

"به نام خدا"



گزارش کار اول

آزمایش 1 : آشنایی با Mininet

اعضای گروه: ستاره باباجانی، زهرا سادات طباطبائی

نیم سال دوم 1402-1403

## سوال اول:

ابتدا به 5 مقصد مختلف ping میکنیم.

```
mininet@mininet-vm:~$ ping -c 5 www.aparat.com
PING www.aparat.com (185.147.178.11) 56(84) bytes of data.
64 bytes from 185.147.178.11 (185.147.178.11): icmp_seq=1 ttl=51 time=61.5 ms
64 bytes from 185.147.178.11 (185.147.178.11): icmp_seq=2 ttl=51 time=56.0 ms
64 bytes from 185.147.178.11 (185.147.178.11): icmp_seq=3 ttl=51 time=55.3 ms
64 bytes from 185.147.178.11 (185.147.178.11): icmp_seq=4 ttl=51 time=51.4 ms
64 bytes from 185.147.178.11 (185.147.178.11): icmp_seq=5 ttl=51 time=52.5 ms
```

```
--- www.aparat.com ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4005ms
rtt min/avg/max/mdev = 51.405/55.350/61.526/3.524 ms
mininet@mininet-vm:~$
```

```
mininet@mininet-vm:~$ ping -c 5 yahoo.com
PING yahoo.com (74.6.231.20) 56(84) bytes of data.
64 bytes from media-router-fp73.prod.media.vip.ne1.yahoo.com (74.6.231.20): icmp_seq=1 ttl=43 time=233 ms
64 bytes from media-router-fp73.prod.media.vip.ne1.yahoo.com (74.6.231.20): icmp_seq=2 ttl=43 time=287 ms
64 bytes from media-router-fp73.prod.media.vip.ne1.yahoo.com (74.6.231.20): icmp_seq=3 ttl=43 time=285 ms
64 bytes from media-router-fp73.prod.media.vip.ne1.yahoo.com (74.6.231.20): icmp_seq=4 ttl=43 time=236 ms
64 bytes from media-router-fp73.prod.media.vip.ne1.yahoo.com (74.6.231.20): icmp_seq=5 ttl=43 time=325 ms
```

```
--- yahoo.com ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4003ms
rtt min/avg/max/mdev = 232.751/273.262/325.042/34.887 ms
mininet@mininet-vm:~$
```

```
mininet@mininet-vm:~$ ping -c 5 github.com
PING github.com (140.82.121.4) 56(84) bytes of data.
64 bytes from lb-140-82-121-4-fra.github.com (140.82.121.4): icmp_seq=1 ttl=45 time=138 ms
64 bytes from lb-140-82-121-4-fra.github.com (140.82.121.4): icmp_seq=2 ttl=45 time=157 ms
64 bytes from lb-140-82-121-4-fra.github.com (140.82.121.4): icmp_seq=3 ttl=45 time=187 ms
64 bytes from lb-140-82-121-4-fra.github.com (140.82.121.4): icmp_seq=4 ttl=45 time=201 ms
64 bytes from lb-140-82-121-4-fra.github.com (140.82.121.4): icmp_seq=5 ttl=45 time=225 ms
```

```
--- github.com ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4005ms
rtt min/avg/max/mdev = 137.935/181.716/224.648/30.864 ms
mininet@mininet-vm:~$
```

```
mininet@mininet-vm:~$ ping -c 5 linkedin.com
PING linkedin.com (13.107.42.14) 56(84) bytes of data.
64 bytes from 13.107.42.14 (13.107.42.14): icmp_seq=1 ttl=113 time=95.9 ms
64 bytes from 13.107.42.14 (13.107.42.14): icmp_seq=2 ttl=113 time=99.8 ms
64 bytes from 13.107.42.14 (13.107.42.14): icmp_seq=3 ttl=113 time=62.6 ms
64 bytes from 13.107.42.14 (13.107.42.14): icmp_seq=4 ttl=113 time=82.6 ms
64 bytes from 13.107.42.14 (13.107.42.14): icmp_seq=5 ttl=113 time=83.8 ms
```

```
--- linkedin.com ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4008ms
rtt min/avg/max/mdev = 62.603/84.947/99.837/13.020 ms
mininet@mininet-vm:~$
```

```

mininet@mininet-vm:~$ ping -c 5 isna.ir
PING isna.ir (185.143.233.3) 56(84) bytes of data:
64 bytes from 185.143.233.3 (185.143.233.3): icmp_seq=1 ttl=51 time=65.0 ms
64 bytes from 185.143.233.3 (185.143.233.3): icmp_seq=2 ttl=51 time=41.7 ms
64 bytes from 185.143.233.3 (185.143.233.3): icmp_seq=3 ttl=51 time=101 ms
64 bytes from 185.143.233.3 (185.143.233.3): icmp_seq=4 ttl=51 time=69.8 ms
64 bytes from 185.143.233.3 (185.143.233.3): icmp_seq=5 ttl=51 time=87.3 ms

--- isna.ir ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4010ms
rtt min/avg/max/mdev = 41.713/72.974/101.073/20.229 ms
mininet@mininet-vm:~$ _

```

مفاهیم RTT (Round-Trip Time) و TTL (Time-To-Live) دو ویژگی مهم هستند که در ارتباط با ارسال داده‌ها در شبکه‌ها استفاده می‌شوند.

1. RTT (Round-Trip Time): مقدار زمانی است که طول می‌کشد تا یک بیت از یک دستگاه به مقصد ارسال شود و به دستگاه فرستنده بازگردد. به عبارت دیگر، این مقدار زمانی است که یک پیام از یک کامپیوتر به مقصد ارسال می‌شود و پس از پیام به فرستنده برمی‌گردد. این معیار به عنوان یکی از اصطلاحات مهم در اندازه‌گیری کیفیت ارتباط در شبکه‌ها مورد استفاده قرار می‌گیرد. زمانی که RTT کم باشد، ارتباط بین دو دستگاه سریع تر است. در کل RTT مجموع زمان رفتن پکت و بازگشت Ack است.
2. TTL (Time-To-Live): مقدار عددی است که به پکت‌های داده اضافه می‌شود و نشان‌دهنده تعداد هاپ‌هایی است که یک پکت می‌تواند در مسیر از دستگاه فرستنده به دستگاه مقصد عبور کند. هر بار که یک پکت از یک هاپ عبور می‌کند، مقدار TTL یک واحد کم می‌شود. اگر مقدار TTL به صفر برسد، پکت از دست می‌رود و دریافت نمی‌شود. این ویژگی معمولاً برای جلوگیری از حلقه‌های بی‌نهایت در شبکه‌ها استفاده می‌شود.

بین RTT و TTL رابطه مستقیمی وجود ندارد. یعنی مقدار RTT و TTL به صورت مستقل از یکدیگر محاسبه و استفاده می‌شوند. در کل میتوان گفت هر چه RTT بیشتر باشد TTL کمتر است. تنظیم مقدار TTL مهم است تا جلوگیری از حلقه‌های بی‌نهایت و از دست رفتن پکت‌ها در شبکه، در حالی که مقدار RTT معمولاً برای اندازه‌گیری کیفیت ارتباط در شبکه‌ها استفاده می‌شود.

## سوال دوم:

ابتدا آدرس کامپیوتر خود را بدست می آوریم:

```
C:\Users\ASUS>ipconfig

Windows IP Configuration

Ethernet adapter vEthernet (Default Switch):

    Connection-specific DNS Suffix  . : 
    Link-local IPv6 Address . . . . . : fe80::8d6a:6a01:40f4:f856%35
    IPv4 Address. . . . . : 172.28.144.1
    Subnet Mask . . . . . : 255.255.240.0
    Default Gateway . . . . . : 

Ethernet adapter Ethernet:

    Connection-specific DNS Suffix  . : 
    Link-local IPv6 Address . . . . . : fe80::b8b7:64da:cdb0:57a3%43
    IPv4 Address. . . . . : 192.168.56.1
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 

Wireless LAN adapter Local Area Connection* 1:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . : 

Wireless LAN adapter Local Area Connection* 10:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . : 

Wireless LAN adapter Wi-Fi:

    Connection-specific DNS Suffix  . : 
    Link-local IPv6 Address . . . . . : fe80::a1b1:65d9:1db:5a84%13
    IPv4 Address. . . . . : 192.168.25.115
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 192.168.25.152

Ethernet adapter Bluetooth Network Connection:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . : 

C:\Users\ASUS>
```

حال با دستور زیر از ماشین مجازی به مقصد کامپیوتر پینگ میکنیم:

```
mininet@mininet-vm:~$ ping 192.168.25.115 -c 5
PING 192.168.25.115 (192.168.25.115) 56(84) bytes of data.
64 bytes from 192.168.25.115: icmp_seq=1 ttl=127 time=1.89 ms
64 bytes from 192.168.25.115: icmp_seq=2 ttl=127 time=3.27 ms
64 bytes from 192.168.25.115: icmp_seq=3 ttl=127 time=2.99 ms
64 bytes from 192.168.25.115: icmp_seq=4 ttl=127 time=3.05 ms
64 bytes from 192.168.25.115: icmp_seq=5 ttl=127 time=3.15 ms

--- 192.168.25.115 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4010ms
rtt min/avg/max/mdev = 1.893/2.870/3.270/0.497 ms
mininet@mininet-vm:~$
```

حال سراغ wireshark میرویم:

icmp.type == 0    icmp.type == 8						
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.25.115	192.168.25.115	ICMP	88	Echo (ping) request id=0x0001, seq=25/6400, ttl=63 (reply in 2)
2	0.000234	192.168.25.115	192.168.25.115	ICMP	88	Echo (ping) reply id=0x0001, seq=25/6400, ttl=128 (request in 1)
3	1.002227	192.168.25.115	192.168.25.115	ICMP	88	Echo (ping) request id=0x0001, seq=26/6656, ttl=63 (reply in 4)
4	1.002644	192.168.25.115	192.168.25.115	ICMP	88	Echo (ping) reply id=0x0001, seq=26/6656, ttl=128 (request in 3)
5	2.004668	192.168.25.115	192.168.25.115	ICMP	88	Echo (ping) request id=0x0001, seq=27/6912, ttl=63 (reply in 6)
6	2.005021	192.168.25.115	192.168.25.115	ICMP	88	Echo (ping) reply id=0x0001, seq=27/6912, ttl=128 (request in 5)
7	3.006768	192.168.25.115	192.168.25.115	ICMP	88	Echo (ping) request id=0x0001, seq=28/7168, ttl=63 (reply in 8)
8	3.007148	192.168.25.115	192.168.25.115	ICMP	88	Echo (ping) reply id=0x0001, seq=28/7168, ttl=128 (request in 7)
9	4.009246	192.168.25.115	192.168.25.115	ICMP	88	Echo (ping) request id=0x0001, seq=29/7424, ttl=63 (reply in 10)
10	4.009459	192.168.25.115	192.168.25.115	ICMP	88	Echo (ping) reply id=0x0001, seq=29/7424, ttl=128 (request in 9)

فیلترهایی که زدیم به شرح زیر است:

- icmp.type == 0: با بسته های پاسخ پینگ همخوانی دارد.
- icmp.type == 8: با بسته های درخواست پینگ همخوانی دارد.

پس بطور کلی بسته های درخواست و پاسخ icmp را ضبط کردیم و ترافیک های دیگر را حذف کردیم.

سوال سوم:

الف) همان توپولوژی minimal میباشد.

```
mininet@mininet-vm:~$ sudo mn --topo minimal
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1)
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Starting CLI:
mininet>
```

```
mininet> net
h1 h1-eth0:s1-eth1
h2 h2-eth0:s1-eth2
s1 lo: s1-eth1:h1-eth0 s1-eth2:h2-eth0
c0
mininet> _
```

تمام موارد زیر دو طرفه میباشد.

مطابق تصویر بالا میدانیم هاست h1 از طریق پورت eth0 به سویچ s1 از طریق پورت eth1 و هاست h2 از طریق پورت eth0 به سویچ s1 از طریق پورت eth2 متصل میباشد.

ب) توپولوژی خطی linear با دو سوئیچ که به هر کدام دو هاست متصل است، می باشد.

```
mininet@mininet-vm:~$ sudo mn --topo linear,2,2
*** Creating network
*** Adding controller
*** Adding hosts:
h1s1 h1s2 h2s1 h2s2
*** Adding switches:
s1 s2
*** Adding links:
(h1s1, s1) (h1s2, s2) (h2s1, s1) (h2s2, s2) (s2, s1)
*** Configuring hosts
h1s1 h1s2 h2s1 h2s2
*** Starting controller
c0
*** Starting 2 switches
s1 s2 ...
*** Starting CLI:
mininet> net
h1s1 h1s1-eth0:s1-eth1
h1s2 h1s2-eth0:s2-eth1
h2s1 h2s1-eth0:s1-eth2
h2s2 h2s2-eth0:s2-eth2
s1 lo: s1-eth1:h1s1-eth0 s1-eth2:h2s1-eth0 s1-eth3:s2-eth3
s2 lo: s2-eth1:h1s2-eth0 s2-eth2:h2s2-eth0 s2-eth3:s1-eth3
c0
mininet>
```

مطابق دستور net اتصالات به صورت فوق میباشد. که میدانیم همگی دوطرفه هستند.

هاست h1s1 از طریق پورت eth0 به سویچ s1 از طریق پورت eth1 و هاست h1s2 از طریق پورت eth0 به سویچ s2 از طریق پورت eth1 و هاست h2s1 از طریق پورت eth0 به سویچ s1 از طریق پورت eth2 و هاست h2s2 از طریق پورت eth0 به سویچ s2 از طریق پورت eth2 و سویچ s1 از طریق پورت eth3 به سویچ s2 از طریق پورت eth3 متصل اند.

ج) توپولوژی درختی با عمق 2 است و هر پدر 3 نود فرزند دارد فلذا دستور ساخت آن به صورت زیر است.

```
mininet@mininet-vm:~$ sudo mn --topo tree,depth=2,fanout=3
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3 h4 h5 h6 h7 h8 h9
*** Adding switches:
s1 s2 s3 s4
*** Adding links:
(s1, s2) (s1, s3) (s1, s4) (s2, h1) (s2, h2) (s2, h3) (s3, h4) (s3, h5) (s3, h6) (s4, h7) (s4, h8) (s4, h9)
*** Configuring hosts
h1 h2 h3 h4 h5 h6 h7 h8 h9
*** Starting controller
c0
*** Starting 4 switches
s1 s2 s3 s4 ...
*** Starting CLI:
mininet>
```

```
mininet> net
h1 h1-eth0:s2-eth1
h2 h2-eth0:s2-eth2
h3 h3-eth0:s2-eth3
h4 h4-eth0:s3-eth1
h5 h5-eth0:s3-eth2
h6 h6-eth0:s3-eth3
h7 h7-eth0:s4-eth1
h8 h8-eth0:s4-eth2
h9 h9-eth0:s4-eth3
s1 lo: s1-eth1:s2-eth4 s1-eth2:s3-eth4 s1-eth3:s4-eth4
s2 lo: s2-eth1:h1-eth0 s2-eth2:h2-eth0 s2-eth3:h3-eth0 s2-eth4:s1-eth1
s3 lo: s3-eth1:h4-eth0 s3-eth2:h5-eth0 s3-eth3:h6-eth0 s3-eth4:s1-eth2
s4 lo: s4-eth1:h7-eth0 s4-eth2:h8-eth0 s4-eth3:h9-eth0 s4-eth4:s1-eth3
c0
mininet>
```

دستور بالا را میتوان به صورت زیر نیز اجرا نمود:

Sudo mn -topo tree,2,3

و میدانیم تمام آنها دوطرفه اند.

- هاست‌های h1، h2 و h3 از طریق پورت eth0 خود به سویچ s2 از طریق پورت‌های eth1، eth2 و eth3
- هاست‌های h4، h5 و h6 از طریق پورت eth0 خود به سویچ s3 از طریق پورت‌های eth1، eth2 و eth3
- هاست‌های h7، h8 و h9 از طریق پورت eth0 خود به سویچ s4 از طریق پورت‌های eth1، eth2 و eth3
- سویچ‌های s2، s3 و s4 از طریق پورت eth4 خود به سویچ s1 از طریق پورت‌های eth1، eth2 و eth3

### سوال چهارم:

ابتدا پهنای باند را ثابت گرفته و تاخیر را برای 10 مقدار داده شده تنظیم مینماییم.

❖ پهنای باند ثابت BW=100

• Delay=0.01ms

دستور زیر را اجرا میکنیم:

```
sudo mn --topo minimal --link tc,bw=100,delay=0.01ms
```

```
mininet@mininet-vm:~$ sudo mn --topo minimal --link tc,bw=100,delay=0.01ms
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(100.00Mbit 0.01ms delay) (100.00Mbit 0.01ms delay) (h1, s1) (100.00Mbit 0.01ms delay) (100.00Mbit 0.01ms delay) (h2, s1)
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s1 ... (100.00Mbit 0.01ms delay) (100.00Mbit 0.01ms delay)
*** Starting CLI:
```



```

mininet> iperf h1 h2
*** Iperf: testing TCP bandwidth between h1 and h2
*** Results: ['94.4 Mbits/sec', '96.6 Mbits/sec']
mininet> pingallfull
*** Ping: testing ping reachability
h1 -> h2
h2 -> h1
*** Results:
h1->h2: 1/1, rtt min/avg/max/mdev 1.487/1.487/1.487/0.000 ms
h2->h1: 1/1, rtt min/avg/max/mdev 2.554/2.554/2.554/0.000 ms
mininet>

```

Delay=0.05ms •

```

mininet@mininet-vm:~$ sudo mn --topo minimal --link tc,bw=100,delay=0.05ms
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(100.00Mbit 0.05ms delay) (100.00Mbit 0.05ms delay) (h1, s1) (100.00Mbit 0.05ms delay) (100.00Mbit 0.05ms delay) (h2, s1)
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s1 ... (100.00Mbit 0.05ms delay) (100.00Mbit 0.05ms delay)
*** Starting CLI:
mininet> pingallfull
*** Ping: testing ping reachability
h1 -> h2
h2 -> h1
*** Results:
h1->h2: 1/1, rtt min/avg/max/mdev 6.558/6.558/6.558/0.000 ms
h2->h1: 1/1, rtt min/avg/max/mdev 1.392/1.392/1.392/0.000 ms
mininet> iperf
*** Iperf: testing TCP bandwidth between h1 and h2
*** Results: ['94.4 Mbits/sec', '112 Mbits/sec']
mininet>

```

Delay=0.1ms •

```

mininet@mininet-vm:~$ sudo mn --topo minimal --link tc,bw=100,delay=0.1ms
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(100.00Mbit 0.1ms delay) (100.00Mbit 0.1ms delay) (h1, s1) (100.00Mbit 0.1ms delay) (100.00Mbit 0.1ms delay) (h2, s1)
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s1 ... (100.00Mbit 0.1ms delay) (100.00Mbit 0.1ms delay)
*** Starting CLI:
mininet> pingallfull
*** Ping: testing ping reachability
h1 -> h2
h2 -> h1
*** Results:
h1->h2: 1/1, rtt min/avg/max/mdev 6.043/6.043/6.043/0.000 ms
h2->h1: 1/1, rtt min/avg/max/mdev 1.558/1.558/1.558/0.000 ms
mininet> iperf
*** Iperf: testing TCP bandwidth between h1 and h2
*** Results: ['93.9 Mbits/sec', '111 Mbits/sec']
mininet>

```

Delay=0.5ms •

```
mininet@mininet-vm:~$ sudo mn --topo minimal --link tc,bw=100,delay=0.5ms
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(100.00Mbit 0.5ms delay) (100.00Mbit 0.5ms delay) (h1, s1) (100.00Mbit 0.5ms delay) (100.00Mbit 0.5ms delay) (h2, s1)
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s1 ... (100.00Mbit 0.5ms delay) (100.00Mbit 0.5ms delay)
*** Starting CLI:
mininet> pingallfull
*** Ping: testing ping reachability
h1 -> h2
h2 -> h1
*** Results:
  h1->h2: 1/1, rtt min/avg/max/mdev 7.973/7.973/7.973/0.000 ms
  h2->h1: 1/1, rtt min/avg/max/mdev 3.177/3.177/3.177/0.000 ms
mininet> iperf
*** Iperf: testing TCP bandwidth between h1 and h2
*** Results: ['94.2 Mbits/sec', '111 Mbits/sec']
mininet>
```

Delay=1ms •

```
mininet@mininet-vm:~$ sudo mn --topo minimal --link tc,bw=100,delay=1ms
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(100.00Mbit 1ms delay) (100.00Mbit 1ms delay) (h1, s1) (100.00Mbit 1ms delay) (100.00Mbit 1ms delay) (h2, s1)
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s1 ... (100.00Mbit 1ms delay) (100.00Mbit 1ms delay)
*** Starting CLI:
mininet> pingallfull
*** Ping: testing ping reachability
h1 -> h2
h2 -> h1
*** Results:
  h1->h2: 1/1, rtt min/avg/max/mdev 12.243/12.243/12.243/0.000 ms
  h2->h1: 1/1, rtt min/avg/max/mdev 6.057/6.057/6.057/0.000 ms
mininet> iperf
*** Iperf: testing TCP bandwidth between h1 and h2
*** Results: ['91.0 Mbits/sec', '108 Mbits/sec']
mininet>
```

Delay=5ms •

```

mininet@mininet-vm:~$ sudo mn --topo minimal --link tc,bw=100,delay=5ms
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(100.00Mbit 5ms delay) (100.00Mbit 5ms delay) (h1, s1) (100.00Mbit 5ms delay) (100.00Mbit 5ms delay)
(h2, s1)
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s1 ... (100.00Mbit 5ms delay) (100.00Mbit 5ms delay)
*** Starting CLI:
mininet> pingallfull
*** Ping: testing ping reachability
h1 -> h2
h2 -> h1
*** Results:
h1->h2: 1/1, rtt min/avg/max/mdev 46.403/46.403/46.403/0.000 ms
h2->h1: 1/1, rtt min/avg/max/mdev 23.877/23.877/23.877/0.000 ms
mininet> iperf
*** Iperf: testing TCP bandwidth between h1 and h2
*** Results: ['93.5 Mbits/sec', '109 Mbits/sec']
mininet> _

```

Delay=10ms •

```

mininet@mininet-vm:~$ sudo mn --topo minimal --link tc,bw=100,delay=10ms
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(100.00Mbit 10ms delay) (100.00Mbit 10ms delay) (h1, s1) (100.00Mbit 10ms delay) (100.00Mbit 10ms de
lay) (h2, s1)
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s1 ... (100.00Mbit 10ms delay) (100.00Mbit 10ms delay)
*** Starting CLI:
mininet> pingallfull
*** Ping: testing ping reachability
h1 -> h2
h2 -> h1
*** Results:
h1->h2: 1/1, rtt min/avg/max/mdev 87.245/87.245/87.245/0.000 ms
h2->h1: 1/1, rtt min/avg/max/mdev 43.674/43.674/43.674/0.000 ms
mininet> iperf
*** Iperf: testing TCP bandwidth between h1 and h2
*** Results: ['90.7 Mbits/sec', '106 Mbits/sec']
mininet>

```

Delay=50ms •

```

mininet@mininet-vm:~$ sudo mn --topo minimal --link tc,bw=100,delay=50ms
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(100.00Mbit 50ms delay) (100.00Mbit 50ms delay) (h1, s1) (100.00Mbit 50ms delay) (100.00Mbit 50ms de
lay) (h2, s1)
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s1 ... (100.00Mbit 50ms delay) (100.00Mbit 50ms delay)
*** Starting CLI:
mininet> pingallfull
*** Ping: testing ping reachability
h1 -> h2
h2 -> h1
*** Results:
  h1->h2: 1/1, rtt min/avg/max/mdev 407.193/407.193/407.193/0.000 ms
  h2->h1: 1/1, rtt min/avg/max/mdev 203.773/203.773/203.773/0.000 ms
mininet> iperf
*** Iperf: testing TCP bandwidth between h1 and h2
*** Results: ['73.7 Mbits/sec', '83.5 Mbits/sec']
mininet>

```

Delay=100ms •

```

mininet@mininet-vm:~$ sudo mn --topo minimal --link tc,bw=100,delay=100ms
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(100.00Mbit 100ms delay) (100.00Mbit 100ms delay) (h1, s1) (100.00Mbit 100ms delay) (100.00Mbit 100m
s delay) (h2, s1)
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s1 ... (100.00Mbit 100ms delay) (100.00Mbit 100ms delay)
*** Starting CLI:
mininet> pingallfull
*** Ping: testing ping reachability
h1 -> h2
h2 -> h1
*** Results:
  h1->h2: 1/1, rtt min/avg/max/mdev 807.394/807.394/807.394/0.000 ms
  h2->h1: 1/1, rtt min/avg/max/mdev 402.926/402.926/402.926/0.000 ms
mininet> iperf
*** Iperf: testing TCP bandwidth between h1 and h2
*** Results: ['46.2 Mbits/sec', '50.5 Mbits/sec']
mininet>

```

Delay=500ms •

```

mininet@mininet-vm:~$ sudo mn --topo minimal --link tc,bw=100,delay=500ms
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(100.00Mbit 500ms delay) (100.00Mbit 500ms delay) (h1, s1) (100.00Mbit 500ms delay) (100.00Mbit 500ms delay) (h2, s1)
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s1 ... (100.00Mbit 500ms delay) (100.00Mbit 500ms delay)
*** Starting CLI:
mininet> pingallfull
*** Ping: testing ping reachability
h1 -> h2
h2 -> h1
*** Results:
h1->h2: 1/1, rtt min/avg/max/mdev 4006.387/4006.387/4006.387/0.000 ms
h2->h1: 1/1, rtt min/avg/max/mdev 2003.702/2003.702/2003.702/0.000 ms
mininet> iperf
*** Iperf: testing TCP bandwidth between h1 and h2
*** Results: ['392 Kbits/sec', '524 Kbits/sec']
mininet> _

```

Delay (ms)	RTT (ms)	Measured BW (Mb/s)
0.01	1.49 - 2.55	94.4 - 96.6
0.05	6.56 - 1.39	94.4 - 112
0.1	6.04 - 1.56	93.9 - 111
0.5	7.97 - 3.18	94.2 - 111
1.0	12.24 - 6.06	91.0 - 108
5.0	46.40 - 23.88	93.5 - 109
10.0	87.25 - 43.68	90.7 - 106
50.0	407.12 - 203.77	73.7 - 83.5
100.0	807.34 - 402.93	46.2 - 50.5
500.0	4006.39 - 2003.70	0.392 - 0.524

❖ تاخير ثابت Delay=1ms

• BW=0.01

دستور زیر را اجرا میکنیم:

```
sudo mn --topo minimal --link tc,bw=0.01,delay=1ms
```

```

mininet@mininet-vm:~$ sudo mn --topo minimal --link tc,bw=0.01,delay=1ms
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(0.01Mbit 1ms delay) (0.01Mbit 1ms delay) (h1, s1) (0.01Mbit 1ms delay) (0.01Mbit 1ms delay) (h2, s1)
)
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s1 ... (0.01Mbit 1ms delay) (0.01Mbit 1ms delay)
*** Starting CLI:
mininet> pingallfull
*** Ping: testing ping reachability
h1 -> h2
h2 -> h1
*** Results:
h1->h2: 1/1, rtt min/avg/max/mdev 12.419/12.419/12.419/0.000 ms
h2->h1: 1/1, rtt min/avg/max/mdev 5.530/5.530/5.530/0.000 ms
mininet> iperf
*** Iperf: testing TCP bandwidth between h1 and h2
*** Results: ['9.55 Kbits/sec', '191 Kbits/sec']
mininet> _

```

BW=0.05 •

```

mininet@mininet-vm:~$ sudo mn --topo minimal --link tc,bw=0.05,delay=1ms
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(0.05Mbit 1ms delay) (0.05Mbit 1ms delay) (h1, s1) (0.05Mbit 1ms delay) (0.05Mbit 1ms delay) (h2, s1)
)
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s1 ... (0.05Mbit 1ms delay) (0.05Mbit 1ms delay)
*** Starting CLI:
mininet> pingallfull
*** Ping: testing ping reachability
h1 -> h2
h2 -> h1
*** Results:
h1->h2: 1/1, rtt min/avg/max/mdev 12.673/12.673/12.673/0.000 ms
h2->h1: 1/1, rtt min/avg/max/mdev 4.941/4.941/4.941/0.000 ms
mininet> iperf
*** Iperf: testing TCP bandwidth between h1 and h2
*** Results: ['48.1 Kbits/sec', '442 Kbits/sec']
mininet>

```

BW=0.1 •

```

mininet@mininet-vm:~$ sudo mn --topo minimal --link tc,bw=0.1,delay=1ms
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(0.10Mbit 1ms delay) (0.10Mbit 1ms delay) (h1, s1) (0.10Mbit 1ms delay) (0.10Mbit 1ms delay) (h2, s1)
)
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s1 ...(0.10Mbit 1ms delay) (0.10Mbit 1ms delay)
*** Starting CLI:
mininet> pingallfull
*** Ping: testing ping reachability
h1 -> h2
h2 -> h1
*** Results:
h1->h2: 1/1, rtt min/avg/max/mdev 13.429/13.429/13.429/0.000 ms
h2->h1: 1/1, rtt min/avg/max/mdev 5.430/5.430/5.430/0.000 ms
mininet> iperf
*** Iperf: testing TCP bandwidth between h1 and h2
*** Results: ['96.4 Kbits/sec', '376 Kbits/sec']
mininet> _

```

BW=0.5 •

```

mininet@mininet-vm:~$ sudo mn --topo minimal --link tc,bw=0.5,delay=1ms
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(0.50Mbit 1ms delay) (0.50Mbit 1ms delay) (h1, s1) (0.50Mbit 1ms delay) (0.50Mbit 1ms delay) (h2, s1)
)
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s1 ...(0.50Mbit 1ms delay) (0.50Mbit 1ms delay)
*** Starting CLI:
mininet> pingallfull
*** Ping: testing ping reachability
h1 -> h2
h2 -> h1
*** Results:
h1->h2: 1/1, rtt min/avg/max/mdev 11.437/11.437/11.437/0.000 ms
h2->h1: 1/1, rtt min/avg/max/mdev 6.826/6.826/6.826/0.000 ms
mininet> iperf
*** Iperf: testing TCP bandwidth between h1 and h2
*** Results: ['480 Kbits/sec', '996 Kbits/sec']
mininet>

```

BW=1 •

```

mininet@mininet-vm:~$ sudo mn --topo minimal --link tc,bw=1,delay=1ms
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(1.00Mbit 1ms delay) (1.00Mbit 1ms delay) (h1, s1) (1.00Mbit 1ms delay) (1.00Mbit 1ms delay) (h2, s1)
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s1 ... (1.00Mbit 1ms delay) (1.00Mbit 1ms delay)
*** Starting CLI:
mininet> pingallfull
*** Ping: testing ping reachability
h1 -> h2
h2 -> h1
*** Results:
  h1->h2: 1/1, rtt min/avg/max/mdev 13.586/13.586/13.586/0.000 ms
  h2->h1: 1/1, rtt min/avg/max/mdev 5.771/5.771/5.771/0.000 ms
mininet> iperf
*** Iperf: testing TCP bandwidth between h1 and h2
*** Results: ['958 Kbits/sec', '1.58 Mbits/sec']
mininet> _

```

BW=5 •

```

mininet@mininet-vm:~$ sudo mn --topo minimal --link tc,bw=5,delay=1ms
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(5.00Mbit 1ms delay) (5.00Mbit 1ms delay) (h1, s1) (5.00Mbit 1ms delay) (5.00Mbit 1ms delay) (h2, s1)
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s1 ... (5.00Mbit 1ms delay) (5.00Mbit 1ms delay)
*** Starting CLI:
mininet> pingallfull
*** Ping: testing ping reachability
h1 -> h2
h2 -> h1
*** Results:
  h1->h2: 1/1, rtt min/avg/max/mdev 12.951/12.951/12.951/0.000 ms
  h2->h1: 1/1, rtt min/avg/max/mdev 6.317/6.317/6.317/0.000 ms
mininet> iperf
*** Iperf: testing TCP bandwidth between h1 and h2
*** Results: ['4.78 Mbits/sec', '6.19 Mbits/sec']
mininet> _

```

BW=10 •



```

mininet@mininet-vm:~$ sudo mn --topo minimal --link tc,bw=10,delay=1ms
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(10.00Mbit 1ms delay) (10.00Mbit 1ms delay) (h1, s1) (10.00Mbit 1ms delay) (10.00Mbit 1ms delay) (h2
, s1)
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s1 ... (10.00Mbit 1ms delay) (10.00Mbit 1ms delay)
*** Starting CLI:
mininet> pingallfull
*** Ping: testing ping reachability
h1 -> h2
h2 -> h1
*** Results:
h1->h2: 1/1, rtt min/avg/max/mdev 12.900/12.900/12.900/0.000 ms
h2->h1: 1/1, rtt min/avg/max/mdev 5.292/5.292/5.292/0.000 ms

```

```

mininet> iperf
*** Iperf: testing TCP bandwidth between h1 and h2
*** Results: ['9.55 Mbits/sec', '12.0 Mbits/sec']
mininet> _

```

BW=50 •

```

mininet@mininet-vm:~$ sudo mn --topo minimal --link tc,bw=50,delay=1ms
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(50.00Mbit 1ms delay) (50.00Mbit 1ms delay) (h1, s1) (50.00Mbit 1ms delay) (50.00Mbit 1ms delay) (h2
, s1)
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s1 ... (50.00Mbit 1ms delay) (50.00Mbit 1ms delay)
*** Starting CLI:
mininet> pingallfull
*** Ping: testing ping reachability
h1 -> h2
h2 -> h1
*** Results:
h1->h2: 1/1, rtt min/avg/max/mdev 12.189/12.189/12.189/0.000 ms
h2->h1: 1/1, rtt min/avg/max/mdev 5.187/5.187/5.187/0.000 ms
mininet> iperf
*** Iperf: testing TCP bandwidth between h1 and h2
*** Results: ['47.0 Mbits/sec', '56.1 Mbits/sec']
mininet>

```

BW=100 •

```

mininet@mininet-vm:~$ sudo mn --topo minimal --link tc,bw=100,delay=1ms
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(100.00Mbit 1ms delay) (100.00Mbit 1ms delay) (h1, s1) (100.00Mbit 1ms delay) (100.00Mbit 1ms delay)
(h2, s1)
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s1 ... (100.00Mbit 1ms delay) (100.00Mbit 1ms delay)
*** Starting CLI:
mininet> pingallfull
*** Ping: testing ping reachability
h1 -> h2
h2 -> h1
*** Results:
h1->h2: 1/1, rtt min/avg/max/mdev 12.515/12.515/12.515/0.000 ms
h2->h1: 1/1, rtt min/avg/max/mdev 5.102/5.102/5.102/0.000 ms
mininet> iperf
*** Iperf: testing TCP bandwidth between h1 and h2
*** Results: ['90.2 Mbits/sec', '107 Mbits/sec']
mininet> _

```

BW=500 •

```

mininet@mininet-vm:~$ sudo mn --topo minimal --link tc,bw=500,delay=1ms
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(500.00Mbit 1ms delay) (500.00Mbit 1ms delay) (h1, s1) (500.00Mbit 1ms delay) (500.00Mbit 1ms delay)
(h2, s1)
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s1 ... (500.00Mbit 1ms delay) (500.00Mbit 1ms delay)
*** Starting CLI:
mininet> pingallfull
*** Ping: testing ping reachability
h1 -> h2
h2 -> h1
*** Results:
h1->h2: 1/1, rtt min/avg/max/mdev 11.813/11.813/11.813/0.000 ms
h2->h1: 1/1, rtt min/avg/max/mdev 5.599/5.599/5.599/0.000 ms
mininet> iperf
*** Iperf: testing TCP bandwidth between h1 and h2
*** Results: ['395 Mbits/sec', '415 Mbits/sec']
mininet>

```

Bandwidth (Mb/s)	RTT (ms)	Measured BW (Mb/s)
0.01	12.42 - 5.53	0.009 - 0.191
0.05	12.67 - 4.94	0.048 - 0.442
0.1	13.43 - 5.43	0.096 - 0.376
0.5	11.44 - 6.83	0.480 - 0.996
1.0	13.59 - 5.77	0.958 - 1.58
5.0	12.95 - 6.32	4.78 - 6.19
10.0	12.90 - 5.29	9.55 - 12.0
50.0	12.19 - 5.19	47.0 - 56.1

100.0	12.52 - 5.10	90.2 - 107
500.0	11.82 - 5.599	395 - 415

نتیجه گیری جدول اول:

✓ طبق جدول اول و با ثابت بودن پهنای باند و تغییر Delay، میبینیم با افزایش delay لینک ها، RTT افزایش میابد. و دلیل آن این است که هر بسته در رفت و برگشت، 4 مرتبه روی لینکها قرار میگیرد که شامل 2 لینک ارسال و پاسخ است. اگر تاخیر لینکها افزایش یابد، پس RTT نیز حداقل به اندازه 4 برابر تاخیر افزایش می یابد.

✓ با افزایش Delay لینک ها، پهنای باند کاهش میابد. به این دلیل که با افزایش تاخیر لینک، داده در زمان بیشتری روی لینک منتقل میشود. و مخرج کسر پهنای باند زیاد و در نتیجه پهنای باند کاهش می یابد. در تاخیر حدود 10ms میزان کاهش پهنای باند ناچیز است اما با افزایش ناگهانی تاخیر، پهنای باند به طور نمایی و شدید کاهش می یابد.

نتیجه گیری جدول دوم:

✓ طبق جدول دوم، وقتی Delay ثابت است، تغییر پهنای باند تاثیری روی RTT ندارد. طول بسته 64 بیت پینگ بوده و در برابر پهنای باند بسیار ناچیز است. و این یعنی لینک توسط یک بسته اشغال شده و همانند جاده ای که تنها یک ماشین در آن عبور میکند، ماشین میتواند با هر سرعتی به حرکتش ادامه دهد. فلذا پهنای باند تاثیری در RTT یا زمان رفت و برگشت ندارد.

✓ با افزایش پهنای باند، پهنای باند اندازه گیری شده افزایش می یابد. زیرا همانطور که در جدول میبینیم پهنای باند لینک تقریباً با پهنای باند اندازه گیری شده برابر است و هر دوی آنها به یک چیز اشاره دارند و رابطه آنها مستقیم است. هرچند پهنای باند اندازه گیری شده کمی کمتر از واقعی است.