

# **LAPORAN TUGAS BESAR JARINGAN KOMPUTER**

**Ditya Ilmi Rizqi**

[dityailmirizqi@student.telkomuniversity.ac.id](mailto:dityailmirizqi@student.telkomuniversity.ac.id)

1303191119

IT-43-03

S1 Teknologi Informasi, Fakultas Informatika

Universitas Telkom, 2021

## **ABSTRAK**

Kebutuhan akan teknologi perangkat jaringan di berbagai perusahaan mengalami perkembangan yang sangat pesat. Performansi jaringan, penambahan konfigurasi yang semakin kompleks dan besar, bagian kontrol jaringan pun akan semakin rumit, tidak fleksibel dan sulit untuk diatur. Mininet yaitu emulator jaringan opensource yang digunakan dalam tugas ini. Dengan menggunakan mininet, hal-hal yang biasa kita jumpai dapat disimulasikan didalam satu komputer yang sama tanpa membutuhkan alat-alat eksternal. Laporan ini akan menunjukkan hasil dari simulasi yang telah saya buat, dimulai dari membangun topologi, menunjukkan routing, melihat penggunaan TCP di dalam jaringan dan juga perbandingannya dengan MPTCP.

## **BAB I**

### **PENDAHULUAN**

#### **1.1 Latar Belakang**

Perkembangan teknologi yang sangat pesat tentunya tidak lepas dari peran Topologi jaringan sebagai dasar perkembangan teknologi yang ada. Fungsi dari topologi jaringan sendiri adalah untuk membuat komputer saling terhubung, sehingga pertukaran informasi, data, ataupun komunikasi bisa dilakukan dengan lebih cepat dan juga mudah.

#### **1.2 Rumusan Masalah**

Rumusan masalah dari laporan ini diantaranya yaitu :

1. Bagaimana melakukan konfigurasi terhadap jaringan?
2. Bagaimana membangun topologi yang sesuai dengan soal yang diberikan?
3. Bagaimana mekanisme routing pada topologi yang ada?

#### **1.3 Tujuan**

Tujuan dari laporan ini diantaranya yaitu :

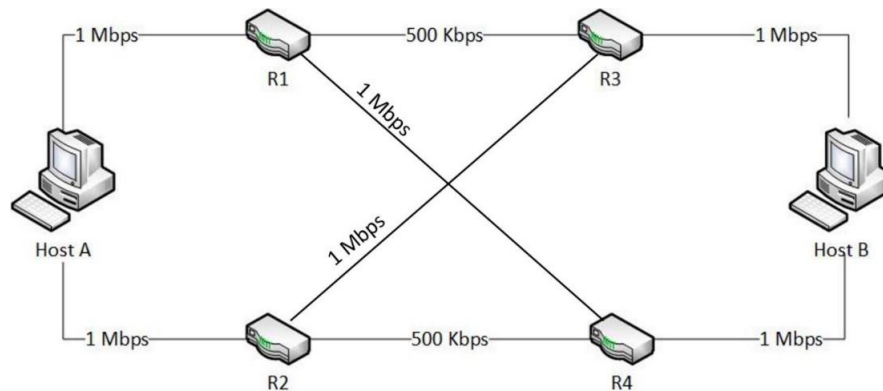
1. Untuk mengetahui bagaimana melakukan konfigurasi terhadap jaringan
2. Untuk mengetahui bagaimana membangun topologi yang sesuai dengan soal yang diberikan
3. Untuk mengetahui bagaimana mekanisme routing pada topologi yang ada

## BAB II

### OVERVIEW TUGAS BESAR

#### 2.1 Gambaran Tugas Besar

Pada tugas besar jaringan komputer, mahasiswa diminta untuk melakukan simulasi jaringan dengan menggunakan mininet. Dengan memanfaatkan antarmuka pemrograman aplikasi (API) yang disediakan oleh mininet dengan menggunakan bahasa pemrograman Python, topologi yang akan dibuat yaitu seperti dibawah ini:



Capaian dari pembelajaran pada matakuliah ini atau pada tubes ini yaitu :

#### A. CLO 1

(Membangun topologi)

Membangun topologi sesuai gambar dari soal yang diberikan.

(Membandingkan hasil MPTCP dan TCP)

Membandingkan dan menganalisis hasil dari mptcp dan tcp yaitu berapa bandwidth yang dibutuhkan dalam pengiriman paket.

#### B. CLO 2

(Subnetting)

Mendesain network dengan masing-masing subnet, Assign IP sesuai subnet, melakukan pengujian konektivitas pada satu jaringan yang sama.

(Static Routing)

Mengimplementasikan mekanisme static routing pada topologi jaringan yang dibuat, menguji konektivitas menggunakan program traceroute.

(Dynamic Routing)

## BAB III

### HASIL DAN ANALISIS

#### 3.1 Subnetting

Nama	Network ID	Host Range	Broadcast	Prefix	Subnet Mask
Link 1	192.168.100.0	192.168.100.1 - 192.168.100.2	192.168.100.3	/30	255.255.255. 252
Link 2	192.168.100.4	192.168.100.5 - 192.168.100.6	192.168.100.7	/30	255.255.255. 252
Link 3	192.168.100.8	192.168.100.9 - 192.168.100.10	192.168.100.11	/30	255.255.255. 252
Link 4	192.168.100.12	192.168.100.13 - 192.168.100.14	192.168.100.15	/30	255.255.255. 252
Link 5	192.168.0.0	192.168.0.1 - 192.168.0.254	192.168.0.255	/24	255.255.255. 0
Link 6	192.168.1.0	192.168.1.1 - 192.168.1.254	192.168.1.255	/24	255.255.255. 0
Link 7	192.168.2.0	192.168.2.1 - 192.168.2.254	192.168.2.255	/24	255.255.255. 0
Link 8	192.168.3.0	192.168.3.1 - 192.168.3.254	192.168.3.255	/24	255.255.255. 0

Sehingga jika diimplementasikan ke dalam topologi yang diminta akan terbentuk topologi pada sub bab 3.2 .

#### 3.2 Bangun Topologi

Disini saya membuat topologi dengan menggunakan ‘net.addHost’ sesuai gambar pada persoalan tugas besar yang diberikan dimana host diberi lambang h dan router diberi lambang r. Kemudian kita sambungkan dengan ‘net.addLink’, yang diikuti dengan memasukan cli dan bandwidth yang sudah di berikan soal

```
# Add hosts
ha = net.addHost('ha')
hb = net.addHost('hb')

# Add routers
r1 = net.addHost('r1')
r2 = net.addHost('r2')
r3 = net.addHost('r3')
r4 = net.addHost('r4')

bandwidth1={'bw':1}
bandwidth2={'bw':0.5}

# Add link hosts to router
net.addLink(ha, r1, cls=TCLink, **bandwidth1) #ha-eth0 r1-eth0
net.addLink(ha, r2, cls=TCLink, **bandwidth1) #ha-eth1 r2-eth0
net.addLink(hb, r3, cls=TCLink, **bandwidth1) #hb-eth0 r3-eth0
net.addLink(hb, r4, cls=TCLink, **bandwidth1) #hb-eth1 r4-eth0

# Add link router to router
net.addLink(r1, r3, cls=TCLink, **bandwidth2) #r1-eth1 r3-eth1
net.addLink(r1, r4, cls=TCLink, **bandwidth1) #r1-eth2 r4-eth1
net.addLink(r2, r3, cls=TCLink, **bandwidth1) #r2-eth1 r3-eth2
net.addLink(r2, r4, cls=TCLink, **bandwidth2) #r2-eth2 r4-eth2

net.build()
```

Kemudian kita buat interface dari masing-masing router, lalu kita build dengan cara memasukan ip untuk host maupun router yang ada.

```

# Interface
ha.cmd("ifconfig ha-eth0 0")
ha.cmd("ifconfig ha-eth1 0")

hb.cmd("ifconfig hb-eth0 0")
hb.cmd("ifconfig hb-eth1 0")

r1.cmd("ifconfig r1-eth0 0")
r1.cmd("ifconfig r1-eth1 0")
r1.cmd("ifconfig r1-eth2 0")

r2.cmd("ifconfig r2-eth0 0")
r2.cmd("ifconfig r2-eth1 0")
r2.cmd("ifconfig r2-eth2 0")

r3.cmd("ifconfig r3-eth0 0")
r3.cmd("ifconfig r3-eth1 0")
r3.cmd("ifconfig r3-eth2 0")

r4.cmd("ifconfig r4-eth0 0")
r4.cmd("ifconfig r4-eth1 0")
r4.cmd("ifconfig r4-eth2 0")

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

ha.cmd("ifconfig ha-eth0 192.168.0.2 netmask 255.255.255.252")
ha.cmd("ifconfig ha-eth1 192.168.1.2 netmask 255.255.255.0")

hb.cmd("ifconfig hb-eth0 192.168.2.2 netmask 255.255.255.0")
hb.cmd("ifconfig hb-eth1 192.168.3.2 netmask 255.255.255.0")

r1.cmd("ifconfig r1-eth0 192.168.0.1 netmask 255.255.255.252")
r1.cmd("ifconfig r1-eth1 192.168.100.1 netmask 255.255.255.252")
r1.cmd("ifconfig r1-eth2 192.168.100.5 netmask 255.255.255.252")

r2.cmd("ifconfig r2-eth0 192.168.1.1 netmask 255.255.255.0")
r2.cmd("ifconfig r2-eth1 192.168.100.9 netmask 255.255.255.252")
r2.cmd("ifconfig r2-eth2 192.168.100.13 netmask 255.255.255.0")

r3.cmd("ifconfig r3-eth0 192.168.2.1 netmask 255.255.255.0")
r3.cmd("ifconfig r3-eth1 192.168.100.2 netmask 255.255.255.252")
r3.cmd("ifconfig r3-eth2 192.168.100.10 netmask 255.255.255.252")

r4.cmd("ifconfig r4-eth0 192.168.3.1 netmask 255.255.255.0")
r4.cmd("ifconfig r4-eth1 192.168.100.6 netmask 255.255.255.252")
r4.cmd("ifconfig r4-eth2 192.168.100.14 netmask 255.255.255.0")

```

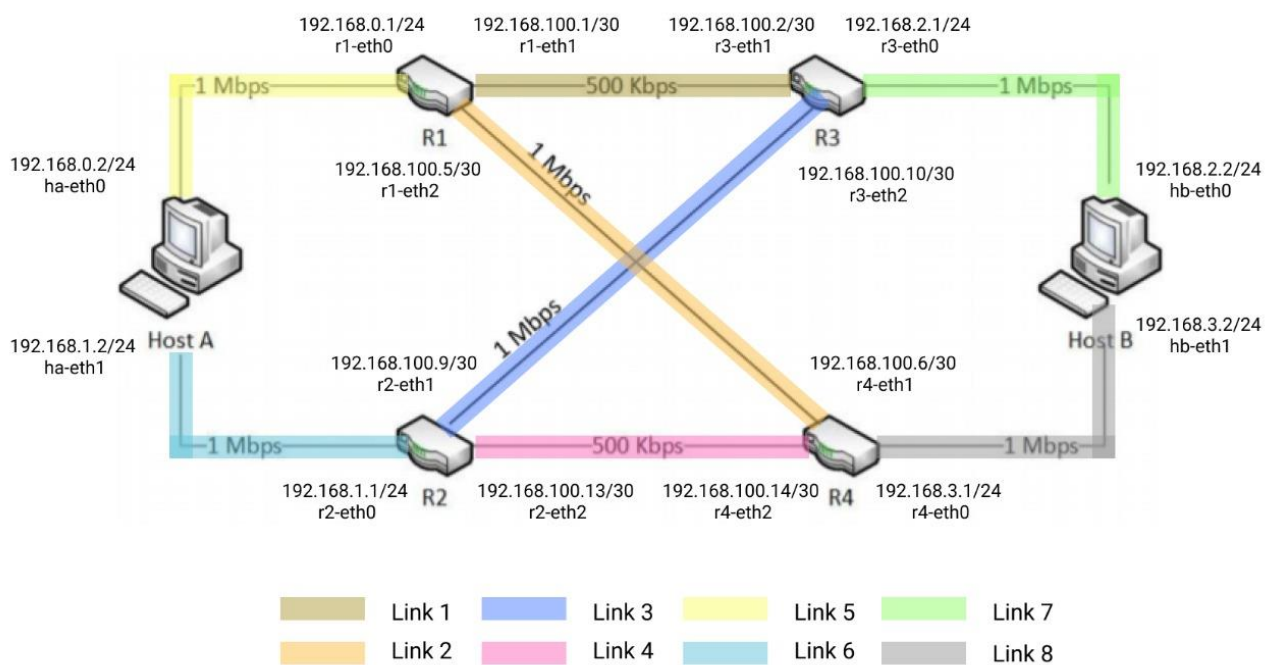
Setelah kita buat topologi diatas kemudian filenya kita run. Untuk running programnya saya gunakan sudo mn --custom tubes.py --topo mytopo. Kemudian kita cek nodesnya. Nodes sudah ada 2 host dan 4 router dimana hostnya yaitu ada ha dan hb, sedangkan routernya yaitu r1, r2, r3, dan r4

```

mininet> nodes
available nodes are:
ha hb r1 r2 r3 r4
mininet>

```

Ilustrasi dari topologi :



Mengecek interface yang dihubungkan sudah sesuai

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

Pada Topologi di atas saya menamakan Host A dan Host B menjadi ha dan hb. Setelah dilakukan konfigurasi saya mencoba melakukan *ping* kepada node-node yang berada satu jaringan, dan output yang dikeluarkan adalah sebagai berikut.

- r1 – r3

```
mininet> r1 ping r3
PING 192.168.2.1 (192.168.2.1) 56(84) bytes of data.
64 bytes from 192.168.2.1: icmp_seq=1 ttl=64 time=0.076 ms
64 bytes from 192.168.2.1: icmp_seq=2 ttl=64 time=0.077 ms
^C64 bytes from 192.168.2.1: icmp_seq=3 ttl=64 time=0.051 ms

--- 192.168.2.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2053ms
rtt min/avg/max/mdev = 0.051/0.068/0.077/0.012 ms
mininet> █
```

- r1 – r4

```
mininet> r1 ping r4
PING 192.168.3.1 (192.168.3.1) 56(84) bytes of data.
64 bytes from 192.168.3.1: icmp_seq=1 ttl=64 time=0.033 ms
64 bytes from 192.168.3.1: icmp_seq=2 ttl=64 time=0.049 ms
^C
--- 192.168.3.1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1026ms
rtt min/avg/max/mdev = 0.033/0.041/0.049/0.008 ms
mininet>
```

- r2 – r3

```
mininet> r2 ping r3
PING 192.168.2.1 (192.168.2.1) 56(84) bytes of data.
64 bytes from 192.168.2.1: icmp_seq=1 ttl=64 time=0.075 ms
64 bytes from 192.168.2.1: icmp_seq=2 ttl=64 time=0.049 ms
64 bytes from 192.168.2.1: icmp_seq=3 ttl=64 time=0.051 ms
^C
--- 192.168.2.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2049ms
rtt min/avg/max/mdev = 0.049/0.058/0.075/0.011 ms
mininet> █
```



- r2 – r4

```
mininet> r2 ping r4
PING 192.168.3.1 (192.168.3.1) 56(84) bytes of data.
64 bytes from 192.168.3.1: icmp_seq=1 ttl=64 time=0.048 ms
64 bytes from 192.168.3.1: icmp_seq=2 ttl=64 time=0.051 ms
64 bytes from 192.168.3.1: icmp_seq=3 ttl=64 time=0.058 ms
^C
--- 192.168.3.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2028ms
rtt min/avg/max/mdev = 0.048/0.052/0.058/0.004 ms
mininet>
```

- ha – r1

```
mininet> ha ping r1
PING 192.168.0.1 (192.168.0.1) 56(84) bytes of data.
64 bytes from 192.168.0.1: icmp_seq=1 ttl=64 time=0.048 ms
64 bytes from 192.168.0.1: icmp_seq=2 ttl=64 time=0.051 ms
64 bytes from 192.168.0.1: icmp_seq=3 ttl=64 time=0.044 ms
^C
--- 192.168.0.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2050ms
rtt min/avg/max/mdev = 0.044/0.047/0.051/0.003 ms
mininet>
```

- ha – r2

```
mininet> ha ping r2
PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data.
64 bytes from 192.168.1.1: icmp_seq=1 ttl=64 time=0.034 ms
64 bytes from 192.168.1.1: icmp_seq=2 ttl=64 time=0.093 ms
64 bytes from 192.168.1.1: icmp_seq=3 ttl=64 time=0.053 ms
^C
--- 192.168.1.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2051ms
rtt min/avg/max/mdev = 0.034/0.060/0.093/0.024 ms
mininet>
```

- hb – r3

```
mininet> hb ping r3
PING 192.168.2.1 (192.168.2.1) 56(84) bytes of data.
64 bytes from 192.168.2.1: icmp_seq=1 ttl=64 time=0.036 ms
64 bytes from 192.168.2.1: icmp_seq=2 ttl=64 time=0.053 ms
64 bytes from 192.168.2.1: icmp_seq=3 ttl=64 time=0.052 ms
^C
--- 192.168.2.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2044ms
rtt min/avg/max/mdev = 0.036/0.047/0.053/0.007 ms
mininet> █
```

- hb – r4

```
mininet> hb ping r4
PING 192.168.3.1 (192.168.3.1) 56(84) bytes of data.
64 bytes from 192.168.3.1: icmp_seq=1 ttl=64 time=0.055 ms
64 bytes from 192.168.3.1: icmp_seq=2 ttl=64 time=0.049 ms
64 bytes from 192.168.3.1: icmp_seq=3 ttl=64 time=0.062 ms
^C
--- 192.168.3.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2036ms
rtt min/avg/max/mdev = 0.049/0.055/0.062/0.005 ms
mininet>
```

- ha – hb

```
mininet> ha ping hb
PING 192.168.2.2 (192.168.2.2) 56(84) bytes of data.
64 bytes from 192.168.2.2: icmp_seq=1 ttl=62 time=0.059 ms
64 bytes from 192.168.2.2: icmp_seq=2 ttl=62 time=0.114 ms
64 bytes from 192.168.2.2: icmp_seq=3 ttl=62 time=0.077 ms
^C
--- 192.168.2.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2050ms
rtt min/avg/max/mdev = 0.059/0.083/0.114/0.022 ms
mininet>
```

- ha – r4

```
mininet> ha ping r4
PING 192.168.3.1 (192.168.3.1) 56(84) bytes of data.
64 bytes from 192.168.3.1: icmp_seq=1 ttl=63 time=0.053 ms
64 bytes from 192.168.3.1: icmp_seq=2 ttl=63 time=0.070 ms
64 bytes from 192.168.3.1: icmp_seq=3 ttl=63 time=0.069 ms
^C
--- 192.168.3.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2026ms
rtt min/avg/max/mdev = 0.053/0.064/0.070/0.007 ms
mininet>
```

- pingall

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

Ketika dilakukan pingall maka semua router dan host terhubung.

Dapat terlihat bahwa ketika saya melakukan ping terhadap node yang berada satu network, statistik menyatakan bahwa tidak ada *Packet Loss*, yang mengartikan

bahwa semua paket terkirim dengan benar. Kemudian kita lakukan pingall terhadap semua node hasilnya mereka terhubung antara satu dengan lainnya.

### 3.3 Static Routing

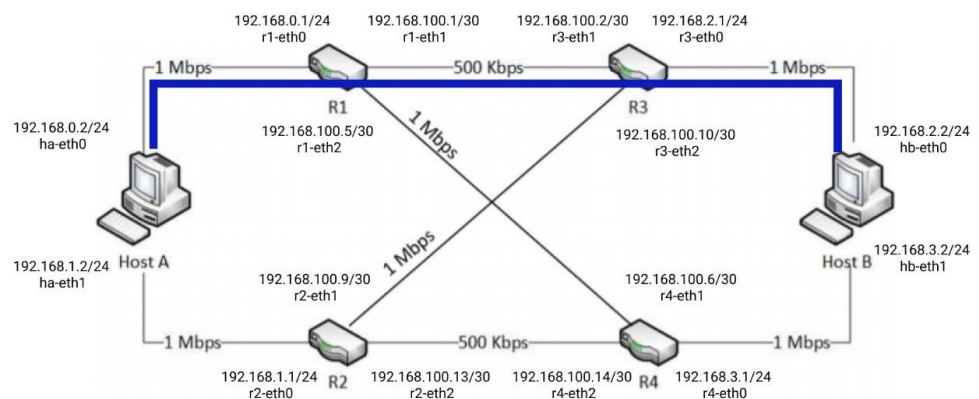
Pada static routing lakukan routing pada setiap node R1, R2, R3, dan R4 ke jaringan-jaringan yang tidak terhubung langsung kepada setiap node-node tersebut. Untuk mengetahui bahwa routing yang dikonfigurasi sudah benar, kita bisa menggunakan program *traceroute*. Dengan program tersebut saya bisa mengetahui bahwa paket yang dikirim melalui node mana saja.

Dicontohkan melakukan pengiriman paket dari

- ha-eth0 ke 192.168.2.2

```
mininet> ha traceroute 192.168.2.2
traceroute to 192.168.2.2 (192.168.2.2), 30 hops max, 60 byte packets
 1 192.168.0.1 (192.168.0.1) 0.039 ms 192.168.1.1 (192.168.1.1) 0.018 ms 192.168.0.1 (192.168.0.1) 0.008 ms
 2 192.168.100.10 (192.168.100.10) 0.019 ms 192.168.100.2 (192.168.100.2) 0.016 ms 192.168.100.10 (192.168.100.10) 0.010 ms
 3 192.168.2.2 (192.168.2.2) 0.023 ms 0.015 ms 0.013 ms
mininet>
```

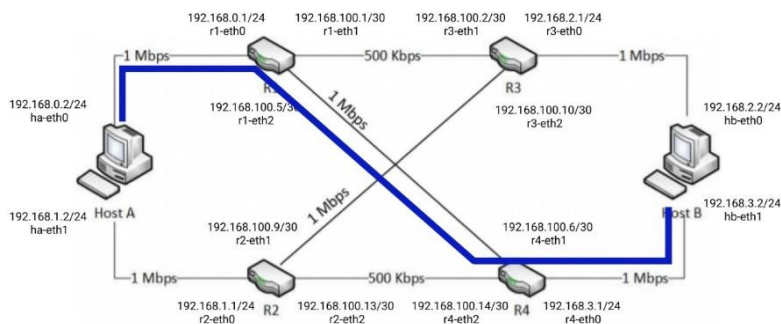
Maka ilustrasi topologi tersebut jalur yang digunakan yaitu :



- ha-eth0 ke 192.168.3.2

```
mininet> ha traceroute 192.168.3.2
traceroute to 192.168.3.2 (192.168.3.2), 30 hops max, 60 byte packets
 1 192.168.0.1 (192.168.0.1) 0.427 ms 192.168.1.1 (192.168.1.1) 0.379 ms 192.168.0.1 (192.168.0.1) 0.354 ms
 2 192.168.100.14 (192.168.100.14) 0.339 ms 192.168.100.6 (192.168.100.6) 0.313 ms 192.168.100.14 (192.168.100.14) 0.290 ms
 3 192.168.3.2 (192.168.3.2) 0.272 ms 0.244 ms 0.222 ms
mininet>
```

Maka ilustrasi topologi tersebut jalur yang digunakan yaitu :

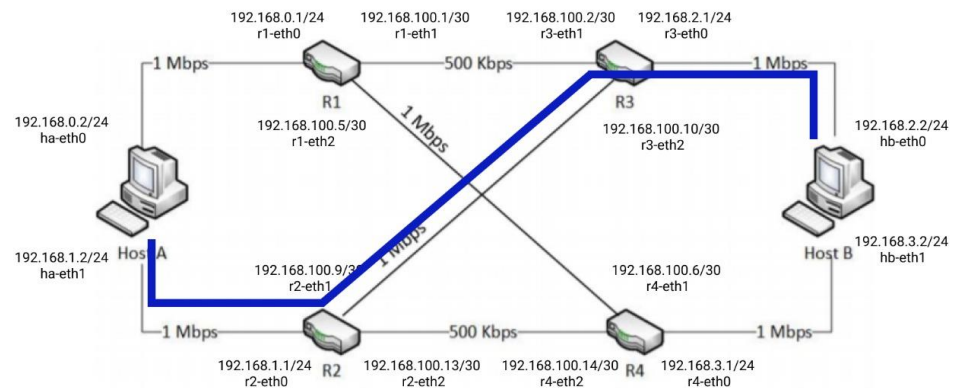




- ha-eth1 ke 192.168.3.2

```
mininet> ha traceroute -i ha-eth1 192.168.2.2
traceroute to 192.168.2.2 (192.168.2.2), 30 hops max, 60 byte packets
 1 192.168.1.1 (192.168.1.1)  0.034 ms  0.007 ms  0.006 ms
 2 192.168.100.10 (192.168.100.10)  0.017 ms  0.017 ms  0.010 ms
 3 192.168.2.2 (192.168.2.2)  0.021 ms  0.013 ms  0.014 ms
mininet>
```

Maka ilustrasi topologi tersebut jalur yang digunakan yaitu :



### 3.4 Perbandingan Hasil MPTCP dan TCP

#### 3.4.1 Hasil TCP

```

"Node: ha"
root@dir-VirtualBox:/home/dir/mininet/custom# tcpdump -w demo_tcp.pcap -c 200
tcpdump: listening on ha-eth0, link-type EN10MB (Ethernet), capture size 262144
bytes
200 packets captured
203 packets received by filter
0 packets dropped by kernel
root@dir-VirtualBox:/home/dir/mininet/custom#
```

```

mininet> xterm ha
mininet> iperf r3 ha
*** Iperf: testing TCP bandwidth between r3 and ha
*** Results: ['481 Kbits/sec', '750 Kbits/sec']
```

Hal yang dilakukan yaitu xterm node, disana saya xterm node ha. Kemudian kita bisa melihat bahwa listening on ha-eth0. Dan ketika di iperf r3 ha hasilnya testing TCP dengan bandwith yaitu 750 Kbits/sec.

### 3.4.2 Hasil MPTCP

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

```
Results: ['481 Kbits/sec', '704 Kbits/sec']
mininet> iperf r3 ha
*** Iperf: testing TCP bandwidth between r3 and ha
*** Results: ['481 Kbits/sec', '704 Kbits/sec']
mininet>
```

Hasil bandwidth yang dihasilkan oleh mptcp yaitu sama 481 namun waktu yang dibutuhkan lebih cepat yaitu 704 Kbits/sec

Dari hasil perbandingan antara tcp dan mptcp bisa kita lihat bahwa performa mptcp lebih baik daripada tcp karena waktu pengiriman data lebih cepat dan bandwidth yang dihasilkan lebih sedikit.