

LAPORAN TUGAS BESAR JARINGAN KOMPUTER

Kaenova Mahendra Auditama

kaenova@student.telkomuniversity.ac.id

1301190324

Jaringan Komputer IF-43-02 (CII2J4-IF-43-02)

Informatika, FT, Universitas Telkom. 2021.

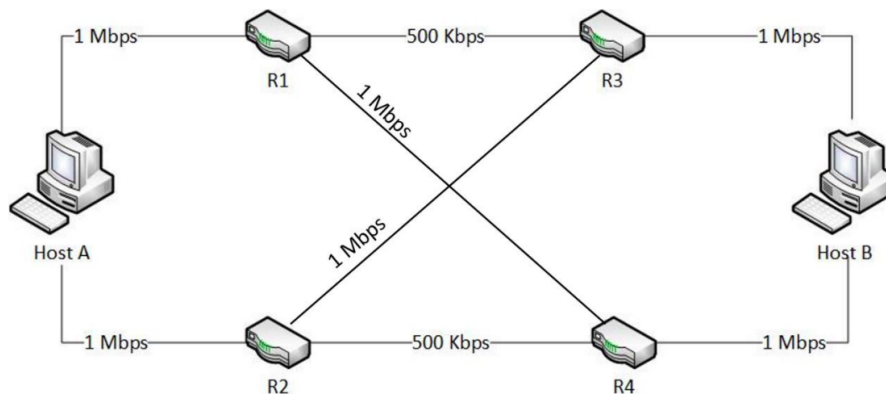
Jl. Telekomunikasi Terusan Buah Batu Indonesia 40257, Bandung, Indonesia

ABSTRAK

Jaringan merupakan salah satu hal terpenting dalam keterhubungan di era saat ini. Tugas ini ditujukan sebagai salah satu penerapan yang terjadi di dalam jaringan di dalam kehidupan sehari-hari. Dengan menggunakan mininet, hal-hal yang biasa kita jumpai dapat disimulasikan di dalam 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 melihat pengaruh ukuran buffer terhadap delay yang terjadi ketika jaringan penuh.

1. Overview Tugas

Tugas besar jaringan komputer merupakan tugas praktikum yang meminta kepada kami untuk melakukan simulasi jaringan dengan menggunakan mininet. Dengan memanfaatkan Antarmuka pemrograman aplikasi (API) yang disediakan oleh mininet dengan menggunakan bahasa pemrograman Python, saya diminta untuk membuat topologi seperti dibawah ini:



Gambar 1.1

Topologi Jaringan

Setelah membangun topologi, ada beberapa hal yang perlu dicapai, hal diantaranya ialah:

A. CLO 1 (Konfigurasi Jaringan)

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

B. CLO 2 (Static Routing)

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

C. CLO 3 (TCP)

Menginspeksi *traffic* TCP dengan mengutilisasi iperf sebagai pembuat *traffic* serta wireshark/tcpdump sebagai pengcapture *traffic* tersebut.

D. CLO 4 (Buffer Size)

Melakukan konfigurasi buffer dimulai dari 20, 40, 60, dan 100. Melakukan pemadatan di dalam jaringan dengan membuat *traffic* menggunakan iperf, melihat dan menganalisa pengaruh delay terhadap nilai buffer.

Untuk source code program dapat dilakukan *git clone* terhadap *repository github* di bawah ini atau mendownload melalui *Google Drive* melalui link di bawah ini. Untuk instruksi penggunaan diharapkan membaca README.txt atau README.md

Git Clone:

https://github.com/kaenova/Jarkom_Tubes

Google Drive:

https://bit.ly/Jarkom_Tubes

Note: Dalam laporan ini akan ada banyak “**ref code x**” dimana **x** adalah angka bilangan bulat. “**ref code x**” digunakan untuk mereferensikan bagian kode yang ada pada *source code*. Bagian kode-kode tersebut dapat ditemukan dengan mencari pada *source code* file *main.py* ataupun pada bagian akhir laporan ini.

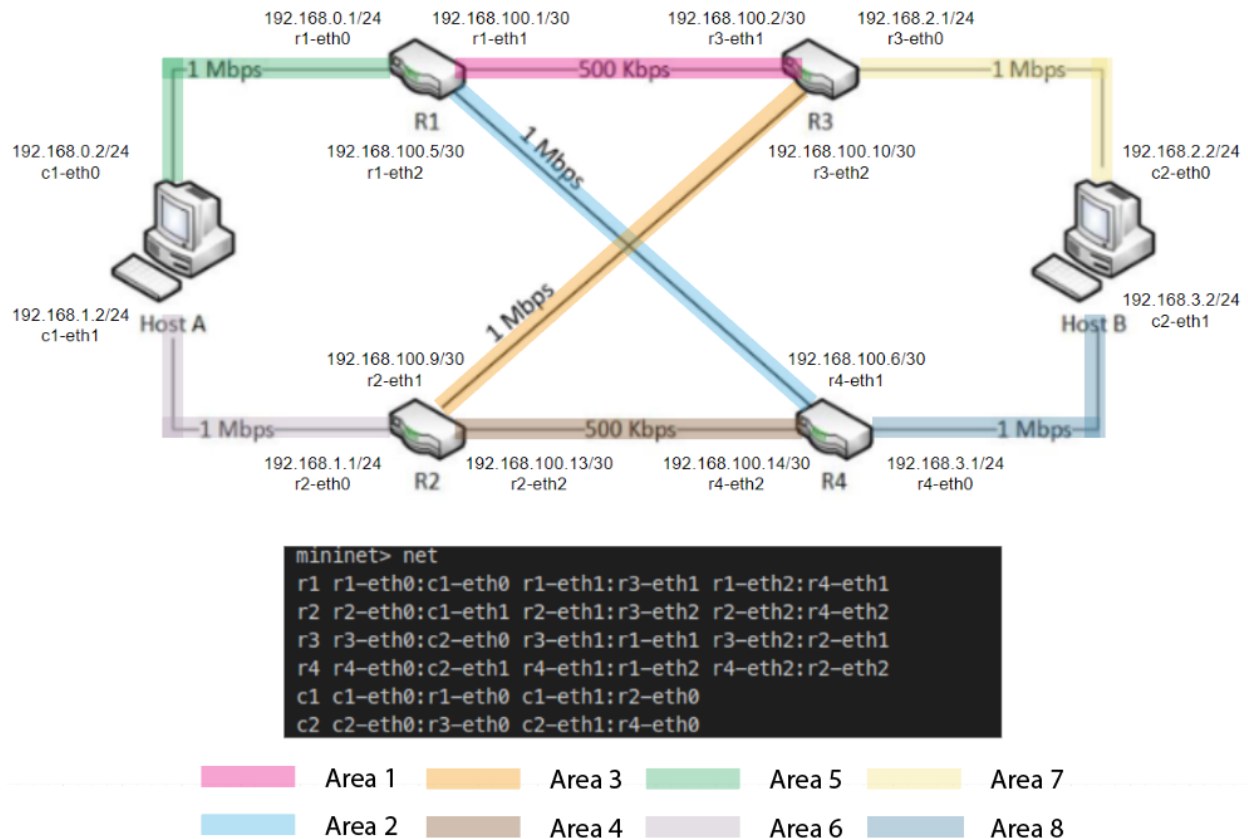
2. CLO 1 (Konfigurasi Jaringan)

Pada CLO 1 ini saya diminta untuk melakukan konfigurasi terhadap jaringan yang tersedia. Dimulai dengan subnetting table yang saya buat seperti ini:

Nama	Needs	Allocation	Network ID	Host Range	Broadcast	Prefix	Subnet Mask
Area 1	2	2	192.168.100.0	192.168.100.1 - 192.168.100.2	192.168.100.3	/30	255.255.255. 252
Area 2	2	2	192.168.100.4	192.168.100.5 - 192.168.100.6	192.168.100.7	/30	255.255.255. 252
Area 3	2	2	192.168.100.8	192.168.100.9 - 192.168.100.10	192.168.100.11	/30	255.255.255. 252
Area 4	2	2	192.168.100.12	192.168.100.13 - 192.168.100.14	192.168.100.15	/30	255.255.255. 252
Area 5	2	254	192.168.0.0	192.168.0.1 - 192.168.0.254	192.168.0.255	/24	255.255.255. 0

Area 6	2	254	192.168.1.0	192.168.1.1 - 192.168.1.254	192.168.1.255	/24	255.255.255. 0
Area 7	2	254	192.168.2.0	192.168.2.1 - 192.168.2.254	192.168.2.255	/24	255.255.255. 0
Area 8	2	254	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 seperti di bawah ini:



Gambar 2.1
Konfigurasi Topologi Jaringan

Pada Topologi di atas saya menamakan Host A dan Host B menjadi c1 dan c2. Untuk membangun kelas saya menggunakan kelas Topologi yang dapat dicari dan dilihat di dalam *source code* dengan kode comment “**ref code 1**” dan untuk konfigurasi setiap router dan komputer dapat dilakukan seperti pada “**ref code 2**”. Setelah dilakukan konfigurasi saya mencoba melakukan *ping* kepada node-node yang berada satu jaringan, dan output yang dikeluarkan adalah sebagai berikut.

Hasil catatan (log):

```
Kaenova Mahendra Auditama  
runCL01  
Check Network Connectivity
```

```
===== Reachable =====
```

```
R1 - R3 (Area 1)
```

```
PING 192.168.100.2 (192.168.100.2) 56(84) bytes of data.
```

```
64 bytes from 192.168.100.2: icmp_seq=1 ttl=64 time=0.119 ms
```

```
64 bytes from 192.168.100.2: icmp_seq=2 ttl=64 time=0.113 ms
```

```
64 bytes from 192.168.100.2: icmp_seq=3 ttl=64 time=0.121 ms
```

```
--- 192.168.100.2 ping statistics ---
```

```
3 packets transmitted, 3 received, 0% packet loss, time 2039ms
```

```
rtt min/avg/max/mdev = 0.113/0.117/0.121/0.003 ms
```

```
R1 - R4 (Area 2)
```

```
PING 192.168.100.6 (192.168.100.6) 56(84) bytes of data.
```

```
64 bytes from 192.168.100.6: icmp_seq=1 ttl=64 time=0.076 ms
```

```
64 bytes from 192.168.100.6: icmp_seq=2 ttl=64 time=0.120 ms
```

```
64 bytes from 192.168.100.6: icmp_seq=3 ttl=64 time=0.040 ms
```

```
--- 192.168.100.6 ping statistics ---
```

```
3 packets transmitted, 3 received, 0% packet loss, time 2015ms
```

```
rtt min/avg/max/mdev = 0.040/0.078/0.120/0.032 ms
```

```
R2 - R3 (Area 3)
```

```
PING 192.168.100.10 (192.168.100.10) 56(84) bytes of data.
```

```
64 bytes from 192.168.100.10: icmp_seq=1 ttl=64 time=0.054 ms
```

```
64 bytes from 192.168.100.10: icmp_seq=2 ttl=64 time=0.112 ms
```

```
64 bytes from 192.168.100.10: icmp_seq=3 ttl=64 time=0.107 ms
```

```
--- 192.168.100.10 ping statistics ---
```

```
3 packets transmitted, 3 received, 0% packet loss, time 2022ms
```

```
rtt min/avg/max/mdev = 0.054/0.091/0.112/0.026 ms
```

```
R2 - R4 (Area 4)
```

```
PING 192.168.100.14 (192.168.100.14) 56(84) bytes of data.
```

```
64 bytes from 192.168.100.14: icmp_seq=1 ttl=64 time=0.077 ms
```

```
64 bytes from 192.168.100.14: icmp_seq=2 ttl=64 time=0.132 ms
```

```
64 bytes from 192.168.100.14: icmp_seq=3 ttl=64 time=0.108 ms
```

```
--- 192.168.100.14 ping statistics ---
```

```
3 packets transmitted, 3 received, 0% packet loss, time 2016ms
```

```
rtt min/avg/max/mdev = 0.077/0.105/0.132/0.022 ms
```

C1 - R1 (Area 5)

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=0.079 ms

64 bytes from 192.168.0.1: icmp_seq=2 ttl=64 time=0.094 ms

64 bytes from 192.168.0.1: icmp_seq=3 ttl=64 time=0.118 ms

--- 192.168.0.1 ping statistics ---

3 packets transmitted, 3 received, 0% packet loss, time 2016ms

rtt min/avg/max/mdev = 0.079/0.097/0.118/0.016 ms

C2 - R3 (Area 7)

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=0.071 ms

64 bytes from 192.168.2.1: icmp_seq=2 ttl=64 time=0.113 ms

64 bytes from 192.168.2.1: icmp_seq=3 ttl=64 time=0.113 ms

--- 192.168.2.1 ping statistics ---

3 packets transmitted, 3 received, 0% packet loss, time 2018ms

rtt min/avg/max/mdev = 0.071/0.099/0.113/0.019 ms

C2 - R4 (Area 8)

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=0.207 ms

64 bytes from 192.168.3.1: icmp_seq=2 ttl=64 time=0.114 ms

64 bytes from 192.168.3.1: icmp_seq=3 ttl=64 time=0.077 ms

--- 192.168.3.1 ping statistics ---

3 packets transmitted, 3 received, 0% packet loss, time 2036ms

rtt min/avg/max/mdev = 0.077/0.132/0.207/0.054 ms

===== Unreachable =====

C1 - C2

PING 192.168.2.2 (192.168.2.2) 56(84) bytes of data.

From 192.168.1.1 icmp_seq=1 Destination Net Unreachable

From 192.168.1.1 icmp_seq=2 Destination Net Unreachable

From 192.168.1.1 icmp_seq=3 Destination Net Unreachable

--- 192.168.2.2 ping statistics ---

3 packets transmitted, 0 received, +3 errors, 100% packet loss, time 2039ms

C1 - R4

PING 192.168.3.1 (192.168.3.1) 56(84) bytes of data.

--- 192.168.3.1 ping statistics ---

3 packets transmitted, 0 received, 100% packet loss, time 2023ms

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, sementara di bagian *Unreachable* ketika saya mencoba melakukan *ping* terhadap area jaringan yang berbeda akan menyatakan “*Destination Net Unreachable*” ataupun tidak ada informasi sama sekali. *Packet Loss* ketika melakukan *ping* terhadap area jaringan yang berbeda pun menjadi 100% yang mengartikan tidak ada *packet* yang sampai ke *address* destinasi yang dituju.

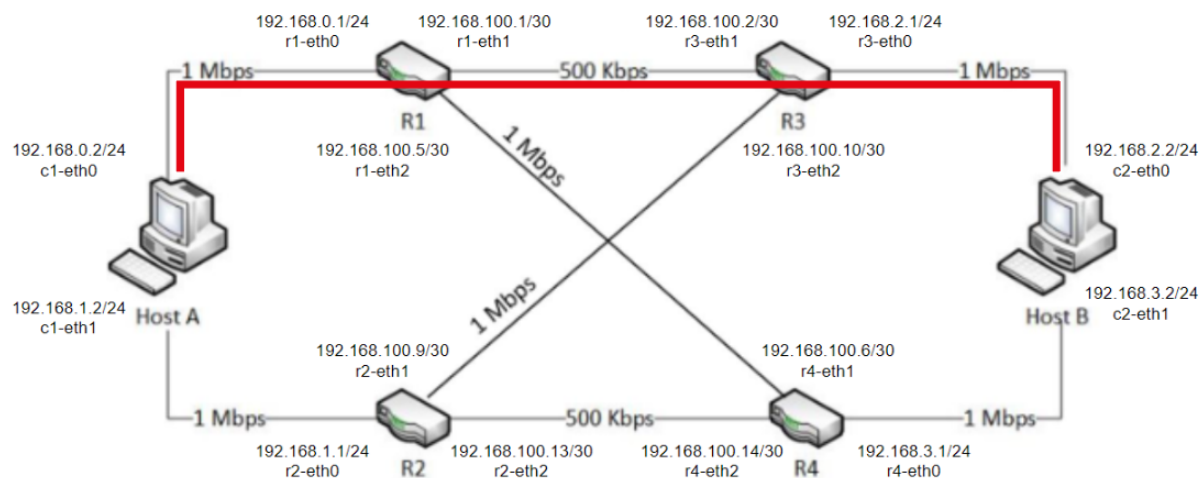
3. CLO 2 (Static Routing)

Setelah meembangun topologi dan memeriksa bahwa koneksi pada jaringan yang sama dapat terhubung. Saya diminta untuk melakukan routing pada setiap node R1, R2, R3, dan R4 kepada jaringan-jaringan yang tidak terhubung langsung kepada setiap *node-node* tersebut. Selain itu saya juga harus mengkonfigurasi *gateway* pada Host 1 dan Host 2. Semua bagian konfigurasi dapat dicari dan dilihat dari source code dengan comment “**ref code 3**”. Untuk mengetahui bahwa routing yang dikonfigurasi sudah benar, saya diminta untuk menggunakan program *traceroute*. Dengan program tersebut saya bisa mengetahui bahwa paket yang dikirim melalui node mana saja.

Dicontohkan melakukan pengiriman paket dari c1-eth0 ke 192.168.2.2:

```
mininet> c1 traceroute 192.168.2.2
traceroute to 192.168.2.2 (192.168.2.2), 30 hops max, 60 byte packets
 1 _gateway (192.168.0.1)  0.114 ms  0.033 ms  0.028 ms
 2 192.168.100.2 (192.168.100.2)  0.062 ms  0.043 ms  0.039 ms
 3 192.168.2.2 (192.168.2.2)  0.076 ms  0.057 ms  0.094 ms
mininet>
```

Jika digambarkan pada topologi maka jalur yang digunakan:



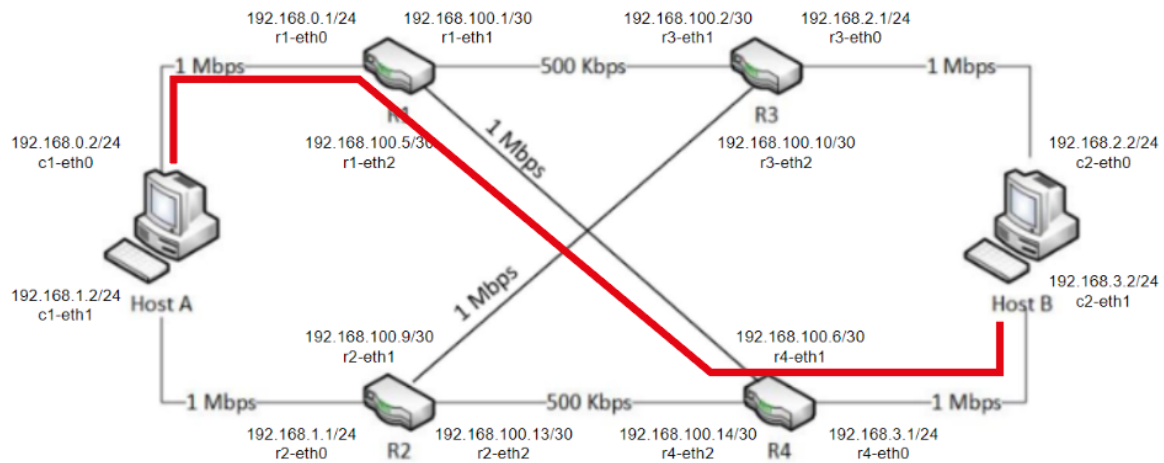
Gambar 3.1

Jalur Packet dari c1-eth0 ke 192.168.2.2

Dicontohkan lagi melakukan pengiriman paket dari c1-eth0 ke 192.168.3.2:

```
mininet> c1 traceroute 192.168.3.2
traceroute to 192.168.3.2 (192.168.3.2), 30 hops max, 60 byte packets
 1 _gateway (192.168.0.1)  0.143 ms  0.063 ms  0.048 ms
 2 192.168.100.6 (192.168.100.6)  0.171 ms  0.081 ms  0.085 ms
 3 192.168.3.2 (192.168.3.2)  0.181 ms  0.120 ms  0.061 ms
mininet>
```

Jika digambarkan pada topologi maka jalur yang digunakan:



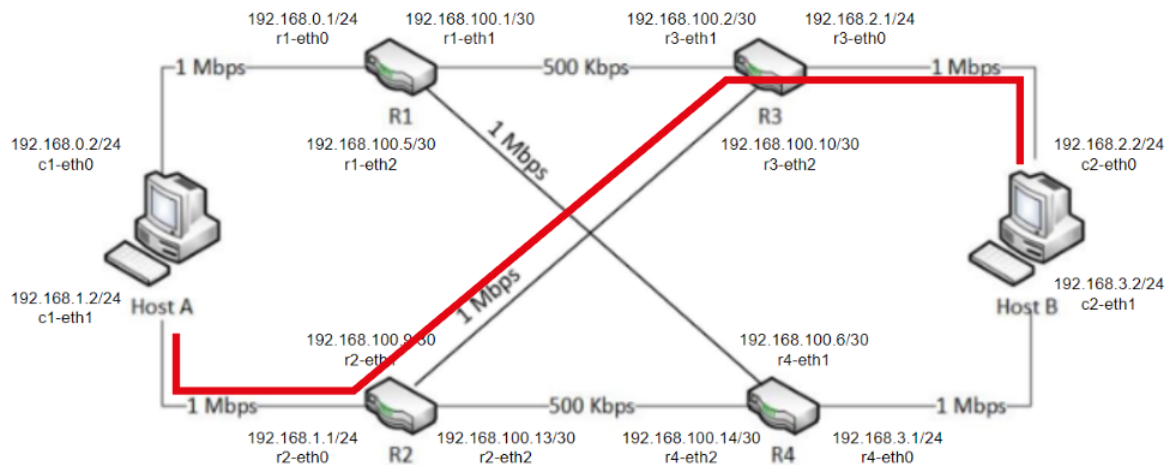
Gambar 3.2

Jalur Packet dari c1-eth0 ke 192.168.3.2

Dicontohkan lagi melakukan pengiriman paket dari c1-eth1 ke 192.168.3.2:

```
mininet> c1 traceroute -i c1-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)  0.044 ms  0.011 ms  0.008 ms
 2 192.168.100.10 (192.168.100.10)  0.022 ms  0.012 ms  0.012 ms
 3 192.168.2.2 (192.168.2.2)  0.024 ms  0.016 ms  0.018 ms
mininet>
```

Jika digambarkan pada topologi maka jalur yang digunakan:



Gambar 3.3

Jalur Packet dari c1-eth1 ke 192.168.3.2

Hasil catatan (log) c1 ping c2 dan pingall pada mininet:

```
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=0.161 ms
64 bytes from 192.168.2.2: icmp_seq=2 ttl=62 time=0.142 ms
64 bytes from 192.168.2.2: icmp_seq=3 ttl=62 time=0.126 ms

--- 192.168.2.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2014ms
rtt min/avg/max/mdev = 0.126/0.143/0.161/0.014 ms
mininet> pingall
*** Ping: testing ping reachability
c1 -> c2 r1 r2 r3 r4
c2 -> c1 r1 r2 r3 r4
r1 -> c1 c2 r2 r3 r4
r2 -> c1 c2 r1 r3 r4
r3 -> c1 c2 r1 r2 r4
r4 -> c1 c2 r1 r2 r3
*** Results: 0% dropped (30/30 received)
mininet>
```

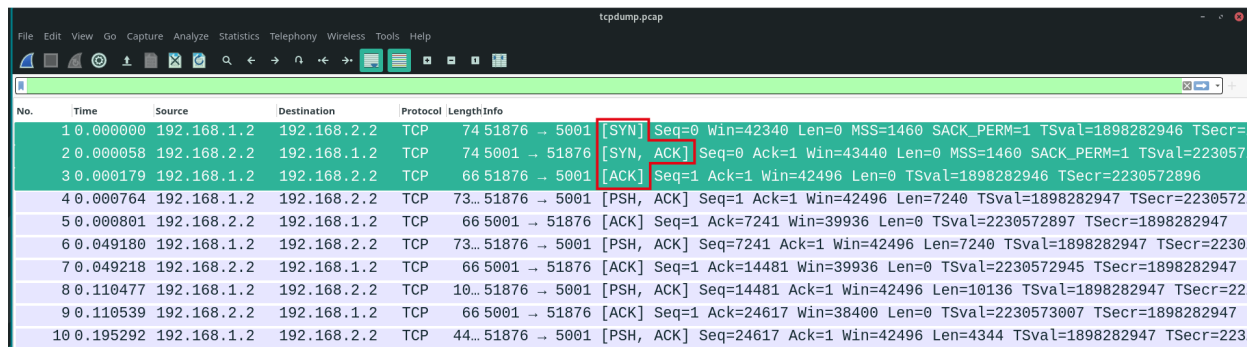

Dari hasil *traceroute*, *c1 ping c2*, dan *pingall* di atas, dapat terlihat bahwa paket-paket yang dikirimkan dapat sampai ke tujuan melalui jaringan atau rute yang berbeda-beda, sehingga dapat disimpulkan bahwa routing yang dilakukan sudah benar.

4. CLO 3 (TCP)

Pada CLO 3, saya diminta untuk membuktikan bahwa topologi ini dapat mengimplementasikan protokol TCP. Dengan memanfaatkan *iperf* yang digunakan untuk membuat trafik dan menggunakan *tcpdump* untuk menangkap trafik-trafik tersebut, kita dapat melihat apakah topologi ini dapat digunakan untuk protokol tersebut. Suatu protokol TCP dapat kita ketahui bahwa saat menginisialisasikan suatu koneksi akan ada paket dengan flag SYN yang dikirimkan dari IP A ke IP B, selanjutnya akan ada suatu paket balasan yang berisi flag SYN-ACK dari IP B ke IP A, dan yang terakhir akan ada paket balasan berisi flag ACK dari IP A ke IP B. Hal tersebut sangat sering ditemui dalam penginisialisasian protokol TCP yang biasa dikenal dengan nama *three-way-handshake*.

Untuk membuat trafik dan meng-*capture* trafik tersebut saya membuat kode pada “**ref code 4**”. Di bagian kode tersebut saya menyiapkan bahwa C2 sebagai *server* dan C1 sebagai *client*, selanjutnya trafik akan ditangkap oleh C2 dan akan dibuatkan file dengan format .pcap yang bisa diinspeksi melalui wireshark ataupun tcpdump untuk dilakukan analisa.

Setelah kode tersebut dijalankan maka kita bisa menginspeksi menggunakan wireshark agar lebih mudah dilihat flag yang ada pada trafik yang ditangkap.



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.1.2	192.168.2.2	TCP	74	51876 → 5001 [SYN] Seq=0 Win=42340 Len=0 MSS=1460 SACK_PERM=1 TSval=1898282946 TSecr=
2	0.000058	192.168.2.2	192.168.1.2	TCP	74	5001 → 51876 [SYN, ACK] Seq=0 Ack=1 Win=43440 Len=0 MSS=1460 SACK_PERM=1 TSval=223057
3	0.000179	192.168.1.2	192.168.2.2	TCP	66	51876 → 5001 [ACK] Seq=1 Ack=1 Win=42496 Len=0 TSval=1898282946 TSecr=2230572896
4	0.000764	192.168.1.2	192.168.2.2	TCP	73...	51876 → 5001 [PSH, ACK] Seq=1 Ack=1 Win=42496 Len=7240 TSval=1898282947 TSecr=2230572
5	0.000801	192.168.2.2	192.168.1.2	TCP	66	5001 → 51876 [ACK] Seq=1 Ack=7241 Win=39936 Len=0 TSval=2230572897 TSecr=1898282947
6	0.049180	192.168.1.2	192.168.2.2	TCP	73...	51876 → 5001 [PSH, ACK] Seq=7241 Ack=1 Win=42496 Len=7240 TSval=1898282947 TSecr=2230
7	0.049218	192.168.2.2	192.168.1.2	TCP	66	5001 → 51876 [ACK] Seq=1 Ack=14481 Win=39936 Len=0 TSval=2230572945 TSecr=1898282947
8	0.110477	192.168.1.2	192.168.2.2	TCP	10...	51876 → 5001 [PSH, ACK] Seq=14481 Ack=1 Win=42496 Len=10136 TSval=1898282947 TSecr=22
9	0.110539	192.168.2.2	192.168.1.2	TCP	66	5001 → 51876 [ACK] Seq=1 Ack=24617 Win=38400 Len=0 TSval=2230573007 TSecr=1898282947
10	0.195292	192.168.1.2	192.168.2.2	TCP	44...	51876 → 5001 [PSH, ACK] Seq=24617 Ack=1 Win=42496 Len=4344 TSval=1898282947 TSecr=223

Gambar 4.1

Paket yang tertangkap dengan menggunakan tcpdump dan diinspeksi dengan wireshark

Pada **Gambar 4.1** bisa kita lihat di kolom *Protocol* bahwa yang digunakan ialah TCP, selain itu seperti yang sudah dijelaskan sebelumnya bahwa terlihat ada *three-way-handshake*, hal tersebut bisa dilihat dari flag yang digunakan (difokuskan dengan garis berwarna merah) bahwa secara berurutan ada flag SYN dari IP 192.168.1.2 (A) ke 192.168.2.2 (B), selanjutnya ada flag SYN-ACK dari IP 192.168.2.2 (B) ke 192.168.1.2 (A), dan yang terakhir ada flag ACK dari IP 192.168.1.2 (A) ke 192.168.2.2 (B).

Dari beberapa hal tersebut dapat disimpulkan bahwa pembuatan dan konfigurasi topologi tersebut sudah siap digunakan dengan protokol TCP.

5. CLO 4 (Buffer Size)

Pada CLO ini saya diminta untuk melihat pengaruh buffer size yang dimulai dari buffer sebesar 20, 40, 60, dan 100. Untuk membangun topologi dengan buffer saya harus membuat topologi yang sedikit berbeda dengan “**ref code 1**” dengan menambahkan *max_queue_size* dan *use_htb=True* pada setiap *self.addLink* sehingga dapat dibuat topologi yang baru dengan kode pada “**ref code 5**”. Dapat terlihat pada bagian *addLink max_queue_size = max_queue*, dimana di dalam program ini saya membuat *max_queue* menjadi variabel sehingga lebih fleksibel untuk melakukan inspeksi terhadap berbagai macam besarnya buffer.

Untuk melakukan analisa *delay* saya menggunakan command *ping* dan data *time* sebagai parameter delay. Karena *time* pada command *ping* merupakan waktu yang dibutuhkan untuk menghubungi *host* yang dituju hingga kembali lagi ke *host* awal sehingga saya membuat waktu delay ialah:

$$Delay = \frac{time}{2}$$

Dengan asumsi jalur yang digunakan ketika mengirimkan sinyal *ping* kepada *host* yang dituju adalah sama dengan jalur yang digunakan untuk membalas sinyal *ping* tersebut.

Melihat keterhubungan delay terhadap besar *buffer* diperlukan untuk memadati jaringan tersebut dengan cara menggunakan *iperf*. Hal tersebut dapat terlihat pada bagian “**ref code 6**” dimana saya melakukan dua kali *iperf* pada *host* yang sama dengan *interface* yang berbeda dan melakukan pencatatan terhadap command ping ping.

Percobaan dilakukan hanya sekali dengan ukuran *buffer* yang berbeda-beda
Catatan untuk Buffer berukuran 20:

```
Kaenova Mahendra Auditama
C1 - runCLO4-c1 at 2021-06-13 18:33:36
with buffer 20
PING 192.168.2.2 (192.168.2.2) from 192.168.1.2 c1-eth1: 56(84) bytes of
data.
64 bytes from 192.168.2.2: icmp_seq=1 ttl=62 time=227 ms
64 bytes from 192.168.2.2: icmp_seq=4 ttl=62 time=313 ms
64 bytes from 192.168.2.2: icmp_seq=5 ttl=62 time=319 ms
64 bytes from 192.168.2.2: icmp_seq=6 ttl=62 time=314 ms
64 bytes from 192.168.2.2: icmp_seq=7 ttl=62 time=356 ms
64 bytes from 192.168.2.2: icmp_seq=8 ttl=62 time=375 ms
64 bytes from 192.168.2.2: icmp_seq=9 ttl=62 time=417 ms
64 bytes from 192.168.2.2: icmp_seq=11 ttl=62 time=233 ms
64 bytes from 192.168.2.2: icmp_seq=12 ttl=62 time=288 ms
```

```
64 bytes from 192.168.2.2: icmp_seq=13 ttl=62 time=318 ms
64 bytes from 192.168.2.2: icmp_seq=14 ttl=62 time=349 ms
64 bytes from 192.168.2.2: icmp_seq=15 ttl=62 time=403 ms
64 bytes from 192.168.2.2: icmp_seq=16 ttl=62 time=432 ms
64 bytes from 192.168.2.2: icmp_seq=18 ttl=62 time=335 ms
64 bytes from 192.168.2.2: icmp_seq=19 ttl=62 time=377 ms
64 bytes from 192.168.2.2: icmp_seq=20 ttl=62 time=345 ms
64 bytes from 192.168.2.2: icmp_seq=21 ttl=62 time=411 ms
64 bytes from 192.168.2.2: icmp_seq=22 ttl=62 time=452 ms
64 bytes from 192.168.2.2: icmp_seq=23 ttl=62 time=396 ms
64 bytes from 192.168.2.2: icmp_seq=24 ttl=62 time=303 ms
64 bytes from 192.168.2.2: icmp_seq=25 ttl=62 time=381 ms
64 bytes from 192.168.2.2: icmp_seq=26 ttl=62 time=386 ms
64 bytes from 192.168.2.2: icmp_seq=27 ttl=62 time=379 ms
64 bytes from 192.168.2.2: icmp_seq=28 ttl=62 time=420 ms
64 bytes from 192.168.2.2: icmp_seq=31 ttl=62 time=273 ms
64 bytes from 192.168.2.2: icmp_seq=32 ttl=62 time=339 ms
64 bytes from 192.168.2.2: icmp_seq=33 ttl=62 time=343 ms
64 bytes from 192.168.2.2: icmp_seq=34 ttl=62 time=385 ms
64 bytes from 192.168.2.2: icmp_seq=35 ttl=62 time=426 ms
64 bytes from 192.168.2.2: icmp_seq=36 ttl=62 time=454 ms
64 bytes from 192.168.2.2: icmp_seq=37 ttl=62 time=143 ms
64 bytes from 192.168.2.2: icmp_seq=38 ttl=62 time=0.151 ms
64 bytes from 192.168.2.2: icmp_seq=39 ttl=62 time=0.146 ms
```

Catatan untuk Buffer berukuran 40:

```
Kaenova Mahendra Auditama
C1 - runCL04-c1 at 2021-06-13 18:34:22
with buffer 40
PING 192.168.2.2 (192.168.2.2) from 192.168.1.2 c1-eth1: 56(84) bytes of
data.
64 bytes from 192.168.2.2: icmp_seq=1 ttl=62 time=36.0 ms
64 bytes from 192.168.2.2: icmp_seq=2 ttl=62 time=601 ms
64 bytes from 192.168.2.2: icmp_seq=3 ttl=62 time=864 ms
64 bytes from 192.168.2.2: icmp_seq=4 ttl=62 time=895 ms
64 bytes from 192.168.2.2: icmp_seq=5 ttl=62 time=853 ms
64 bytes from 192.168.2.2: icmp_seq=7 ttl=62 time=594 ms
64 bytes from 192.168.2.2: icmp_seq=8 ttl=62 time=588 ms
64 bytes from 192.168.2.2: icmp_seq=9 ttl=62 time=655 ms
64 bytes from 192.168.2.2: icmp_seq=10 ttl=62 time=662 ms
64 bytes from 192.168.2.2: icmp_seq=11 ttl=62 time=655 ms
64 bytes from 192.168.2.2: icmp_seq=12 ttl=62 time=710 ms
64 bytes from 192.168.2.2: icmp_seq=13 ttl=62 time=850 ms
64 bytes from 192.168.2.2: icmp_seq=15 ttl=62 time=811 ms
64 bytes from 192.168.2.2: icmp_seq=17 ttl=62 time=653 ms
64 bytes from 192.168.2.2: icmp_seq=18 ttl=62 time=647 ms
```

```
64 bytes from 192.168.2.2: icmp_seq=19 ttl=62 time=653 ms
64 bytes from 192.168.2.2: icmp_seq=20 ttl=62 time=696 ms
64 bytes from 192.168.2.2: icmp_seq=21 ttl=62 time=690 ms
64 bytes from 192.168.2.2: icmp_seq=22 ttl=62 time=709 ms
64 bytes from 192.168.2.2: icmp_seq=23 ttl=62 time=800 ms
64 bytes from 192.168.2.2: icmp_seq=24 ttl=62 time=940 ms
64 bytes from 192.168.2.2: icmp_seq=27 ttl=62 time=601 ms
64 bytes from 192.168.2.2: icmp_seq=28 ttl=62 time=523 ms
64 bytes from 192.168.2.2: icmp_seq=29 ttl=62 time=528 ms
64 bytes from 192.168.2.2: icmp_seq=30 ttl=62 time=571 ms
64 bytes from 192.168.2.2: icmp_seq=31 ttl=62 time=576 ms
64 bytes from 192.168.2.2: icmp_seq=32 ttl=62 time=570 ms
64 bytes from 192.168.2.2: icmp_seq=33 ttl=62 time=698 ms
64 bytes from 192.168.2.2: icmp_seq=34 ttl=62 time=826 ms
64 bytes from 192.168.2.2: icmp_seq=39 ttl=62 time=333 ms
64 bytes from 192.168.2.2: icmp_seq=40 ttl=62 time=0.134 ms
64 bytes from 192.168.2.2: icmp_seq=41 ttl=62 time=0.173 ms
64 bytes from 192.168.2.2: icmp_seq=42 ttl=62 time=0.128 ms
```

Catatan untuk Buffer berukuran 60:

```
Kaenova Mahendra Auditama
C1 - runCLO4-c1 at 2021-06-13 18:35:31
with buffer 60
PING 192.168.2.2 (192.168.2.2) from 192.168.1.2 c1-eth1: 56(84) bytes of
data.
64 bytes from 192.168.2.2: icmp_seq=1 ttl=62 time=45.3 ms
64 bytes from 192.168.2.2: icmp_seq=2 ttl=62 time=581 ms
64 bytes from 192.168.2.2: icmp_seq=3 ttl=62 time=871 ms
64 bytes from 192.168.2.2: icmp_seq=4 ttl=62 time=974 ms
64 bytes from 192.168.2.2: icmp_seq=5 ttl=62 time=1090 ms
64 bytes from 192.168.2.2: icmp_seq=6 ttl=62 time=1204 ms
64 bytes from 192.168.2.2: icmp_seq=7 ttl=62 time=1319 ms
64 bytes from 192.168.2.2: icmp_seq=8 ttl=62 time=1385 ms
64 bytes from 192.168.2.2: icmp_seq=9 ttl=62 time=1184 ms
64 bytes from 192.168.2.2: icmp_seq=11 ttl=62 time=1012 ms
64 bytes from 192.168.2.2: icmp_seq=12 ttl=62 time=897 ms
64 bytes from 192.168.2.2: icmp_seq=13 ttl=62 time=989 ms
64 bytes from 192.168.2.2: icmp_seq=14 ttl=62 time=1055 ms
64 bytes from 192.168.2.2: icmp_seq=17 ttl=62 time=1022 ms
64 bytes from 192.168.2.2: icmp_seq=18 ttl=62 time=1075 ms
64 bytes from 192.168.2.2: icmp_seq=19 ttl=62 time=1201 ms
64 bytes from 192.168.2.2: icmp_seq=24 ttl=62 time=1079 ms
64 bytes from 192.168.2.2: icmp_seq=25 ttl=62 time=963 ms
64 bytes from 192.168.2.2: icmp_seq=26 ttl=62 time=909 ms
64 bytes from 192.168.2.2: icmp_seq=27 ttl=62 time=904 ms
64 bytes from 192.168.2.2: icmp_seq=28 ttl=62 time=995 ms
```

```
64 bytes from 192.168.2.2: icmp_seq=29 ttl=62 time=1050 ms
64 bytes from 192.168.2.2: icmp_seq=30 ttl=62 time=1065 ms
64 bytes from 192.168.2.2: icmp_seq=31 ttl=62 time=1094 ms
64 bytes from 192.168.2.2: icmp_seq=32 ttl=62 time=1208 ms
64 bytes from 192.168.2.2: icmp_seq=33 ttl=62 time=1407 ms
64 bytes from 192.168.2.2: icmp_seq=34 ttl=62 time=1436 ms
64 bytes from 192.168.2.2: icmp_seq=36 ttl=62 time=1093 ms
64 bytes from 192.168.2.2: icmp_seq=37 ttl=62 time=1126 ms
64 bytes from 192.168.2.2: icmp_seq=39 ttl=62 time=928 ms
64 bytes from 192.168.2.2: icmp_seq=40 ttl=62 time=959 ms
64 bytes from 192.168.2.2: icmp_seq=41 ttl=62 time=892 ms
64 bytes from 192.168.2.2: icmp_seq=42 ttl=62 time=343 ms
64 bytes from 192.168.2.2: icmp_seq=43 ttl=62 time=0.134 ms
```

Catatan untuk Buffer berukuran 100:

```
Kaenova Mahendra Auditama
C1 - runCLO4-c1 at 2021-06-13 18:36:28
with buffer 100
PING 192.168.2.2 (192.168.2.2) from 192.168.1.2 c1-eth1: 56(84) bytes of
data.
64 bytes from 192.168.2.2: icmp_seq=1 ttl=62 time=45.4 ms
64 bytes from 192.168.2.2: icmp_seq=2 ttl=62 time=561 ms
64 bytes from 192.168.2.2: icmp_seq=3 ttl=62 time=872 ms
64 bytes from 192.168.2.2: icmp_seq=4 ttl=62 time=964 ms
64 bytes from 192.168.2.2: icmp_seq=5 ttl=62 time=1043 ms
64 bytes from 192.168.2.2: icmp_seq=6 ttl=62 time=1173 ms
64 bytes from 192.168.2.2: icmp_seq=7 ttl=62 time=1360 ms
64 bytes from 192.168.2.2: icmp_seq=8 ttl=62 time=1668 ms
64 bytes from 192.168.2.2: icmp_seq=9 ttl=62 time=2121 ms
64 bytes from 192.168.2.2: icmp_seq=10 ttl=62 time=2345 ms
64 bytes from 192.168.2.2: icmp_seq=11 ttl=62 time=2333 ms
64 bytes from 192.168.2.2: icmp_seq=17 ttl=62 time=1680 ms
64 bytes from 192.168.2.2: icmp_seq=18 ttl=62 time=1724 ms
64 bytes from 192.168.2.2: icmp_seq=19 ttl=62 time=1767 ms
64 bytes from 192.168.2.2: icmp_seq=21 ttl=62 time=1695 ms
64 bytes from 192.168.2.2: icmp_seq=22 ttl=62 time=1603 ms
64 bytes from 192.168.2.2: icmp_seq=23 ttl=62 time=1597 ms
64 bytes from 192.168.2.2: icmp_seq=24 ttl=62 time=1591 ms
64 bytes from 192.168.2.2: icmp_seq=25 ttl=62 time=1645 ms
64 bytes from 192.168.2.2: icmp_seq=26 ttl=62 time=1736 ms
64 bytes from 192.168.2.2: icmp_seq=27 ttl=62 time=1779 ms
64 bytes from 192.168.2.2: icmp_seq=28 ttl=62 time=1809 ms
64 bytes from 192.168.2.2: icmp_seq=29 ttl=62 time=1901 ms
64 bytes from 192.168.2.2: icmp_seq=30 ttl=62 time=1956 ms
64 bytes from 192.168.2.2: icmp_seq=31 ttl=62 time=2144 ms
64 bytes from 192.168.2.2: icmp_seq=32 ttl=62 time=2356 ms
```

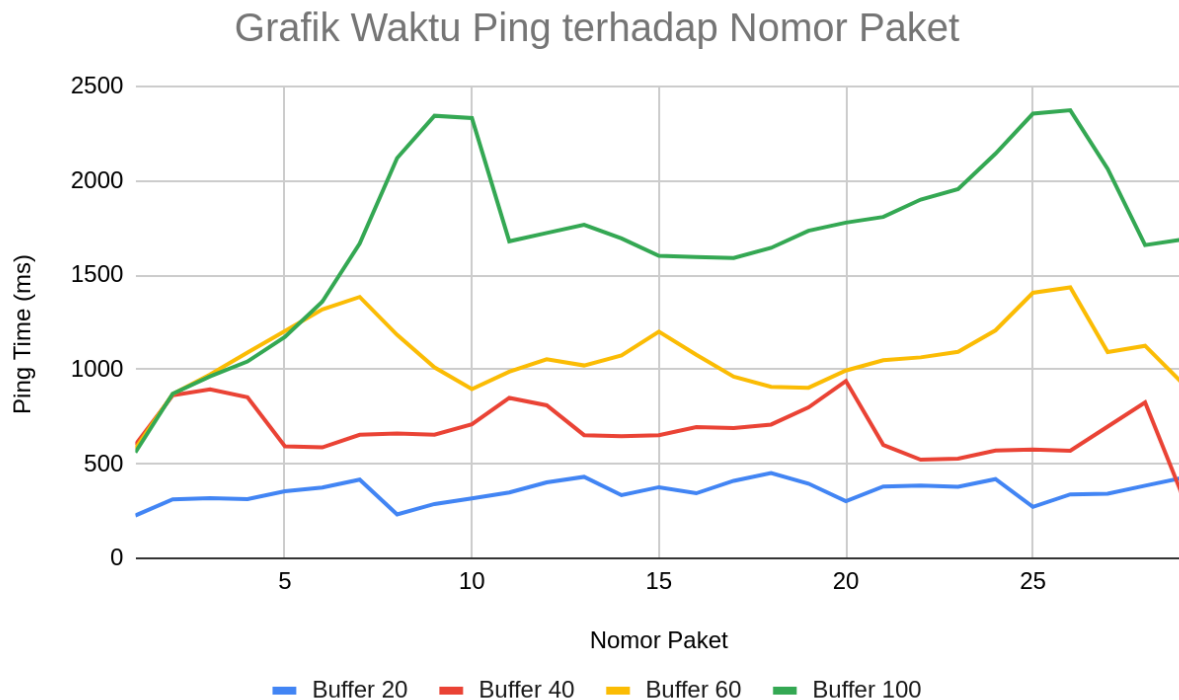
```
64 bytes from 192.168.2.2: icmp_seq=34 ttl=62 time=2374 ms
64 bytes from 192.168.2.2: icmp_seq=37 ttl=62 time=2064 ms
64 bytes from 192.168.2.2: icmp_seq=39 ttl=62 time=1660 ms
64 bytes from 192.168.2.2: icmp_seq=40 ttl=62 time=1689 ms
64 bytes from 192.168.2.2: icmp_seq=41 ttl=62 time=1756 ms
64 bytes from 192.168.2.2: icmp_seq=42 ttl=62 time=1617 ms
64 bytes from 192.168.2.2: icmp_seq=43 ttl=62 time=1684 ms
64 bytes from 192.168.2.2: icmp_seq=44 ttl=62 time=1702 ms
64 bytes from 192.168.2.2: icmp_seq=45 ttl=62 time=1624 ms
64 bytes from 192.168.2.2: icmp_seq=46 ttl=62 time=1619 ms
64 bytes from 192.168.2.2: icmp_seq=47 ttl=62 time=1649 ms
64 bytes from 192.168.2.2: icmp_seq=48 ttl=62 time=1680 ms
64 bytes from 192.168.2.2: icmp_seq=49 ttl=62 time=1722 ms
64 bytes from 192.168.2.2: icmp_seq=50 ttl=62 time=1729 ms
64 bytes from 192.168.2.2: icmp_seq=51 ttl=62 time=1808 ms
64 bytes from 192.168.2.2: icmp_seq=52 ttl=62 time=1875 ms
64 bytes from 192.168.2.2: icmp_seq=53 ttl=62 time=1452 ms
64 bytes from 192.168.2.2: icmp_seq=54 ttl=62 time=950 ms
64 bytes from 192.168.2.2: icmp_seq=55 ttl=62 time=473 ms
64 bytes from 192.168.2.2: icmp_seq=56 ttl=62 time=0.148 ms
64 bytes from 192.168.2.2: icmp_seq=57 ttl=62 time=0.160 ms
```

Disini saya tidak menganggap time yang di bawah 100 ms dikarenakan ketika itu trafik yang dibuat dengan *iperf* sudah atau sedang tidak berjalan, sehingga mengakibatkan kekosongan pada jaringan sehingga buffer akan kosong. Setelah itu, saya juga menormalkan jumlah data dengan mencari jumlah data dengan jumlah *ping* paling sedikit lalu disamaratakan. Pada kasus ini jumlah data *ping* yang paling sedikit ialah catatan dengan *buffer*.

Hasil pengolahan data ping:

Ukuran Buffer			
20	40	60	100
Time (ms)			
227	601	581	561
313	864	871	872
319	895	974	964
314	853	1090	1043
356	594	1204	1173
375	588	1319	1360
417	655	1385	1668
233	662	1184	2121
288	655	1012	2345
318	710	897	2333
349	850	989	1680
403	811	1055	1724
432	653	1022	1767
335	647	1075	1695
377	653	1201	1603
345	696	1079	1597
411	690	963	1591
452	709	909	1645
396	800	904	1736
303	940	995	1779
381	601	1050	1809
386	523	1065	1901
379	528	1094	1956
420	571	1208	2144
273	576	1407	2356
339	570	1436	2374
343	698	1093	2064
385	826	1126	1660
426	333	928	1689
Rata-Rata			
355	681.1034483	1072.965517	1696.896552

Dari tabel di samping dapat terlihat rata-rata dari setiap ukuran buffer memiliki hasil yang berbeda-beda. Didapatkan bahwa buffer 20 memiliki rata-rata waktu terendah dibandingkan dengan yang lainnya dengan sebesar 355 dimana yang artinya delaynya ialah 177.5. Pada ukuran buffer 40 memiliki rata-rata delay 340,552, ukuran buffer 60 memiliki rata-rata delay 536,483, ukuran buffer 100 memiliki rata-rata delay 848,448.



Gambar 5.1

Grafik Waktu Ping terhadap Nomor Paket dengan Buffer yang Berbeda

Dari **Gambar 5.1** dapat terlihat juga bahwa ping dengan *buffer* 20 relatif lebih rendah dibandingkan dengan *buffer* yang lebih besar dibandingkan dari 20. Selain itu juga terlihat bahwa dengan *buffer* 20, waktu ping terlihat lebih stabil, dibandingkan dengan *buffer* sebesar 100. Berdasarkan data-data tersebut dapat terlihat bahwa dengan *buffer* 20 ping relatif kecil dan terlihat lebih stabil dibandingkan dengan *buffer* yang berukuran lebih besar sehingga dapat disimpulkan bahwa delay dengan *buffer* 20 lebih stabil dan lebih kecil.

REFERENSI CODE

Source code program dapat dilakukan *git clone* terhadap *repository github* di bawah ini atau mendownload melalui *Google Drive* melalui link di bawah ini. Untuk instruksi penggunaan diharapkan membaca README.txt atau README.md

Git Clone:

https://github.com/kaenova/Jarkom_Tubes

Google Drive:

https://bit.ly/Jarkom_Tubes

ref code 1

```
# ref code 1
# Membuat Topologi untuk CLO1, CLO2, CLO3
class MyTopo(Topo):
    #mendeklarasikan def/fungsi untuk membangun topologi
    def __init__(self, **opts):
        Topo.__init__( self, **opts)
        #Add host and switch
        linkopt = {'delay' : '0ms', 'loss' : 0}
        r1 =self.addHost("r1", cls=LinuxRouter)
        r2 =self.addHost("r2", cls=LinuxRouter)
        r3 =self.addHost("r3", cls=LinuxRouter)
        r4 =self.addHost("r4", cls=LinuxRouter)

        c1 = self.addHost("c1", cls=Host)
        c2 = self.addHost("c2", cls=Host)

        self.addLink(c1, r1, bw=1, cls=TCLink, **linkopt)
        self.addLink(c1, r2, bw=1, cls=TCLink, **linkopt)
        ## C2 to Router
        self.addLink(c2, r3, bw=1, cls=TCLink, **linkopt)
        self.addLink(c2, r4, bw=1, cls=TCLink, **linkopt)
        ## Router to Router
        self.addLink(r1, r3, bw=0.5, cls=TCLink, **linkopt)
        self.addLink(r1, r4, bw=1, cls=TCLink, **linkopt)
        self.addLink(r2, r3, bw=1, cls=TCLink, **linkopt)
        self.addLink(r2, r4, bw=0.5, cls=TCLink, **linkopt)
```

ref code 2

```
# ref code 2
# Konfigurasi IP Address di C1
c1.cmd("ifconfig c1-eth0 192.168.0.2/24")
c1.cmd("route add default gw 192.168.0.1 c1-eth0")
c1.cmd("ifconfig c1-eth1 192.168.1.2/24")
c1.cmd("route add default gw 192.168.1.1 c1-eth1")

# Konfigurasi IP Address di C2
c2.cmd("ifconfig c2-eth0 192.168.2.2/24")
c2.cmd("route add default gw 192.168.2.1 c2-eth0")
c2.cmd("ifconfig c2-eth1 192.168.3.2/24")
c2.cmd("route add default gw 192.168.3.1 c2-eth1")

# Konfigurasi IP Address di R1
r1.cmd("ifconfig r1-eth0 192.168.0.1/24")
r1.cmd("ifconfig r1-eth1 192.168.100.1/30")
```

```

r1.cmd("ifconfig r1-eth2 192.168.100.5/30")

# Konfigurasi IP Address di R2
r2.cmd("ifconfig r2-eth0 192.168.1.1/24")
r2.cmd("ifconfig r2-eth1 192.168.100.9/30")
r2.cmd("ifconfig r2-eth2 192.168.100.13/30")

# Konfigurasi IP Address di R3
r3.cmd("ifconfig r3-eth0 192.168.2.1/24")
r3.cmd("ifconfig r3-eth1 192.168.100.2/30")
r3.cmd("ifconfig r3-eth2 192.168.100.10/30")

# Konfigurasi IP Address di R4
r4.cmd("ifconfig r4-eth0 192.168.3.1/24")
r4.cmd("ifconfig r4-eth1 192.168.100.6/30")
r4.cmd("ifconfig r4-eth2 192.168.100.14/30")

```

ref code 3

```

# ref code 3
# Konfigurasi C1
c1.cmd("ifconfig c1-eth0 192.168.0.2/24")
c1.cmd("ifconfig c1-eth1 192.168.1.2/24")
c1.cmd("ip rule add from 192.168.0.2 table 1")
c1.cmd("ip rule add from 192.168.1.2 table 2")
c1.cmd("ip route add default via 192.168.0.1 dev c1-eth0 table 1")
c1.cmd("ip route add default via 192.168.1.1 dev c1-eth1 table 2")
c1.cmd("ip route add 192.168.0.0/24 dev c1-eth0 scope link table 1")
c1.cmd("ip route add 192.168.1.0/24 dev c1-eth1 scope link table 2")
c1.cmd("ip route add default scope global nexthop via 192.168.0.1 dev c1-eth0")
c1.cmd("ip route add default scope global nexthop via 192.168.1.1 dev c1-eth1")
c1.cmd("route add default gw 192.168.1.1 dev c1-eth1")
c1.cmd("route add default gw 192.168.0.1 dev c1-eth0")

# Konfigurasi C2
c2.cmd("ifconfig c2-eth0 192.168.2.2/24")
c2.cmd("ifconfig c2-eth1 192.168.3.2/24")
c2.cmd("ip rule add from 192.168.2.2 table 1")
c2.cmd("ip rule add from 192.168.3.2 table 2")
c2.cmd("ip route add default via 192.168.2.1 dev c2-eth0 table 1")
c2.cmd("ip route add default via 192.168.3.1 dev c2-eth1 table 2")
c2.cmd("ip route add 192.168.2.0/24 dev c2-eth0 scope link table 1")
c2.cmd("ip route add 192.168.3.0/24 dev c2-eth1 scope link table 2")
c2.cmd("ip route add default scope global nexthop via 192.168.3.1 dev c2-eth1")
c2.cmd("ip route add default scope global nexthop via 192.168.2.1 dev c2-eth0")
c2.cmd("route add default gw 192.168.3.1 dev c2-eth1")

```

```
c2.cmd("route add default gw 192.168.2.1 dev c2-eth0")

# Konfigurasi R1
r1.cmd("ifconfig r1-eth0 192.168.0.1/24")
r1.cmd("ifconfig r1-eth1 192.168.100.1/30")
r1.cmd("ifconfig r1-eth2 192.168.100.5/30")
r1.cmd("sysctl net.ipv4.ip_forward=1")
# Routing R1
r1.cmd("route add -net 192.168.2.0/24 gw 192.168.100.2")
r1.cmd("route add -net 192.168.3.0/24 gw 192.168.100.6")
r1.cmd("route add -net 192.168.1.0/24 gw 192.168.100.6")
r1.cmd("route add -net 192.168.100.8/30 gw 192.168.100.2")
r1.cmd("route add -net 192.168.100.12/30 gw 192.168.100.6")

# Konfigurasi R3
r3.cmd("ifconfig r3-eth0 192.168.2.1/24")
r3.cmd("ifconfig r3-eth1 192.168.100.2/30")
r3.cmd("ifconfig r3-eth2 192.168.100.10/30")
r3.cmd("sysctl net.ipv4.ip_forward=1")
# Routing R3
r3.cmd("route add -net 192.168.0.0/24 gw 192.168.100.1")
r3.cmd("route add -net 192.168.1.0/24 gw 192.168.100.9")
r3.cmd("route add -net 192.168.3.0/24 gw 192.168.100.9")
r3.cmd("route add -net 192.168.100.4/30 gw 192.168.100.1")
r3.cmd("route add -net 192.168.100.12/30 gw 192.168.100.9")

# Konfigurasi R2
r2.cmd("ifconfig r2-eth0 192.168.1.1/24")
r2.cmd("ifconfig r2-eth1 192.168.100.9/30")
r2.cmd("ifconfig r2-eth2 192.168.100.13/30")
r2.cmd("sysctl net.ipv4.ip_forward=1")
# Routing R2
r2.cmd("route add -net 192.168.0.0/24 gw 192.168.100.10")
r2.cmd("route add -net 192.168.2.0/24 gw 192.168.100.10")
r2.cmd("route add -net 192.168.3.0/24 gw 192.168.100.14")
r2.cmd("route add -net 192.168.100.4/30 gw 192.168.100.14")
r2.cmd("route add -net 192.168.100.0/39 gw 192.168.100.10")

# Konfigurasi R4
r4.cmd("ifconfig r4-eth0 192.168.3.1/24")
r4.cmd("ifconfig r4-eth1 192.168.100.6/30")
r4.cmd("ifconfig r4-eth2 192.168.100.14/30")
r4.cmd("sysctl net.ipv4.ip_forward=1")
# Routing R4
r4.cmd("route add -net 192.168.0.0/24 gw 192.168.100.5")
r4.cmd("route add -net 192.168.1.0/24 gw 192.168.100.13")
```

```
r4.cmd("route add -net 192.168.2.0/24 gw 192.168.100.5")
r4.cmd("route add -net 192.168.100.8/30 gw 192.168.100.13")
r4.cmd("route add -net 192.168.100.0/30 gw 192.168.100.5")
```

ref code 4

```
# ref code 4
# Server iperf C2
c2.cmd("iperf -s &")
# Setting up tcpdump
c2.cmd("nohup tcpdump -c 10 -i c2-eth0 -w {}/tcpdump.pcap
tcp&".format(logs_path))
time.sleep(1)
# iperfing
c1.cmd("iperf -t 5 -c 192.168.2.2 &")
print("currently generating traffic, please wait for 5 second")
time.sleep(5)
c1.cmdPrint("tcpdump -r {}/tcpdump.pcap".format(logs_path))
```

ref code 5

```
class MyTopoWithBuffer(Topo):
    #mendeklarasikan def/fungsi untuk membangun topologi
    def __init__(self, max_queue, **opts):
        Topo.__init__( self, **opts)
        #Add host and switch
        linkopt = {'delay' : '0ms', 'loss' : 0}
        r1 =self.addHost("r1", cls=LinuxRouter)
        r2 =self.addHost("r2", cls=LinuxRouter)
        r3 =self.addHost("r3", cls=LinuxRouter)
        r4 =self.addHost("r4", cls=LinuxRouter)

        c1 = self.addHost("c1", cls=Host)
        c2 = self.addHost("c2", cls=Host)

        self.addLink(c1, r1, max_queue_size=max_queue, use_htb=True, bw=1,
cls=TCLink, **linkopt)
        self.addLink(c1, r2, max_queue_size=max_queue, use_htb=True, bw=1,
cls=TCLink, **linkopt)
        ## C2 to Router
        self.addLink(c2, r3, max_queue_size=max_queue,use_htb=True, bw=1,
cls=TCLink, **linkopt)
        self.addLink(c2, r4, max_queue_size=max_queue,use_htb=True, bw=1,
cls=TCLink, **linkopt)
        ## Router to Router
        self.addLink(r1, r3, max_queue_size=max_queue, use_htb=True, bw=0.5,
```

```

cls=TCLink, **linkopt)
    self.addLink(r1, r4, max_queue_size=max_queue, use_htb=True, bw=1,
cls=TCLink, **linkopt)
    self.addLink(r2, r3, max_queue_size=max_queue, use_htb=True, bw=1,
cls=TCLink, **linkopt)
    self.addLink(r2, r4, max_queue_size=max_queue, use_htb=True, bw=0.5,
cls=TCLink, **linkopt)

```

ref code 6

```

# ref code 6
# Setting up traffic
c2.cmd("iperf -s &")
c1.cmd("iperf -t 30 -B 192.168.0.2 -c 192.168.2.2 &")
c1.cmd("iperf -t 30 -B 192.168.1.2 -c 192.168.2.2 &")

# set Computer 1 as iperf client
c1.cmd('echo "Kaenova Mahendra Auditama \n C1 - runCLO4-c1 at {} \n with buffer
{}" > {}/runCLO4-c1-{}-iperf.txt && ping -I c1-eth1 192.168.2.2 >>
{}/runCLO4-c1-{}-iperf.txt && echo "\n Finish" >> {}/runCLO4-c1-{}-iperf.txt
&'.format(f"{currDateStr} {currTimeStr}",buffer_size, logs_path, buffer_size,
logs_path, buffer_size, logs_path, buffer_size))
info('\n')
print("Currently generating ping log file in the background")
print("log file is created at {}".format(logs_path))

```