


بخش اول

```
PS C:\Windows\system32> ping 1.1.1.1

Pinging 1.1.1.1 with 32 bytes of data:
Reply from 1.1.1.1: bytes=32 time=111ms TTL=48
Reply from 1.1.1.1: bytes=32 time=112ms TTL=48
Reply from 1.1.1.1: bytes=32 time=109ms TTL=48
Reply from 1.1.1.1: bytes=32 time=108ms TTL=48


Ping statistics for 1.1.1.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 108ms, Maximum = 112ms, Average = 110ms
PS C:\Windows\system32>
```



```
PS C:\Windows\system32> ping 8.8.8.8

Pinging 8.8.8.8 with 32 bytes of data:
Reply from 8.8.8.8: bytes=32 time=54ms TTL=108
Reply from 8.8.8.8: bytes=32 time=53ms TTL=108
Reply from 8.8.8.8: bytes=32 time=51ms TTL=108
Reply from 8.8.8.8: bytes=32 time=53ms TTL=108

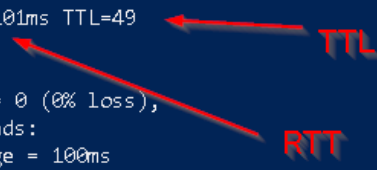
Ping statistics for 8.8.8.8:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 51ms, Maximum = 54ms, Average = 52ms
PS C:\Windows\system32>
```



```
PS C:\Windows\system32> ping www.twitch.tv

Pinging twitch.map.fastly.net [199.232.138.167] with 32 bytes of data:
Reply from 199.232.138.167: bytes=32 time=100ms TTL=49
Reply from 199.232.138.167: bytes=32 time=101ms TTL=49
Reply from 199.232.138.167: bytes=32 time=99ms TTL=49
Reply from 199.232.138.167: bytes=32 time=101ms TTL=49

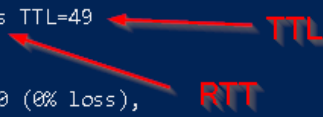
Ping statistics for 199.232.138.167:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 99ms, Maximum = 101ms, Average = 100ms
PS C:\Windows\system32>
```



```
PS C:\Windows\system32> ping www.google.com

Pinging www.google.com [142.250.185.36] with 32 bytes of data:
Reply from 142.250.185.36: bytes=32 time=51ms TTL=49
Reply from 142.250.185.36: bytes=32 time=50ms TTL=49
Reply from 142.250.185.36: bytes=32 time=53ms TTL=49
Reply from 142.250.185.36: bytes=32 time=50ms TTL=49


Ping statistics for 142.250.185.36:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 50ms, Maximum = 53ms, Average = 51ms
PS C:\Windows\system32>
```



```
PS C:\Windows\system32> ping www.crackwatch.com

Pinging www.crackwatch.com [104.21.94.211] with 32 bytes of data:
Reply from 104.21.94.211: bytes=32 time=113ms TTL=48
Reply from 104.21.94.211: bytes=32 time=109ms TTL=48
Reply from 104.21.94.211: bytes=32 time=110ms TTL=48
Reply from 104.21.94.211: bytes=32 time=187ms TTL=48

Ping statistics for 104.21.94.211:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 109ms, Maximum = 187ms, Average = 129ms
PS C:\Windows\system32>
```



RTT به معنای مدت زمانی است که یک packet طول میکشد که مسیر رفت و برگشت به آدرس ip هدف را طی کند.  
TTL به معنای میزان باقی مانده از عمر یک packet است که به ازای هر عبور از یک router در شبکه میزان آن یک واحد کاهش میابد.

با این تعاریف میتوان گفت که با افزایش RTT عموماً مقدار TTL کاهش میابد

بخش دوم)

ابتدا با استفاده از دستور زیر IP ماشین مجازی mininet را به دست می آوریم

```
mininet@mininet-vm:~$ ifconfig eth0
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.125.128 netmask 255.255.255.0 broadcast 192.168.125.255
    ether 00:0c:29:79:fb:3e txqueuelen 1000 (Ethernet)
    RX packets 97 bytes 19766 (19.7 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 93 bytes 9221 (9.2 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

mininet@mininet-vm:~$
```

حال به آدرس گفته شده ping میزنیم

```
PS C:\Windows\system32> ping 192.168.125.128 -n 5

Pinging 192.168.125.128 with 32 bytes of data:
Reply from 192.168.125.128: bytes=32 time<1ms TTL=64
Reply from 192.168.125.128: bytes=32 time<1ms TTL=64
Reply from 192.168.125.128: bytes=32 time<1ms TTL=64
Reply from 192.168.125.128: bytes=32 time<1ms TTL=64
Reply from 192.168.125.128: bytes=32 time<1ms TTL=64

Ping statistics for 192.168.125.128:
    Packets: Sent = 5, Received = 5, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
PS C:\Windows\system32>
```

با استفاده از flag اشاره شده در تصویر بالا ( -n ) میتوانیم تعداد packet ها را تعیین کنیم

در wireshark نوع درخواست را بر روی icmp قرار داده که در این حالت 10 عدد request خواهیم داشت

5 عدد برای درخواست پینگ و 5 عدد برای جواب پینگ

icmp						
No.	Time	Source	Destination	Protocol	Length	Info
5	3.972295	192.168.125.1	192.168.125.128	ICMP	74	Echo (ping) request id=0x0001, seq=68/17408, ttl=128 (reply in 6)
6	3.972526	192.168.125.128	192.168.125.1	ICMP	74	Echo (ping) reply id=0x0001, seq=68/17408, ttl=64 (request in 5)
7	4.982629	192.168.125.1	192.168.125.128	ICMP	74	Echo (ping) request id=0x0001, seq=69/17664, ttl=128 (reply in 8)
8	4.982929	192.168.125.128	192.168.125.1	ICMP	74	Echo (ping) reply id=0x0001, seq=69/17664, ttl=64 (request in 7)
9	5.994059	192.168.125.1	192.168.125.128	ICMP	74	Echo (ping) request id=0x0001, seq=70/17920, ttl=128 (reply in 10)
10	5.994356	192.168.125.128	192.168.125.1	ICMP	74	Echo (ping) reply id=0x0001, seq=70/17920, ttl=64 (request in 9)
11	7.002546	192.168.125.1	192.168.125.128	ICMP	74	Echo (ping) request id=0x0001, seq=71/18176, ttl=128 (reply in 12)
12	7.002810	192.168.125.128	192.168.125.1	ICMP	74	Echo (ping) reply id=0x0001, seq=71/18176, ttl=64 (request in 11)
13	8.014810	192.168.125.1	192.168.125.128	ICMP	74	Echo (ping) request id=0x0001, seq=72/18432, ttl=128 (reply in 14)
14	8.015114	192.168.125.128	192.168.125.1	ICMP	74	Echo (ping) reply id=0x0001, seq=72/18432, ttl=64 (request in 13)

بخش سوم)

الف:

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

ب)

```
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> _
```

```

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> 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> nodes
available nodes are:
c0 h1 h2 h3 h4 h5 h6 h7 h8 h9 s1 s2 s3 s4
mininet>

```