininet> hb ping r3
PING 192.168.53.1 (192.168.53.1) 56(84) bytes of data.
64 bytes from 192.168.53.1: icmp_seq=1 ttl=64 time=0.072 ms
^C
--- 192.168.53.1 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.072/0.072/0.072/0.000 ms
mininet> hb ping r4
PING 192.168.55.2 (192.168.55.2) 56(84) bytes of data.
64 bytes from 192.168.55.2: icmp_seq=1 ttl=64 time=0.058 ms
^C
--- 192.168.55.2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.058/0.058/0.058/0.000 ms
mininet>
```

2. Berbeda area

```
mininet> ha ping r3
ping: connect: Network is unreachable
mininet> ha ping r4
ping: connect: Network is unreachable
mininet> ha ping hb
ping: connect: Network is unreachable
mininet> hb ping r1
ping: connect: Network is unreachable
mininet> hb ping r2
ping: connect: Network is unreachable
mininet> hb ping ha
ping: connect: Network is unreachable
mininet>
```

Program:

```
r2 = net.addHost('r2')
r3 = net.addHost('r3')
r4 = net.addHost('r4')

# Add Host
ha = net.addHost('ha')
hb = net.addHost('hb')

# Bandwidth
bw1={'bw':1}
bw2={'bw':0.5}

# Add Link
net.addLink(r1, ha, intfName1 = 'r1-eth0', intfName2 = 'ha-eth0', cls=TCLink, **bw1)
net.addLink(r2, ha, intfName1 = 'r2-eth1', intfName2 = 'ha-eth1', cls=TCLink, **bw1)
net.addLink(r3, hb, intfName1 = 'r3-eth0', intfName2 = 'hb-eth0', cls=TCLink, **bw1)
net.addLink(r4, hb, intfName1 = 'r4-eth1', intfName2 = 'hb-eth1', cls=TCLink, **bw1)
net.addLink(r1, r3, intfName1 = 'r1-eth1', intfName2 = 'r3-eth1', cls=TCLink, **bw2)
net.addLink(r1, r4, intfName1 = 'r1-eth2', intfName2 = 'r4-eth2', cls=TCLink, **bw1)
net.addLink(r2, r4, intfName1 = 'r2-eth0', intfName2 = 'r4-eth0', cls=TCLink, **bw2)
net.addLink(r2, r3, intfName1 = 'r2-eth2', intfName2 = 'r3-eth2', cls=TCLink, **bw1)
net.build()

#Host Configuration
ha.cmd("ifconfig ha-eth0 0")
ha.cmd("ifconfig ha-eth1 0")
ha.cmd("ifconfig ha-eth0 192.168.50.1 netmask 255.255.255.0")
ha.cmd("ifconfig ha-eth1 192.168.57.2 netmask 255.255.255.0")

hb.cmd("ifconfig hb-eth0 0")
hb.cmd("ifconfig hb-eth1 0")
hb.cmd("ifconfig hb-eth0 192.168.53.2 netmask 255.255.255.0")
hb.cmd("ifconfig hb-eth1 192.168.55.1 netmask 255.255.255.0")

#Router Configuration
r1.cmd("echo 1 > /proc/sys/net/ipv4/ip_forward")
r2.cmd("echo 2 > /proc/sys/net/ipv4/ip_forward")
r3.cmd("echo 3 > /proc/sys/net/ipv4/ip_forward")
r4.cmd("echo 4 > /proc/sys/net/ipv4/ip_forward")

r1.cmd("ifconfig r1-eth0 0")
r1.cmd("ifconfig r1-eth1 0")
r1.cmd("ifconfig r1-eth2 0")
r1.cmd("ifconfig r1-eth0 192.168.50.2 netmask 255.255.255.0")
r1.cmd("ifconfig r1-eth1 192.168.51.1 netmask 255.255.255.0")
r1.cmd("ifconfig r1-eth2 192.168.52.1 netmask 255.255.255.0")
```

B. CLO 2 (Mekanisme Routing)

CLO 2 memiliki tujuan yang dimana membuat mahasiswa memahami bagaimana cara melakukan mekanisme routing terhadap topologi yang sudah dibuat. Berikut adalah hasil dari CLO 2 yang saya kerjakan.

Static Routing Host dan Router:

```
ha.cmd("ip rule add from 192.168.50.1 table 1")
ha.cmd("ip rule add from 192.168.57.2 table 2")
ha.cmd("ip rule add 192.168.50.0/24 dev ha-eth0 scope link table 1")
ha.cmd("ip rule add default via 192.168.50.2 dev ha-eth0 table 1")
ha.cmd("ip rule add 192.168.57.0/24 dev ha-eth1 scope link table 2")
ha.cmd("ip rule add default via 192.168.57.1 dev ha-eth1 table 2")
ha.cmd("ip rule add default scope global nexthop via 192.168.50.2 dev ha-eth0")
ha.cmd("ip rule add default scope global nexthop via 192.168.57.1 dev ha-eth1")
ha.cmd("route add default gw 192.168.50.2 dev ha-eth0")
ha.cmd("route add default gw 192.168.57.1 dev ha-eth1")

hb.cmd("ip rule add from 192.168.53.2 table 3")
hb.cmd("ip rule add from 192.168.55.1 table 4")
hb.cmd("ip rule add 192.168.53.0/24 dev hb-eth0 scope link table 1")
hb.cmd("ip rule add default via 192.168.53.1 dev hb-eth0 table 1")
hb.cmd("ip rule add 192.168.55.0/24 dev hb-eth1 scope link table 2")
hb.cmd("ip rule add default via 192.168.55.2 dev hb-eth1 table 2")
hb.cmd("ip rule add default scope global nexthop via 192.168.53.1 dev hb-eth0")
hb.cmd("ip rule add default scope global nexthop via 192.168.55.2 dev hb-eth1")
hb.cmd("route add default gw 192.168.53.1 dev hb-eth0")
hb.cmd("route add default gw 192.168.55.2 dev hb-eth1")

#Static Routing Router
r1.cmd("route add -net 192.168.53.0/24 gw 192.168.51.2")
r1.cmd("route add -net 192.168.54.0/24 gw 192.168.51.2")
r1.cmd("route add -net 192.168.55.0/24 gw 192.168.52.2")
r1.cmd("route add -net 192.168.56.0/24 gw 192.168.52.2")
r1.cmd("route add -net 192.168.57.0/24 gw 192.168.51.2")

r2.cmd("route add -net 192.168.50.0/24 gw 192.168.54.1")
r2.cmd("route add -net 192.168.52.0/24 gw 192.168.56.1")
r2.cmd("route add -net 192.168.55.0/24 gw 192.168.56.1")
r2.cmd("route add -net 192.168.51.0/24 gw 192.168.54.1")
r2.cmd("route add -net 192.168.53.0/24 gw 192.168.54.1")

r3.cmd("route add -net 192.168.50.0/24 gw 192.168.51.1")
r3.cmd("route add -net 192.168.55.0/24 gw 192.168.54.2")
r3.cmd("route add -net 192.168.57.0/24 gw 192.168.54.2")
r3.cmd("route add -net 192.168.52.0/24 gw 192.168.51.1")
r3.cmd("route add -net 192.168.56.0/24 gw 192.168.54.2")

r4.cmd("route add -net 192.168.57.0/24 gw 192.168.56.2")
r4.cmd("route add -net 192.168.50.0/24 gw 192.168.52.1")
r4.cmd("route add -net 192.168.54.0/24 gw 192.168.56.2")
r4.cmd("route add -net 192.168.53.0/24 gw 192.168.52.1")
r4.cmd("route add -net 192.168.51.0/24 gw 192.168.52.1")
```

Uji konektivitas r1 dengan r2 dan r3 dengan r4:

```
mininet> r1 ping r2
PING 192.168.57.1 (192.168.57.1) 56(84) bytes of data.
64 bytes from 192.168.57.1: icmp_seq=1 ttl=63 time=5.01 ms
^C
--- 192.168.57.1 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 5.005/5.005/5.005/0.000 ms
mininet> r3 ping r4
PING 192.168.55.2 (192.168.55.2) 56(84) bytes of data.
64 bytes from 192.168.55.2: icmp_seq=1 ttl=63 time=21.9 ms
^C
--- 192.168.55.2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 21.860/21.860/21.860/0.000 ms
mininet>
```

Hasil pingall:

```
mininet> pingall
*** Ping: testing ping reachability
r1 -> r2 r3 r4 ha hb
r2 -> r1 r3 r4 ha hb
r3 -> r1 r2 r4 ha hb
r4 -> r1 r2 r3 ha hb
ha -> r1 r2 r3 r4 hb
hb -> r1 r2 r3 r4 ha
*** Results: 0% dropped (30/30 received)
mininet>
```

Hasil konektivitas Host A ke seluruh node:

```
mininet> ha ping r1
connect failed: Connection refused
------------------------------------------------------------
Client connecting to 192.168.53.2, TCP port 5001
Binding to local address 192.168.57.2
TCP window size:  128 KByte (default)
------------------------------------------------------------
[  3] local 192.168.57.2 port 58031 connected with 192.168.53.2 port 5001
PING 192.168.50.2 (192.168.50.2) 56(84) bytes of data.
64 bytes from 192.168.50.2: icmp_seq=1 ttl=64 time=0.039 ms
^C
--- 192.168.50.2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.039/0.039/0.039/0.000 ms
mininet> ha ping r2
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0- 5.8 sec   896 KBytes  1.26 Mbits/sec
PING 192.168.57.1 (192.168.57.1) 56(84) bytes of data.
64 bytes from 192.168.57.1: icmp_seq=1 ttl=64 time=0.046 ms
^C
--- 192.168.57.1 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.046/0.046/0.046/0.000 ms
mininet> ha ping r3
PING 192.168.53.1 (192.168.53.1) 56(84) bytes of data.
64 bytes from 192.168.53.1: icmp_seq=1 ttl=63 time=0.058 ms
^C
--- 192.168.53.1 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.058/0.058/0.058/0.000 ms
mininet> ha ping r4
PING 192.168.55.2 (192.168.55.2) 56(84) bytes of data.
64 bytes from 192.168.55.2: icmp_seq=1 ttl=63 time=0.044 ms
^C
--- 192.168.55.2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.044/0.044/0.044/0.000 ms
mininet> ha ping hb
PING 192.168.53.2 (192.168.53.2) 56(84) bytes of data.
64 bytes from 192.168.53.2: icmp_seq=1 ttl=62 time=0.074 ms
^C
--- 192.168.53.2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.074/0.074/0.074/0.000 ms
mininet>
```

Hasil konektivitas Host B ke seluruh node:

```
mininet> hb ping r1
------------------------------------------------------------
Server listening on TCP port 5001
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
[  4] local 192.168.53.2 port 5001 connected with 192.168.57.2 port 58031
[ ID] Interval       Transfer     Bandwidth
[  4]  0.0- 7.7 sec   896 KBytes   958 Kbits/sec
PING 192.168.50.2 (192.168.50.2) 56(84) bytes of data.
64 bytes from 192.168.50.2: icmp_seq=1 ttl=63 time=0.076 ms
^C
--- 192.168.50.2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.076/0.076/0.076/0.000 ms
mininet> hb ping r2
PING 192.168.57.1 (192.168.57.1) 56(84) bytes of data.
64 bytes from 192.168.57.1: icmp_seq=1 ttl=63 time=0.076 ms
^C
--- 192.168.57.1 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.076/0.076/0.076/0.000 ms
mininet> hb ping r3
PING 192.168.53.1 (192.168.53.1) 56(84) bytes of data.
64 bytes from 192.168.53.1: icmp_seq=1 ttl=64 time=0.047 ms
^C
--- 192.168.53.1 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.047/0.047/0.047/0.000 ms
mininet> hb ping r4
PING 192.168.55.2 (192.168.55.2) 56(84) bytes of data.
64 bytes from 192.168.55.2: icmp_seq=1 ttl=64 time=0.054 ms
^C
--- 192.168.55.2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.054/0.054/0.054/0.000 ms
mininet> hb ping ha
PING 192.168.50.1 (192.168.50.1) 56(84) bytes of data.
64 bytes from 192.168.50.1: icmp_seq=1 ttl=62 time=0.068 ms
^C
--- 192.168.50.1 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.068/0.068/0.068/0.000 ms
mininet>
```

Traceroute Host A ke seluruh node:

```
mininet> ha traceroute -n r1
traceroute to 192.168.50.2 (192.168.50.2), 30 hops max, 60 byte packets
 1  192.168.50.2  0.255 ms  0.209 ms  0.192 ms
mininet> ha traceroute -n r2
traceroute to 192.168.57.1 (192.168.57.1), 30 hops max, 60 byte packets
 1  192.168.57.1  0.282 ms  0.234 ms  0.217 ms
mininet> ha traceroute -n r3
traceroute to 192.168.53.1 (192.168.53.1), 30 hops max, 60 byte packets
 1  192.168.57.1  0.350 ms  0.305 ms  0.288 ms
 2  192.168.53.1  0.276 ms  0.249 ms  0.226 ms
mininet> ha traceroute -n r4
traceroute to 192.168.55.2 (192.168.55.2), 30 hops max, 60 byte packets
 1  192.168.57.1  0.364 ms  0.317 ms  0.301 ms
 2  192.168.55.2  0.287 ms  0.259 ms  0.240 ms
mininet> ha traceroute -n hb
traceroute to 192.168.53.2 (192.168.53.2), 30 hops max, 60 byte packets
 1  192.168.57.1  0.433 ms  0.388 ms  0.372 ms
 2  192.168.54.1  0.358 ms  0.333 ms  0.313 ms
 3  192.168.53.2  0.294 ms  0.258 ms  0.230 ms
mininet>
```

Traceroute Host B ke seluruh node:

```
mininet> hb traceroute -n r1
traceroute to 192.168.50.2 (192.168.50.2), 30 hops max, 60 byte packets
 1  192.168.55.2  0.327 ms  0.282 ms  0.264 ms
 2  192.168.50.2  0.250 ms  0.222 ms  0.204 ms
mininet> hb traceroute -n r2
traceroute to 192.168.57.1 (192.168.57.1), 30 hops max, 60 byte packets
 1  192.168.55.2  0.329 ms  0.283 ms  0.266 ms
 2  192.168.57.1  0.252 ms  0.221 ms  0.203 ms
mininet> hb traceroute -n r3
traceroute to 192.168.53.1 (192.168.53.1), 30 hops max, 60 byte packets
 1  192.168.53.1  0.265 ms  0.214 ms  0.195 ms
mininet> hb traceroute -n r4
traceroute to 192.168.55.2 (192.168.55.2), 30 hops max, 60 byte packets
 1  192.168.55.2  0.272 ms  0.225 ms  0.205 ms
mininet> hb traceroute -n ha
traceroute to 192.168.50.1 (192.168.50.1), 30 hops max, 60 byte packets
 1  192.168.55.2  0.532 ms  0.500 ms  0.464 ms
 2  192.168.52.1  0.453 ms  0.416 ms  0.379 ms
 3  192.168.50.1  0.365 ms  0.302 ms  0.275 ms
mininet>
```

Traceroute r1 dengan r2 dan r3 dengan r4:

```
mininet> r1 traceroute -n r2
traceroute to 192.168.57.1 (192.168.57.1), 30 hops max, 60 byte packets
 1  192.168.51.2  0.316 ms  0.259 ms  0.242 ms
 2  192.168.57.1  0.228 ms  0.203 ms  0.185 ms
mininet> r3 traceroute -n r4
traceroute to 192.168.55.2 (192.168.55.2), 30 hops max, 60 byte packets
 1  192.168.54.2  0.440 ms  0.393 ms  0.373 ms
 2  192.168.55.2  0.360 ms  0.332 ms  0.314 ms
mininet>
```

C. CLO 3 (Capture Packet)

CLO 3 memiliki tujuan yang dimana membuat mahasiswa memahami bagaimana cara melakukan capture paket yang nantinya akan digunakan di wireshark dan tcp. Berikut adalah hasil dari CLO 3 yang saya kerjakan yang dimana menampilkan cara atau langkah - langkah capture packet:

1. CLI (Command Line Interface) linux mode mininet dan ketikkan xterm Host A

```
mininet> xterm ha
mininet>
```

2. Ketikkan di terminal Host A perintah tcpdump

```
root@rafi:/home/rafi# tcpdump -w tcpdump.pcap -c 200
tcpdump: listening on ha-eth0, link-type EN10MB (Ethernet), capture size 262144
bytes
```

Penjelasan kode: Membuat file protocol tcp yang berbentuk pcap dapat diakses melalui wireshark dengan tampilan 200 paket

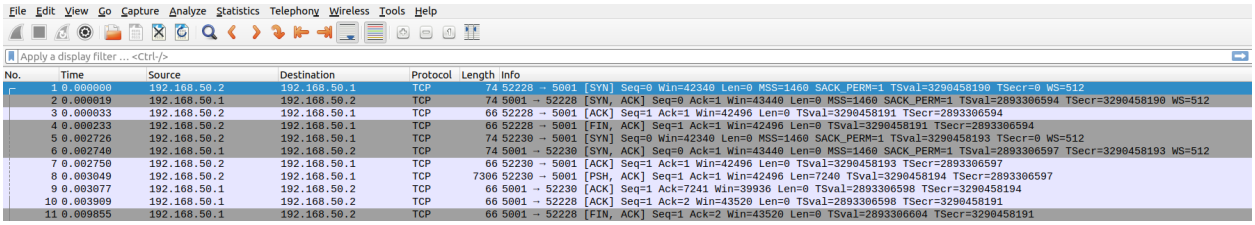
3. Kembali ke CLI linux lalu ketikkan iperf

```
mininet> iperf r1 ha
*** Iperf: testing TCP bandwidth between r1 and ha
*** Results: ['959 Kbits/sec', '1.23 Mbits/sec']
mininet>
```

Hasil dari iperf testing:

```
root@rafi:/home/rafi# tcpdump -w tcpdump.pcap -c 200
tcpdump: listening on ha-eth0, link-type EN10MB (Ethernet), capture size 262144
bytes
200 packets captured
262 packets received by filter
0 packets dropped by kernel
root@rafi:/home/rafi#
```

4. Buka file pcap yang sudah dibuat dari perintah sebelumnya



Dapat dilihat bahwa file pcap berisi paket yang mempunyai protocol TCP dengan adanya flag dari proses Three-Way Handshake. Proses tersebut memerlukan flag SYN, SYN - ACK, dan ACK agar dapat dikatakan proses Three-Way Handshake. Protocol TCP berbeda dengan UDP yang dimana UDP adalah connectionless protocol, sedangkan TCP adalah connection-oriented protocol. Di dalam file pcap juga terdapat ip source, ip destination, flags, protocol, length, dan lainnya.

D. CLO 4 (Jaringan Buffer)

CLO 4 memiliki tujuan yang dimana membuat mahasiswa memahami bagaimana cara menganalisis dan mengatur buffer, serta memasang trafic menggunakan iperf. Berikut adalah hasil dari CLO 4 yang saya kerjakan:

Perubahan pada program:

```
# Add Link
net.addLink(r1, ha, max_queue_size = 100, use_htb = True, intfName1 = 'r1-eth0', intfName2 = 'ha-eth0', cls=TCLink, **bw1)
net.addLink(r2, ha, max_queue_size = 100, use_htb = True, intfName1 = 'r2-eth1', intfName2 = 'ha-eth1', cls=TCLink, **bw1)
net.addLink(r3, hb, max_queue_size = 100, use_htb = True, intfName1 = 'r3-eth0', intfName2 = 'hb-eth0', cls=TCLink, **bw1)
net.addLink(r4, hb, max_queue_size = 100, use_htb = True, intfName1 = 'r4-eth1', intfName2 = 'hb-eth1', cls=TCLink, **bw1)
net.addLink(r1, r3, max_queue_size = 100, use_htb = True, intfName1 = 'r1-eth1', intfName2 = 'r3-eth1', cls=TCLink, **bw2)
net.addLink(r1, r4, max_queue_size = 100, use_htb = True, intfName1 = 'r1-eth2', intfName2 = 'r4-eth2', cls=TCLink, **bw1)
net.addLink(r2, r4, max_queue_size = 100, use_htb = True, intfName1 = 'r2-eth0', intfName2 = 'r4-eth0', cls=TCLink, **bw2)
net.addLink(r2, r3, max_queue_size = 100, use_htb = True, intfName1 = 'r2-eth2', intfName2 = 'r3-eth2', cls=TCLink, **bw1)
net.build()
```

```
# Iperf
hb.cmd('iperf -s &')
ha.cmd('iperf -t 10 -B 192.168.50.1 -c 192.168.53.2 &')
ha.cmd('iperf -t 10 -B 192.168.57.2 -c 192.168.53.2 &')

#Static Routing Host
ha.cmd("ip rule add from 192.168.50.1 table 1")
ha.cmd("ip rule add from 192.168.57.2 table 2")
ha.cmd("ip rule add 192.168.50.0/24 dev ha-eth0 scope link table 1")
ha.cmd("ip rule add default via 192.168.50.2 dev ha-eth0 table 1")
ha.cmd("ip rule add 192.168.57.0/24 dev ha-eth1 scope link table 2")
ha.cmd("ip rule add default via 192.168.57.1 dev ha-eth1 table 2")
ha.cmd("ip rule add default scope global nexthop via 192.168.50.2 dev ha-eth0")
ha.cmd("ip rule add default scope global nexthop via 192.168.57.1 dev ha-eth1")
ha.cmd("route add default gw 192.168.50.2 dev ha-eth0")
ha.cmd("route add default gw 192.168.57.1 dev ha-eth1")

hb.cmd("ip rule add from 192.168.53.2 table 3")
hb.cmd("ip rule add from 192.168.55.1 table 4")
hb.cmd("ip rule add 192.168.53.0/24 dev hb-eth0 scope link table 1")
hb.cmd("ip rule add default via 192.168.53.1 dev hb-eth0 table 1")
hb.cmd("ip rule add 192.168.55.0/24 dev hb-eth1 scope link table 2")
hb.cmd("ip rule add default via 192.168.55.2 dev hb-eth1 table 2")
hb.cmd("ip rule add default scope global nexthop via 192.168.53.1 dev hb-eth0")
hb.cmd("ip rule add default scope global nexthop via 192.168.55.2 dev hb-eth1")
hb.cmd("route add default gw 192.168.53.1 dev hb-eth0")
hb.cmd("route add default gw 192.168.55.2 dev hb-eth1")
```

Potongan kode yang mengatur buffer size adalah di max_queue_size, kode tersebut ditambahkan di bagian add link dengan menambahkan use_htb juga di samping kode tersebut. Traffic diatur dibagian routing host yang dimana setiap host ditambahkan dua baris kode terakhir sebagai rute agar buffer berjalan. Iperf juga perlu ditambahkan supaya traffic bisa dapat digunakan antar host.

1. Mengatur buffer menjadi 20

```
mininet> ha ping hb
------------------------------------------------------------
Client connecting to 192.168.53.2, TCP port 5001
Binding to local address 192.168.50.1
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
[  3] local 192.168.50.1 port 35019 connected with 192.168.53.2 port 5001
------------------------------------------------------------
Client connecting to 192.168.53.2, TCP port 5001
Binding to local address 192.168.57.2
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
[  3] local 192.168.57.2 port 52721 connected with 192.168.53.2 port 5001
PING 192.168.53.2 (192.168.53.2) 56(84) bytes of data.
64 bytes from 192.168.53.2: icmp_seq=1 ttl=62 time=111 ms
64 bytes from 192.168.53.2: icmp_seq=2 ttl=62 time=103 ms
64 bytes from 192.168.53.2: icmp_seq=3 ttl=62 time=120 ms
64 bytes from 192.168.53.2: icmp_seq=4 ttl=62 time=123 ms
64 bytes from 192.168.53.2: icmp_seq=5 ttl=62 time=104 ms
64 bytes from 192.168.53.2: icmp_seq=6 ttl=62 time=121 ms
64 bytes from 192.168.53.2: icmp_seq=7 ttl=62 time=114 ms
64 bytes from 192.168.53.2: icmp_seq=8 ttl=62 time=119 ms
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0-10.4 sec   896 KBytes   707 Kbits/sec
64 bytes from 192.168.53.2: icmp_seq=9 ttl=62 time=99.6 ms
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0-11.3 sec   768 KBytes   556 Kbits/sec
64 bytes from 192.168.53.2: icmp_seq=10 ttl=62 time=115 ms
^C
--- 192.168.53.2 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9015ms
rtt min/avg/max/mdev = 99.559/112.970/122.680/7.660 ms
mininet>
```

2. Mengatur buffer menjadi 40

```
mininet> ha ping hb
------------------------------------------------------------
Client connecting to 192.168.53.2, TCP port 5001
Binding to local address 192.168.50.1
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
[  3] local 192.168.50.1 port 60725 connected with 192.168.53.2 port 5001
------------------------------------------------------------
Client connecting to 192.168.53.2, TCP port 5001
Binding to local address 192.168.57.2
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
[  3] local 192.168.57.2 port 51263 connected with 192.168.53.2 port 5001
PING 192.168.53.2 (192.168.53.2) 56(84) bytes of data.
64 bytes from 192.168.53.2: icmp_seq=1 ttl=62 time=101 ms
64 bytes from 192.168.53.2: icmp_seq=2 ttl=62 time=105 ms
64 bytes from 192.168.53.2: icmp_seq=3 ttl=62 time=97.1 ms
64 bytes from 192.168.53.2: icmp_seq=4 ttl=62 time=126 ms
64 bytes from 192.168.53.2: icmp_seq=5 ttl=62 time=107 ms
64 bytes from 192.168.53.2: icmp_seq=6 ttl=62 time=113 ms
64 bytes from 192.168.53.2: icmp_seq=7 ttl=62 time=106 ms
64 bytes from 192.168.53.2: icmp_seq=8 ttl=62 time=112 ms
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0-10.3 sec   896 KBytes   713 Kbits/sec
64 bytes from 192.168.53.2: icmp_seq=9 ttl=62 time=129 ms
64 bytes from 192.168.53.2: icmp_seq=10 ttl=62 time=98.4 ms
^C
--- 192.168.53.2 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9009ms
rtt min/avg/max/mdev = 97.146/109.633/129.308/10.404 ms
mininet>
```

3. Mengatur buffer menjadi 60

```
mininet> ha ping hb
------------------------------------------------------------
Client connecting to 192.168.53.2, TCP port 5001
Binding to local address 192.168.50.1
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
[  3] local 192.168.50.1 port 33657 connected with 192.168.53.2 port 5001
------------------------------------------------------------
Client connecting to 192.168.53.2, TCP port 5001
Binding to local address 192.168.57.2
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
[  3] local 192.168.57.2 port 48237 connected with 192.168.53.2 port 5001
PING 192.168.53.2 (192.168.53.2) 56(84) bytes of data.
64 bytes from 192.168.53.2: icmp_seq=1 ttl=62 time=117 ms
64 bytes from 192.168.53.2: icmp_seq=2 ttl=62 time=98.3 ms
64 bytes from 192.168.53.2: icmp_seq=3 ttl=62 time=115 ms
64 bytes from 192.168.53.2: icmp_seq=4 ttl=62 time=133 ms
64 bytes from 192.168.53.2: icmp_seq=5 ttl=62 time=113 ms
64 bytes from 192.168.53.2: icmp_seq=6 ttl=62 time=119 ms
64 bytes from 192.168.53.2: icmp_seq=7 ttl=62 time=109 ms
64 bytes from 192.168.53.2: icmp_seq=8 ttl=62 time=127 ms
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0-10.4 sec   896 KBytes   707 Kbits/sec
64 bytes from 192.168.53.2: icmp_seq=9 ttl=62 time=132 ms
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0-11.3 sec   768 KBytes   556 Kbits/sec
64 bytes from 192.168.53.2: icmp_seq=10 ttl=62 time=112 ms
^C
--- 192.168.53.2 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9012ms
rtt min/avg/max/mdev = 98.288/117.573/133.289/10.218 ms
mininet>
```

4. Mengatur buffer menjadi 100

```
mininet> ha ping hb
------------------------------------------------------------
Client connecting to 192.168.53.2, TCP port 5001
Binding to local address 192.168.50.1
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
[  3] local 192.168.50.1 port 51371 connected with 192.168.53.2 port 5001
------------------------------------------------------------
Client connecting to 192.168.53.2, TCP port 5001
Binding to local address 192.168.57.2
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
[  3] local 192.168.57.2 port 39143 connected with 192.168.53.2 port 5001
PING 192.168.53.2 (192.168.53.2) 56(84) bytes of data.
64 bytes from 192.168.53.2: icmp_seq=1 ttl=62 time=129 ms
64 bytes from 192.168.53.2: icmp_seq=2 ttl=62 time=121 ms
64 bytes from 192.168.53.2: icmp_seq=3 ttl=62 time=125 ms
64 bytes from 192.168.53.2: icmp_seq=4 ttl=62 time=106 ms
64 bytes from 192.168.53.2: icmp_seq=5 ttl=62 time=123 ms
64 bytes from 192.168.53.2: icmp_seq=6 ttl=62 time=104 ms
64 bytes from 192.168.53.2: icmp_seq=7 ttl=62 time=108 ms
64 bytes from 192.168.53.2: icmp_seq=8 ttl=62 time=101 ms
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0-10.4 sec   896 KBytes   707 Kbits/sec
64 bytes from 192.168.53.2: icmp_seq=9 ttl=62 time=117 ms
64 bytes from 192.168.53.2: icmp_seq=10 ttl=62 time=133 ms
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0-11.7 sec   768 KBytes   538 Kbits/sec
^C
--- 192.168.53.2 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9014ms
rtt min/avg/max/mdev = 100.576/116.606/132.931/10.736 ms
mininet>
```

Analisa Buffer:

Buffer mempengaruhi bandwidth yang dimana semakin besar buffer yang digunakan maka semakin besar juga waktu yang diperlukan untuk mengirim paket. Buffer 100 akan memiliki waktu yang lebih lama untuk mengirim paket maka buffer 20 tentu lebih baik daripada buffer 100. Dilihat dari hasil maka buffer 20 dan 40 memiliki perbedaan sedikit dibandingkan dengan buffer 60 dan 100 yang memiliki perbedaan cukup jauh.

BAB III

PENUTUPAN

A. Kesimpulan

Kesimpulan yang dapat saya berikan adalah membangun sebuah topologi jaringan yang berfungsi dapat dilakukan dengan mininet. Diperlukan subnetting terlebih dahulu agar topologi jaringan dapat berjalan sebaik mungkin. Topologi jaringan juga memiliki protocol untuk tiap paket yang dimana dibedakan menjadi 2 yaitu TCP dan UDP. Berhubungan dengan kualitas jaringan atau bandwidth, adanya buffer atau delay yang akan mempengaruhi kinerja jaringan. Topologi jaringan dengan router juga memperlukan adanya mekanisme routing untuk menghubungkan jaringan yang berada dalam area yang berbeda. Paket dari topologi jaringan yang dibuat juga dapat di analisis melalui aplikasi wireshark untuk menganalisis flag, protocol, ip source - destination, dan sebagainya.

B. Saran

Laporan dan program berikut masih jauh dari kata sempurna, saya sangat berterima kasih kepada asprak saya ABDUL ROZZAK JUNAIDI yang sudah membimbing saya agar saya dapat memahami bagaimana cara menyelesaikan tugas besar berikut. Saya berharap bahwa laporan dan program saya dapat berguna dengan tujuan pembelajaran pada mata kuliah berikutnya yang berhubungan dengan jaringan komputer.