

Part1: Web Browsing (DNS, TCP)

1. Find the first DNS request packet sent by the client.(Request for **cse.nsysu.edu.tw**)

You can find a record like below on Wireshark. And you can answer the question.

```
5729 16.137999 140.117.171.179 140.117.11.1 DNS 76 Standard query 0x8b98 A lis.nsysu.edu.tw
```

(1) Examine the Ethernet

```
Ethernet II, Src: HewlettP_4f:6b:41 (40:a8:f0:4f:6b:41), Dst: JuniperN_73:14:01 (88:e0:f3:73:14:01)
> Destination: JuniperN_73:14:01 (88:e0:f3:73:14:01)
> Source: HewlettP_4f:6b:41 (40:a8:f0:4f:6b:41)
Type: IPv4 (0x0800)
```

a. What is the Ethernet address of the source and destination?

Source: HewlettP_4f:6b:41 (40:a8:f0:4f:6b:41)

Destination: JuniperN_73:14:01 (88:e0:f3:73:14:01)

b. What is the content of the type field in the Ethernet frame?

Type: IPv4 (0x0800)

(2) Examine the Internet Protocol

```
Internet Protocol Version 4, Src: 140.117.171.179, Dst: 140.117.11.1
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
> Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
  Total Length: 62
  Identification: 0x24c5 (9413)
> Flags: 0x00
  Fragment offset: 0
  Time to live: 128
  Protocol: UDP (17)
  Header checksum: 0x0000 [validation disabled]
  [Header checksum status: Unverified]
  Source: 140.117.171.179
  Destination: 140.117.11.1
```

a. What is the IP address of the source and destination?

source: 140.117.171.179

destination: 140.117.11.1

b. What is the header length? What is the total packet length?

Header Length: 20 bytes

Total Length: 62 bytes

c. Identify the protocol type field. What is the number and type of the protocol in the payload?

Protocol: UDP (17)

(3) Examine the **User Datagram Protocol**

```
User Datagram Protocol, Src Port: 55438, Dst Port: 53
```

```
Source Port: 55438
```

```
Destination Port: 53
```

```
Length: 42
```

```
Checksum: 0xcfd8 [unverified]
```

```
[Checksum Status: Unverified]
```

```
[Stream index: 194]
```

- a. Identify the client ephemeral port number and the server well-known port number .

client:55438

Server:53

- b. What type of application layer protocol is in the payload?

UDP

(4) Examine the **Domain Name System (query)**

```
Domain Name System (query)
```

```
\[Response In: 5807\]
```

```
Transaction ID: 0x8b98
```

```
> Flags: 0x0100 Standard query
```

```
Questions: 1
```

```
Answer RRs: 0
```

```
Authority RRs: 0
```

```
Additional RRs: 0
```

```
> Queries
```

- a. What field indicates whether the message is a query or a response?

Domain Name System (query)

- b. What is the query transaction ID?

Transaction ID: 0x8b98

- c. Identify the fields that carry the type and class of the query.

Flags: 0x0100 Standard query

Questions: 1

Answer RRs: 0

Authority RRs: 0

Additional RRs: 0

2. Find the DNS response packet which is response to the DNS request packet from the above question.

You can find a record like below on Wireshark. And you can answer the question.

```
5806 16.143660 140.117.11.1 140.117.171.179 DNS 196 Standard query response 0x61a4 A alumni.sec.nsysu.edu.tw A 140.117.13.244
```

(1) Examine the Ethernet.

```
Ethernet II, Src: JuniperN_73:14:01 (88:e0:f3:73:14:01), Dst: HewlettP_4f:6b:41 (40:a8:f0:4f:6b:41)
> Destination: HewlettP_4f:6b:41 (40:a8:f0:4f:6b:41)
> Source: JuniperN_73:14:01 (88:e0:f3:73:14:01)
Type: IPv4 (0x0800)
```

a. What is the Ethernet address of the source and destination?

Source: JuniperN_73:14:01 (88:e0:f3:73:14:01)

Destination: HewlettP_4f:6b:41 (40:a8:f0:4f:6b:41)

b. What is the content of the type field in the Ethernet frame?

Type: IPv4 (0x0800)

(2) Examine the Internet Protocol & Domain Name System (response)

```
Internet Protocol Version 4, Src: 140.117.11.1, Dst: 140.117.171.179
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
> Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
  Total Length: 182
  Identification: 0xeda4 (60842)
> Flags: 0x02 (Don't Fragment)
  Fragment offset: 0
  Time to live: 251
  Protocol: UDP (17)
  Header checksum: 0xc1ec [validation disabled]
  [Header checksum status: Unverified]
  Source: 140.117.11.1
  Destination: 140.117.171.179
```

a. What is the IP address of the source and destination?

Source: 140.117.11.1

Destination: 140.117.171.179

b. What is the header length? What is the total packet length? Is it longer than the query?

Header Length: 20 bytes (5)

Total Length: 182 bytes

Yes , response 比 query 長

c. How many answers are provided in the response message? Compare the answers and their time-to-live values.

One , time to live:3

3. Find the first TCP packet sent by client. (The destination IP address is response from above question.) You can find three record like below on Wireshark. It's TCP three-way handshake.

(1) Examine the Transmission Control Protocol.

996	8.132505	140.117.171.179	104.115.172.121	TCP	66	50936→80	[SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
999	8.137538	104.115.172.121	140.117.171.179	TCP	66	80→50936	[SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=32
1000	8.137569	140.117.171.179	104.115.172.121	TCP	54	50936→80	[ACK] Seq=1 Ack=1 Win=65536 Len=0

- a. What are the ephemeral port number used by the client and the well-known port number used by the server?

client :50936

server:80

- b. What is the length of the TCP segment?

Len:0

- c. What is the initial sequence number for the segments from the client to the server?

```

Transmission Control Protocol, Src Port: 50936, Dst Port: 80, Seq: 0, Len: 0
  Source Port: 50936
  Destination Port: 80
  [Stream index: 38]
  [TCP Segment Len: 0]
  Sequence number: 0 (relative sequence number)
  Acknowledgment number: 0
  Header Length: 32 bytes
  > Flags: 0x002 (SYN)
    Window size value: 8192
    [Calculated window size: 8192]
    Checksum: 0x4d3c [unverified]
    [Checksum Status: Unverified]
    Urgent pointer: 0
  > Options: (12 bytes), Maximum segment size, No-Operation (NOP), Window scale, No-Operation (NOP), No-Operation (NOP), SACK permitted

```

Sequence number: 0 (relative sequence number)

- d. What is the initial window size?

Window size value: 8192

- e. What is the maximum segment size?

```

  > Options: (12 bytes), Maximum segment size, No-Operation (NOP), Window scale, No-Operation (NOP), No-Operation (NOP), SACK permitted
    > Maximum segment size: 1460 bytes
    > No-Operation (NOP)
    > Window scale: 8 (multiply by 256)
    > No-Operation (NOP)
    > No-Operation (NOP)
    > TCP SACK Permitted Option: True

```

Maximum segment size: 1460 bytes

- f. Find the hex character that contains the SYN flag bit

Flags: 0x002 (SYN)

Part 2 Probing the Internet (ICMP, PING, Traceroute)

```
命令提示字元
Microsoft Windows [版本 10.0.10240]
(c) 2015 Microsoft Corporation. All rights reserved.

C:\Users\chen jia ming>ping 8.8.8.8

Ping 8.8.8.8 (使用 32 位元組的資料):
回覆自 8.8.8.8: 位元組=32 時間=15ms TTL=48
回覆自 8.8.8.8: 位元組=32 時間=15ms TTL=48
回覆自 8.8.8.8: 位元組=32 時間=15ms TTL=48
回覆自 8.8.8.8: 位元組=32 時間=15ms TTL=48

8.8.8.8 的 Ping 統計資料:
    封包: 已傳送 = 4, 已收到 = 4, 已遺失 = 0 (0% 遺失),
    大約的來回時間 (毫秒):
        最小值 = 15ms, 最大值 = 15ms, 平均 = 15ms

C:\Users\chen jia ming>
```

1. Ping Captured.

(1) Find the first **ICMP Echo Request** packet.

454	4.380460	140.117.171.179	8.8.8.8	ICMP	74 Echo (ping) request id=0x0001, seq=1/256, ttl=128 (reply in 455)
-----	----------	-----------------	---------	------	---

a. First, examine the **Internet Protocol**. What is the Time-to-Live?

Time to live: 128

```
Internet Protocol Version 4, Src: 140.117.171.179, Dst: 8.8.8.8
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 60
    Identification: 0x2611 (9745)
  > Flags: 0x00
    Fragment offset: 0
    Time to live: 128
    Protocol: ICMP (1)
    Header checksum: 0x0000 [validation disabled]
    [Header checksum status: Unverified]
    Source: 140.117.171.179
    Destination: 8.8.8.8
```

b. Next examine the **Internet Control Message Protocol**. What is the ICMP message type?

Type: 8 (Echo (ping) request)

```

Internet Control Message Protocol
Type: 8 (Echo (ping) request)
Code: 0
Checksum: 0x4d5a [correct]
[Checksum Status: Good]
Identifier (BE): 1 (0x0001)
Identifier (LE): 256 (0x0100)
Sequence number (BE): 1 (0x0001)
Sequence number (LE): 256 (0x0100)
[Response frame: 455]

```

c. What is the message identifier and sequence number?

Identifier (BE): 1 (0x0001)

Identifier (LE): 256 (0x0100)

Sequence number (BE): 1 (0x0001)

Sequence number (LE): 256 (0x0100)

(2) Find the **first ICMP Echo Reply** packet.

```

455 4.396176 8.8.8.8 140.117.171.179 ICMP 74 Echo (ping) reply id=0x0001, seq=1/256, ttl=48 (request in 454)

```

a. Examine the **Internet Protocol**. What are the source and destination addresses?

Source: 8.8.8.8

Destination: 140.117.171.179

```

Internet Protocol Version 4, Src: 8.8.8.8, Dst: 140.117.171.179
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
  Total Length: 60
  Identification: 0x0000 (0)
  > Flags: 0x00
  Fragment offset: 0
  Time to live: 48
  Protocol: ICMP (1)
  Header checksum: 0x4289 [validation disabled]
  [Header checksum status: Unverified]
  Source: 8.8.8.8
  Destination: 140.117.171.179

```

b. Next, examine the **Internet Control Message Protocol**. What is the ICMP message type?

Type: 0 (Echo (ping) reply)

```

Internet Control Message Protocol
Type: 0 (Echo (ping) reply)
Code: 0
Checksum: 0x555a [correct]
[Checksum Status: Good]
Identifier (BE): 1 (0x0001)
Identifier (LE): 256 (0x0100)
Sequence number (BE): 1 (0x0001)
Sequence number (LE): 256 (0x0100)
[Request frame: 454]
[Response time: 15.716 ms]

```

2. Traceroute Captured.

```

C:\Users\chen jia ming>tracert 8.8.8.8

在上限 30 個躍點上
追蹤 google-public-dns-a.google.com [8.8.8.8] 的路由:

 1    2 ms    1 ms    1 ms  140.117.162.254
 2    1 ms    1 ms   <1 ms  10.10.129.254
 3   <1 ms   <1 ms   <1 ms  140.117.232.85
 4   <1 ms   <1 ms   <1 ms  140.117.232.33
 5    1 ms    1 ms    1 ms  140.117.232.25
 6    6 ms    6 ms    7 ms  202.169.174.161
 7    7 ms    7 ms    7 ms  72.14.196.229
 8   10 ms   10 ms   10 ms  72.14.233.20
 9   10 ms   10 ms   18 ms  209.85.242.163
10   15 ms   15 ms   15 ms  209.85.243.21
11    *      *      *    要求等候逾時。
12   15 ms   16 ms   15 ms  google-public-dns-a.google.com [8.8.8.8]

追蹤完成。

```

(1) Find the first **ICMP Echo Request** packet.

```

11004 91.124827  140.117.171.179  8.8.8.8  ICMP  106 Echo (ping) request id=0x0001, seq=5/1280, ttl=1 (no response found!)

```

a. Examine the Internet Protocol. What are the source and destination addresses?

Src: 140.117.171.179, Dst: 8.8.8.8

Internet Protocol Version 4, Src: 140.117.171.179, Dst: 8.8.8.8

0100 = Version: 4

.... 0101 = Header Length: 20 bytes (5)

- > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
- Total Length: 92

Identification: 0x2615 (9749)

- > Flags: 0x00

Fragment offset: 0

- > Time to live: 1

Protocol: ICMP (1)

Header checksum: 0x0000 [validation disabled]

[Header checksum status: Unverified]

Source: 140.117.171.179

Destination: 8.8.8.8

- b. What are the protocol type and the Time-to-Live in the IP packet?

Time to live: 1

Protocol: ICMP (1)

- c. Next, examine the **Internet Control Message Protocol**. What is the ICMP message type? What are the message identifier and sequence number?

Identifier (BE): 1 (0x0001)

Identifier (LE): 256 (0x0100)

Sequence number (BE): 5 (0x0005)

Sequence number (LE): 1280 (0x0500)

Internet Control Message Protocol

Type: 8 (Echo (ping) request)

Code: 0

Checksum: 0xf7f9 [correct]

[Checksum Status: Good]

Identifier (BE): 1 (0x0001)

Identifier (LE): 256 (0x0100)

Sequence number (BE): 5 (0x0005)

Sequence number (LE): 1280 (0x0500)

- > [No response seen]

- > Data (64 bytes)

- (2) Find an **ICMP Time-to-live exceeded** packet.

437 3.759229 140.117.162.254 140.117.171.179 ICMP 70 Time-to-live exceeded (Time to live exceeded in transit)

- a. Examine the **Internet Protocol**. What are the source and destination addresses?


```
Source: 140.117.162.254
Destination: 140.117.171.179
[Source GeoIP: Unknown]
[Destination GeoIP: Unknown]
Internet Control Message Protocol
Type: 11 (Time-to-live exceeded)
Code: 0 (Time to live exceeded in transit)
Checksum: 0xf4ff [correct]
```

Source: 140.117.162.254

Destination: 140.117.171.179

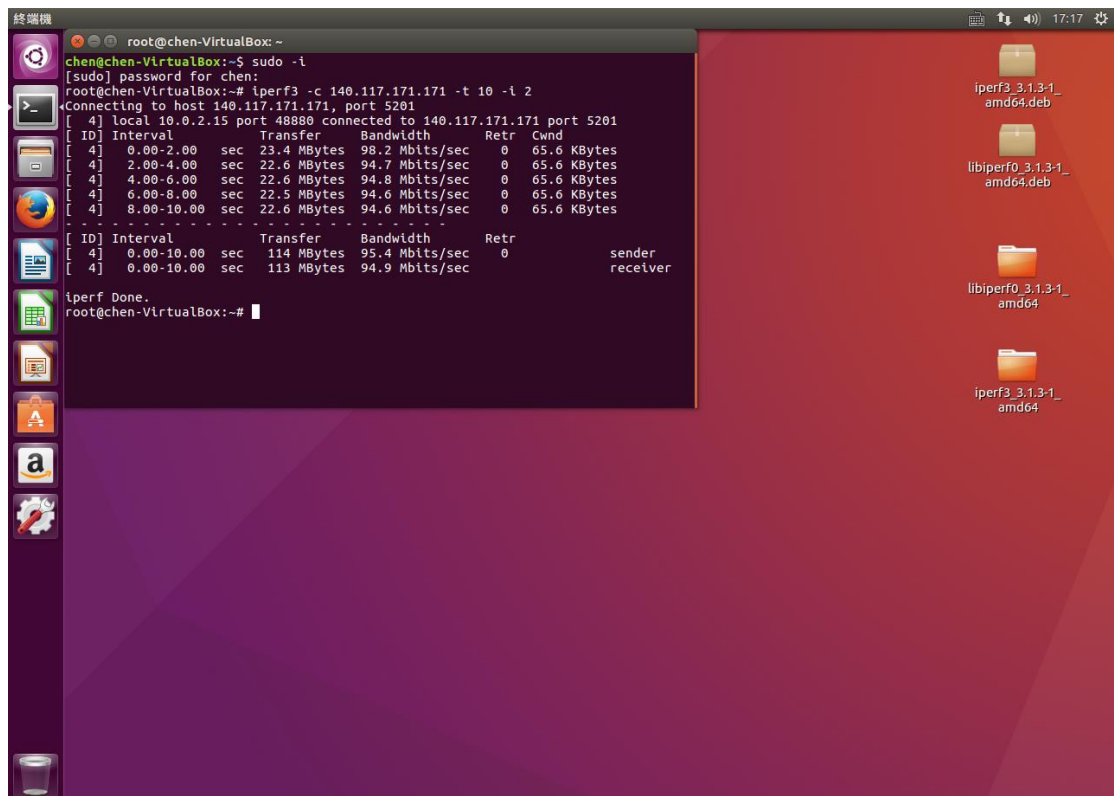
- b. Next, examine the **Internet Control Message Protocol**. What is the ICMP message type?

Type: 11 (Time-to-live exceeded)

Part 3 Measuring Network Bandwidth

1. Measure the bandwidth for **Transmission Control Protocol**

Type "iperf3 -c 140.117.171.171 -t 10 -i 2"



```
root@chen-VirtualBox: ~
chen@chen-VirtualBox:~$ sudo -l
[sudo] password for chen:
root@chen-VirtualBox:~# iperf3 -c 140.117.171.171 -t 10 -i 2
Connecting to host 140.117.171.171, port 5201
[ 4] local 10.0.2.15 port 48880 connected to 140.117.171.171 port 5201
[ ID] Interval      Transfer    Bandwidth  Retr  Cwnd
[ 4] 0.00-2.00 sec  23.4 MBytes 98.2 Mbits/sec    0   65.6 KBytes
[ 4] 2.00-4.00 sec  22.6 MBytes 94.7 Mbits/sec    0   65.6 KBytes
[ 4] 4.00-6.00 sec  22.6 MBytes 94.8 Mbits/sec    0   65.6 KBytes
[ 4] 6.00-8.00 sec  22.5 MBytes 94.6 Mbits/sec    0   65.6 KBytes
[ 4] 8.00-10.00 sec 22.6 MBytes 94.6 Mbits/sec    0   65.6 KBytes
- - - - -
[ ID] Interval      Transfer    Bandwidth  Retr  sender receiver
[ 4] 0.00-10.00 sec  114 MBytes 95.4 Mbits/sec    0
[ 4] 0.00-10.00 sec  113 MBytes 94.9 Mbits/sec    0
iperf Done.
root@chen-VirtualBox:~#
```

2. Adjust the window size for **Transmission Control Protocol**. See what's different.

Type "iperf3 -c 140.117.171.171 -w 2000 -t 10 -i 2"

```
root@chen-VirtualBox: ~  
[ 4] 6.00-8.00 sec 22.5 MBytes 94.6 Mbits/sec 0 65.6 KBytes  
[ 4] 8.00-10.00 sec 22.6 MBytes 94.6 Mbits/sec 0 65.6 KBytes  
-----  
[ ID] Interval      Transfer      Bandwidth      Retr  
[ 4] 0.00-10.00 sec 114 MBytes 95.4 Mbits/sec 0 sender  
[ 4] 0.00-10.00 sec 113 MBytes 94.9 Mbits/sec receiver  
iperf Done.  
root@chen-VirtualBox:~# iperf3 -c 140.117.171.171 -w 2000 -t 10 -i 2  
Connecting to host 140.117.171.171, port 5201  
[ 4] local 10.0.2.15 port 48890 connected to 140.117.171.171 port 5201  
[ ID] Interval      Transfer      Bandwidth      Retr  Cwnd  
[ 4] 0.00-2.00 sec 3.85 MBytes 16.2 Mbits/sec 0 39.9 KBytes  
[ 4] 2.00-4.00 sec 3.51 MBytes 14.7 Mbits/sec 0 39.9 KBytes  
[ 4] 4.00-6.00 sec 3.57 MBytes 15.0 Mbits/sec 0 39.9 KBytes  
[ 4] 6.00-8.00 sec 3.70 MBytes 15.5 Mbits/sec 0 39.9 KBytes  
[ 4] 8.00-10.00 sec 3.51 MBytes 14.7 Mbits/sec 0 39.9 KBytes  
-----  
[ ID] Interval      Transfer      Bandwidth      Retr  
[ 4] 0.00-10.00 sec 18.1 MBytes 15.2 Mbits/sec 0 sender  
[ 4] 0.00-10.00 sec 17.9 MBytes 15.1 Mbits/sec receiver  
iperf Done.  
root@chen-VirtualBox:~#
```

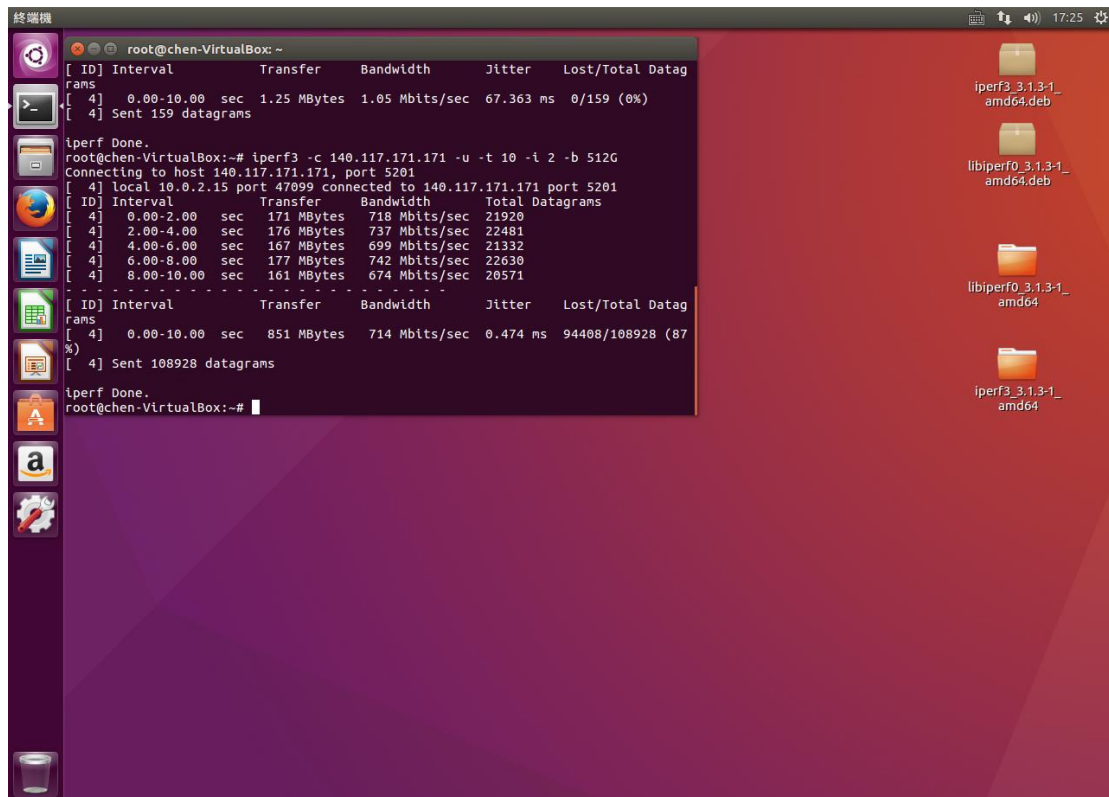
3. Measure the bandwidth for **User Datagram Protocol**

Type "iperf3 -c 140.117.171.171 -u -t 10 -i 2"

```
root@chen-VirtualBox: ~  
[ 4] 8.00-10.00 sec 3.51 MBytes 14.7 Mbits/sec 0 39.9 KBytes  
-----  
[ ID] Interval      Transfer      Bandwidth      Retr  
[ 4] 0.00-10.00 sec 18.1 MBytes 15.2 Mbits/sec 0 sender  
[ 4] 0.00-10.00 sec 17.9 MBytes 15.1 Mbits/sec receiver  
iperf Done.  
root@chen-VirtualBox:~# iperf3 -c 140.117.171.171 -u -t 10 -i 2  
Connecting to host 140.117.171.171, port 5201  
[ 4] local 10.0.2.15 port 34910 connected to 140.117.171.171 port 5201  
[ ID] Interval      Transfer      Bandwidth      Total Datagrams  
[ 4] 0.00-2.00 sec 256 KBytes 1.05 Mbits/sec 32  
[ 4] 2.00-4.00 sec 256 KBytes 1.05 Mbits/sec 32  
[ 4] 4.00-6.00 sec 256 KBytes 1.05 Mbits/sec 32  
[ 4] 6.00-8.00 sec 256 KBytes 1.05 Mbits/sec 32  
[ 4] 8.00-10.00 sec 256 KBytes 1.05 Mbits/sec 32  
-----  
[ ID] Interval      Transfer      Bandwidth      Jitter    Lost/Total Datag  
rams  
[ 4] 0.00-10.00 sec 1.25 MBytes 1.05 Mbits/sec 67.363 ms 0/159 (0%)  
[ 4] Sent 159 datagrams  
iperf Done.  
root@chen-VirtualBox:~#
```

4. Adjust the bandwidth for **User Datagram Protocol**. Measure the package lost rate or any else happened.

Type “iperf3 -c 140.117.171.171 -u -t 10 -i 2 -b 512G”



The screenshot shows a terminal window titled "终端机" (Terminal) with the prompt "root@chen-VirtualBox: ~". The terminal displays the output of an iperf3 test. The first test shows a transfer of 1.25 MBytes at 1.05 Mbits/sec with 0% loss. The second test, initiated by the command "iperf3 -c 140.117.171.171 -u -t 10 -i 2 -b 512G", shows a transfer of 851 MBytes at 714 Mbits/sec with a loss rate of 87% (94408/108928 datagrams lost).

```
root@chen-VirtualBox: ~  
[ ID] Interval      Transfer      Bandwidth      Jitter    Lost/Totl Datagrams  
[  4] 0.00-10.00 sec  1.25 MBytes   1.05 Mbits/sec  67.363 ms  0/159 (0%)  
[  4] Sent 159 datagrams  
  
iperf Done.  
root@chen-VirtualBox:~# iperf3 -c 140.117.171.171 -u -t 10 -i 2 -b 512G  
Connecting to host 140.117.171.171, port 5201  
[  4] local 10.0.2.15 port 47099 connected to 140.117.171.171 port 5201  
[ ID] Interval      Transfer      Bandwidth      Jitter    Total Datagrams  
[  4] 0.00-2.00 sec  171 MBytes    718 Mbits/sec  21920  
[  4] 2.00-4.00 sec  176 MBytes    737 Mbits/sec  22481  
[  4] 4.00-6.00 sec  167 MBytes    699 Mbits/sec  21332  
[  4] 6.00-8.00 sec  177 MBytes    742 Mbits/sec  22630  
[  4] 8.00-10.00 sec 161 MBytes    674 Mbits/sec  20571  
-----  
[ ID] Interval      Transfer      Bandwidth      Jitter    Lost/Totl Datagrams  
[  4] 0.00-10.00 sec  851 MBytes    714 Mbits/sec  0.474 ms  94408/108928 (87%)  
[  4] Sent 108928 datagrams  
  
iperf Done.  
root@chen-VirtualBox:~#
```