

Task 1

1. What does the above ping command do?

The ping command sends a ICMP echo request to the defined server. It uses ICMP protocol's ECHO_REQUEST datagram to elicit an ICMP ECHO_RESPONSE from a host.

The options used in the command are:

-s: Specifies the size of ping packet to be sent. We have set this to 3500 bytes which contains 8 bytes of ICMP header data and 3492 bytes of data.

-c: It defines the count of the number of ping packets to be sent. We have defined 5 so a total of 5 ping packets are sent to the host.

2. How many total IP packets are exchanged in the communication between your host and the remote server representing ping-ams1.online.net ? count fragments

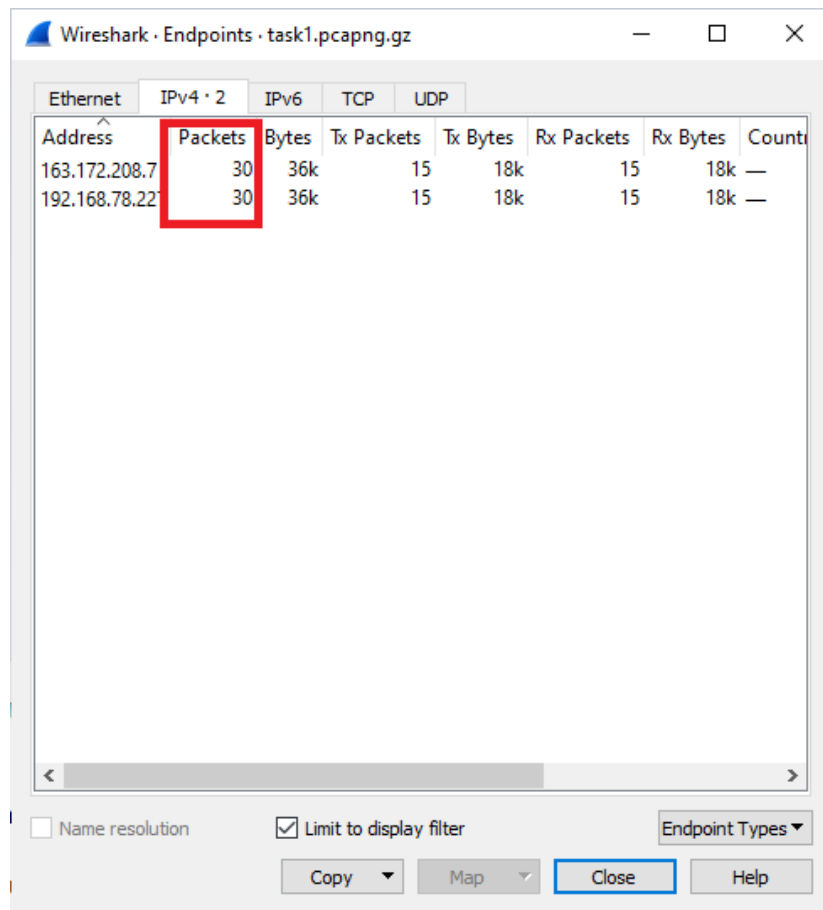
We find the IP address of ping-ams1.online.net is 163.172.208.7. We use the filter *ip.addr == 163.172.208.7* to filter packets from our capture. We find that a total of 30 packets were sent / received during this communication between our host and the server.

The image shows a Wireshark packet capture window titled 'task1.pcapng.gz'. The filter bar at the top displays 'ip.addr == 163.172.208.7'. The packet list shows 30 packets, alternating between ICMP Echo (ping) requests and replies. The details pane for the selected packet (No. 290) shows the following information:

- Header Checksum: 0x3cd9 [validation disabled]
- [Header checksum status: Unverified]
- Source Address: 192.168.78.227
- Destination Address: 163.172.208.7
- 3 IPv4 Fragments (3508 bytes): #271(1480), #272(1480), #273(548)
- [Frame: 271, payload: 0-1479 (1480 bytes)]
- [Frame: 272, payload: 1480-2959 (1480 bytes)]
- [Frame: 273, payload: 2960-3507 (548 bytes)]
- [Fragment count: 3]

The packet bytes pane shows the raw data of the selected packet, including the IPv4 header and the fragmented payload. The status bar at the bottom indicates 'Packets: 491 · Displayed: 30 (6.1%)'.

The same can also be verified by checking the packet count in conversations and endpoints.



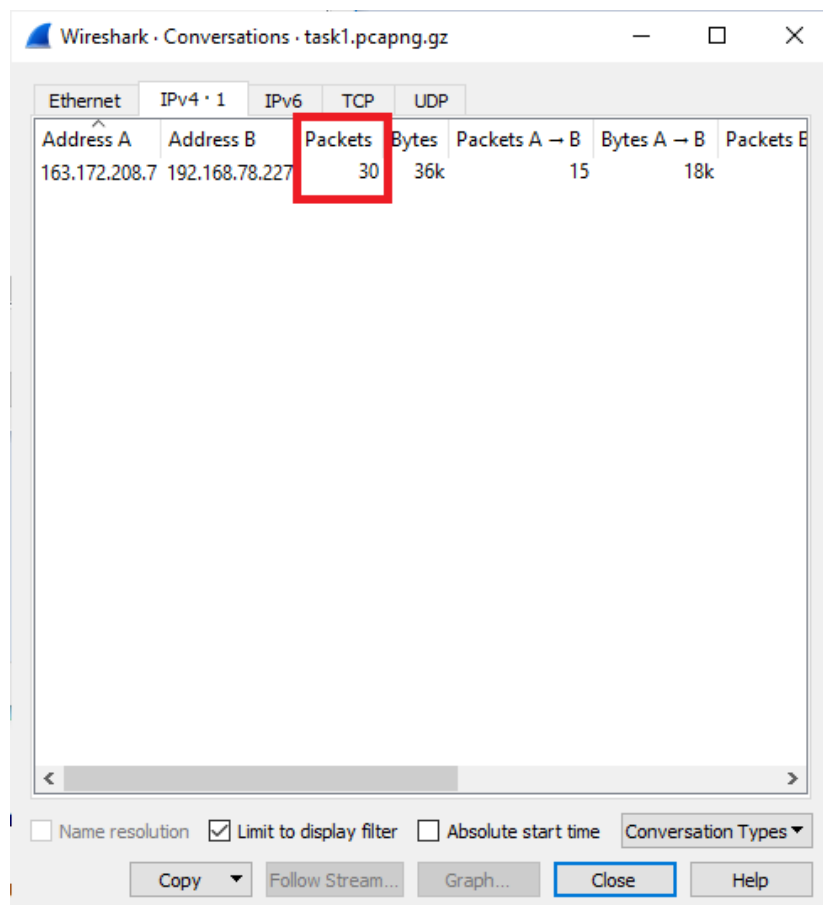
Wireshark · Endpoints · task1.pcapng.gz

Endpoint Types: Ethernet, IPv4 · 2, IPv6, TCP, UDP

Address	Packets	Bytes	Tx Packets	Tx Bytes	Rx Packets	Rx Bytes	Count
163.172.208.7	30	36k	15	18k	15	18k	—
192.168.78.227	30	36k	15	18k	15	18k	—

☐ Name resolution ☒ Limit to display filter Endpoint Types ▼

Copy ▼ Map ▼ Close Help



Wireshark · Conversations · task1.pcapng.gz

Conversation Types: Ethernet, IPv4 · 1, IPv6, TCP, UDP

Address A	Address B	Packets	Bytes	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A
163.172.208.7	192.168.78.227	30	36k	15	18k	15	18k

☐ Name resolution ☒ Limit to display filter ☐ Absolute start time Conversation Types ▼

Copy ▼ Follow Stream... Graph... Close Help

3. What is the size of each ping request sent from your host to remote server ?

We check the data size in a request packet. We find the data size is 3492 bytes. ICMP header of 8 bytes is added over this data. So our final packet size is 3500 bytes which we defined in the ping command.

task1.pcapng.gz

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

ip.addr == 163.172.208.7

No.	Time	Source	Destination	Protocol	Length	Info
237	5.883529734	192.168.78.227	163.172.208.7	IPv4	1516	Fragmented IP protocol (proto=ICMP 1, o
238	5.883531786	192.168.78.227	163.172.208.7	IPv4	1516	Fragmented IP protocol (proto=ICMP 1, o
239	5.883532522	192.168.78.227	163.172.208.7	ICMP	584	Echo (ping) request id=0x0001, seq=1/2
251	6.079725678	163.172.208.7	192.168.78.227	IPv4	1516	Fragmented IP protocol (proto=ICMP 1, o
252	6.079740506	163.172.208.7	192.168.78.227	IPv4	1516	Fragmented IP protocol (proto=ICMP 1, o
253	6.079740556	163.172.208.7	192.168.78.227	ICMP	584	Echo (ping) reply id=0x0001, seq=1/2
271	6.885195169	192.168.78.227	163.172.208.7	IPv4	1516	Fragmented IP protocol (proto=ICMP 1, o
272	6.885204514	192.168.78.227	163.172.208.7	IPv4	1516	Fragmented IP protocol (proto=ICMP 1, o
273	6.885206639	192.168.78.227	163.172.208.7	ICMP	584	Echo (ping) request id=0x0001, seq=2/5
286	7.072173381	163.172.208.7	192.168.78.227	IPv4	1516	Fragmented IP protocol (proto=ICMP 1, o
289	7.072195721	163.172.208.7	192.168.78.227	IPv4	1516	Fragmented IP protocol (proto=ICMP 1, o
290	7.072195766	163.172.208.7	192.168.78.227	ICMP	584	Echo (ping) reply id=0x0001, seq=2/5
315	7.886087832	192.168.78.227	163.172.208.7	IPv4	1516	Fragmented IP protocol (proto=ICMP 1, o
316	7.886091509	192.168.78.227	163.172.208.7	IPv4	1516	Fragmented IP protocol (proto=ICMP 1, o
317	7.886092257	192.168.78.227	163.172.208.7	ICMP	584	Echo (ping) request id=0x0001, seq=3/7
325	8.073347423	163.172.208.7	192.168.78.227	IPv4	1516	Fragmented IP protocol (proto=ICMP 1, o
326	8.073348132	163.172.208.7	192.168.78.227	IPv4	1516	Fragmented IP protocol (proto=ICMP 1, o
327	8.073348209	163.172.208.7	192.168.78.227	ICMP	584	Echo (ping) reply id=0x0001, seq=3/7
339	8.887284227	192.168.78.227	163.172.208.7	IPv4	1516	Fragmented IP protocol (proto=ICMP 1, o
340	8.887287288	192.168.78.227	163.172.208.7	IPv4	1516	Fragmented IP protocol (proto=ICMP 1, o

Identifier (LE): 256 (0x0100)
Sequence Number (BE): 1 (0x0001)
Sequence Number (LE): 256 (0x0100)
[Response frame: 253]
Timestamp from icmp data: Nov 10, 2020 12:07:40.000000000 India Standard Time
[Timestamp from icmp data (relative): 0.50035002 seconds]
Data (3492 bytes)
Data: 66a2070000000000101112131415161718191a1b1c1d1e1f202122232425262728292a2b...
[Length: 3492]

0010 66 a2 07 00 00 00 00 10 11 12 13 14 15 16 17 f.....
0020 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25 26 27 !"#%&'
0030 28 29 2a 2b 2c 2d 2e 2f 30 31 32 33 34 35 36 37 ()*+,-./ 01234567
0040 38 39 3a 3b 3c 3d 3e 3f 40 41 42 43 44 45 46 47 89:;<=> @ABCDEFG
0050 48 49 4a 4b 4c 4d 4e 4f 50 51 52 53 54 55 56 57 HIJKLMNO PQRSTU
0060 58 59 5a 5b 5c 5d 5e 5f 60 61 62 63 64 65 66 67 XYZ[\]^_`abcde

Frame (584 bytes) Reassembled IPv4 (3508 bytes)
Data (data), 3,492 bytes

Packets: 491 · Displayed: 30 (6.1%) Profile: Default

- Make a table for each ping request packet sent from your host to remote, the respective field indicating it, if the request packet is fragmented or not. If packet is fragmented (add details on number of IP fragments and on each fragment), Time of sending each individual fragment/packet, length of the individual fragment/packet, time of receiving ping response, the respective field indicating if response packet is fragmented or not, if response packet is fragmented, include the number of IP fragments, total actual length of data carried by the respective fragment in respective ping request and response.

Request Packet	Field indicating request packet	Details of request packet fragments	Time of receiving ping response	Field indicating if the response is fragmented	Details of response packet fragments	Actual data carried by each fragment.
239	Internet Control	3 fragments	6.079740556	Internet Protocol	3 fragments	Packet 237: 1480 bytes

	Message Protocol , Type : 8 (Echo(ping) request)	<p>Packet 237: Time – 5.883529734, Length – 1500 bytes</p> <p>Packet 238: Time – 5.883531786, Length – 1500 bytes</p> <p>Packet 239: Time – 5.883532522, Length – 568 bytes</p>		Version 4, IPv4 Fragments	Packet 251, 252 and 253	<p>Packet 238: 1480 bytes</p> <p>Packet 239: 548 bytes</p> <p>Packet 251: 1480 bytes</p> <p>Packet 252: 1480 bytes</p> <p>Packet 253: 548 bytes</p>
273	Internet Control Message Protocol , Type : 8 (Echo(ping) request)	<p>3 fragments</p> <p>Packet 271: Time – 6.885195169, Length – 1500 bytes</p> <p>Packet 272: Time – 6.885204514, Length – 1500 bytes</p> <p>Packet 273: Time – 6.885206639, Length – 568 bytes</p>	7.072195766	Internet Protocol Version 4, IPv4 Fragments	3 fragments Packet 286, 289 and 290	<p>Packet 271: 1480 bytes</p> <p>Packet 272: 1480 bytes</p> <p>Packet 273: 548 bytes</p> <p>Packet 286: 1480 bytes</p> <p>Packet 289: 1480 bytes</p> <p>Packet 290: 548 bytes</p>
317	Internet Control Message Protocol , Type : 8 (Echo(ping) request)	<p>3 fragments</p> <p>Packet 315: Time – 7.886087832, Length – 1500 bytes</p> <p>Packet 316: Time – 7.886091509, Length – 1500</p> <p>Packet 317: Time – 7.886092257, Length – 568 bytes</p>	8.073348209	Internet Protocol Version 4, IPv4 Fragments	3 fragments Packet 325, 326 and 327	<p>Packet 315: 1480 bytes</p> <p>Packet 316: 1480 bytes</p> <p>Packet 317: 548 bytes</p> <p>Packet 325: 1480 bytes</p> <p>Packet 326: 1480 bytes</p> <p>Packet 327: 548 bytes</p>

341	Internet Control Message Protocol , Type : 8 (Echo(ping) request)	3 fragments Packet 339: Time – 8.887284227, Length – 1500 bytes Packet 340: Time – 8.887287288, Length – 1500 bytes Packet 341: Time – 8.8872877781, Length – 568 bytes	9.066614160	Internet Protocol Version 4, IPv4 Fragments	3 fragments Packet 351, 352 and 353	Packet 339: 1480 bytes Packet 340: 1480 bytes Packet 341: 548 bytes Packet 351: 1480 bytes Packet 352: 1480 bytes Packet 353: 548 bytes
367	Internet Control Message Protocol , Type : 8 (Echo(ping) request)	3 fragments Packet 365: Time – 9.88323331, Length – 1500 bytes Packet 366: Time – 9.88326878, Length – 1500 bytes Packet 367: Time – 9.88327392, Length – 568 bytes	10.068313497	Internet Protocol Version 4, IPv4 Fragments	3 fragments Packet 375, 376 and 377	Packet 365: 1480 bytes Packet 366: 1480 bytes Packet 367: 548 bytes Packet 375: 1480 bytes Packet 376: 1480 bytes Packet 377: 548 bytes

5. Pick any fragmented ping request and response used in question #4. Explain how you find the length of actual data in individual fragments of the associated ping request and response ? Where is the total/final length of the respective ping request and response at IP level visible in Wireshark ?

We can check the Data field in Wireshark which shows the actual data after removing headers. For packet 237, we see the data is 1480 bytes.

task1.pcapng.gz

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

ip.addr == 163.172.208.7

No.	Time	Source	Destination	Protocol	Length	Info
237	5.883529734	192.168.78.227	163.172.208.7	IPv4	1516	Fragmented IP protocol (proto=ICMP 1, o
238	5.883531786	192.168.78.227	163.172.208.7	IPv4	1516	Fragmented IP protocol (proto=ICMP 1, o
239	5.883532522	192.168.78.227	163.172.208.7	ICMP	584	Echo (ping) request id=0x0001, seq=1/2
251	6.079725678	163.172.208.7	192.168.78.227	IPv4	1516	Fragmented IP protocol (proto=ICMP 1, o
252	6.079740506	163.172.208.7	192.168.78.227	IPv4	1516	Fragmented IP protocol (proto=ICMP 1, o
253	6.079740556	163.172.208.7	192.168.78.227	ICMP	584	Echo (ping) reply id=0x0001, seq=1/2
271	6.885195169	192.168.78.227	163.172.208.7	IPv4	1516	Fragmented IP protocol (proto=ICMP 1, o
272	6.885204514	192.168.78.227	163.172.208.7	IPv4	1516	Fragmented IP protocol (proto=ICMP 1, o
273	6.885206639	192.168.78.227	163.172.208.7	ICMP	584	Echo (ping) request id=0x0001, seq=2/5
286	7.072173381	163.172.208.7	192.168.78.227	IPv4	1516	Fragmented IP protocol (proto=ICMP 1, o
289	7.072195721	163.172.208.7	192.168.78.227	IPv4	1516	Fragmented IP protocol (proto=ICMP 1, o
290	7.072195766	163.172.208.7	192.168.78.227	ICMP	584	Echo (ping) reply id=0x0001, seq=2/5
315	7.886087832	192.168.78.227	163.172.208.7	IPv4	1516	Fragmented IP protocol (proto=ICMP 1, o
316	7.886091509	192.168.78.227	163.172.208.7	IPv4	1516	Fragmented IP protocol (proto=ICMP 1, o
317	7.886092257	192.168.78.227	163.172.208.7	ICMP	584	Echo (ping) request id=0x0001, seq=3/7
325	8.073347423	163.172.208.7	192.168.78.227	IPv4	1516	Fragmented IP protocol (proto=ICMP 1, o
326	8.073348132	163.172.208.7	192.168.78.227	IPv4	1516	Fragmented IP protocol (proto=ICMP 1, o
327	8.073348209	163.172.208.7	192.168.78.227	ICMP	584	Echo (ping) reply id=0x0001, seq=3/7
339	8.887284227	192.168.78.227	163.172.208.7	IPv4	1516	Fragmented IP protocol (proto=ICMP 1, o

> Frame 237: 1516 bytes on wire (12128 bits), 1516 bytes captured (12128 bits) on interface any, id 0

> Linux cooked capture v1

> Internet Protocol Version 4, Src: 192.168.78.227, Dst: 163.172.208.7

▼ Data (1480 bytes)

Data: 080075c9000100013435aa5f0000000066a207000000000101112131415161718191a1b...

[Length: 1480]

```

0000  00 04 00 01 00 06 70 85 c2 3c 36 3a 43 b0 08 00  .....p.  <6:C...
0010  45 00 05 dc b6 ee 20 00 40 01 1a f3 c0 a8 4e e3  E..... @...N.
0020  a3 ac d0 07 08 00 75 c9 00 01 00 01 34 35 aa 5f  .....u. ....45.
0030  00 00 00 00 66 a2 07 00 00 00 00 00 10 11 12 13  ....f.....
0040  14 15 16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23  ..... !"#
0050  24 25 26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 32 33  $%&'()*+ ,-. /0123
0060  34 35 36 37 38 39 3a 3b 3c 3d 3e 3f 40 41 42 43  456789:;<=>?@ABC

```

Bytes 28-31: Source Address (ip.src) | Packets: 491 • Displayed: 30 (6.1%) | Profile: Default

Similarly we can check for other fragments as well. We find that the last fragment shows the total of all fragments. To view the data in last fragment we check the IPv4 fragments field where it shows payload size of all fragments.

We can check IPv4 fragments under Internet Protocol version 4 for any ping request or response. It shows us the fragments and the data transmitted in each fragment.

task1.pcapng.gz

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

ip.addr == 163.172.208.7

Source	Destination	Protocol	Length	Info
192.168.78.227	163.172.208.7	IPv4	1516	Fragmented IP protocol (proto=ICMP 1, off=0, ID=b6ee) [Reasse...
192.168.78.227	163.172.208.7	IPv4	1516	Fragmented IP protocol (proto=ICMP 1, off=1480, ID=b6ee) [Rea...
192.168.78.227	163.172.208.7	ICMP	584	Echo (ping) request id=0x0001, seq=1/256, ttl=64 (reply in 2...
163.172.208.7	192.168.78.227	IPv4	1516	Fragmented IP protocol (proto=ICMP 1, off=0, ID=d8f5) [Reasse...
163.172.208.7	192.168.78.227	IPv4	1516	Fragmented IP protocol (proto=ICMP 1, off=1480, ID=d8f5) [Rea...
163.172.208.7	192.168.78.227	ICMP	584	Echo (ping) reply id=0x0001, seq=1/256, ttl=53 (request in...
192.168.78.227	163.172.208.7	IPv4	1516	Fragmented IP protocol (proto=ICMP 1, off=0, ID=b73a) [Reasse...
192.168.78.227	163.172.208.7	IPv4	1516	Fragmented IP protocol (proto=ICMP 1, off=1480, ID=b73a) [Rea...
192.168.78.227	163.172.208.7	ICMP	584	Echo (ping) request id=0x0001, seq=2/512, ttl=64 (reply in 2...
163.172.208.7	192.168.78.227	IPv4	1516	Fragmented IP protocol (proto=ICMP 1, off=0, ID=d9c0) [Reasse...
163.172.208.7	192.168.78.227	IPv4	1516	Fragmented IP protocol (proto=ICMP 1, off=1480, ID=d9c0) [Rea...
163.172.208.7	192.168.78.227	ICMP	584	Echo (ping) reply id=0x0001, seq=2/512, ttl=53 (request in...
192.168.78.227	163.172.208.7	IPv4	1516	Fragmented IP protocol (proto=ICMP 1, off=0, ID=b7fa) [Reasse...
192.168.78.227	163.172.208.7	IPv4	1516	Fragmented IP protocol (proto=ICMP 1, off=1480, ID=b7fa) [Rea...
192.168.78.227	163.172.208.7	ICMP	584	Echo (ping) request id=0x0001, seq=3/768, ttl=64 (reply in 3...
163.172.208.7	192.168.78.227	IPv4	1516	Fragmented IP protocol (proto=ICMP 1, off=0, ID=da24) [Reasse...
163.172.208.7	192.168.78.227	IPv4	1516	Fragmented IP protocol (proto=ICMP 1, off=1480, ID=da24) [Rea...
163.172.208.7	192.168.78.227	ICMP	584	Echo (ping) reply id=0x0001, seq=3/768, ttl=53 (request in...
192.168.78.227	163.172.208.7	IPv4	1516	Fragmented IP protocol (proto=ICMP 1, off=0, ID=b8cb) [Reasse...

Source Address: 192.168.78.227
Destination Address: 163.172.208.7

[3 IPv4 Fragments (3508 bytes): #237(1480), #238(1480), #239(548)]

[Frame: 237, payload: 0-1479 (1480 bytes)]
[Frame: 238, payload: 1480-2959 (1480 bytes)]
[Frame: 239, payload: 2960-3507 (548 bytes)]

[Fragment count: 3]
[Reassembled IPv4 length: 3508]
[Reassembled IPv4 data: 080075c9000100013435aa5f000000066a20700000000101112131415161718191a1b...]

Internet Control Message Protocol

0000	08 00 75 c9 00 01 00 01 34 35 aa 5f 00 00 00 00	..u..... 45_....
0010	66 a2 07 00 00 00 00 00 10 11 12 13 14 15 16 17	f.....
0020	18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25 26 27 !"#\$%&'
0030	28 29 2a 2b 2c 2d 2e 2f 30 31 32 33 34 35 36 37	()*+,.-/ 01234567
0040	38 39 3a 3b 3c 3d 3e 3f 40 41 42 43 44 45 46 47	89:;<=> @ABCDEFGH
0050	48 49 4a 4b 4c 4d 4e 4f 50 51 52 53 54 55 56 57	IJKLMNOPQRSTUVWXYZ

Frame (584 bytes) Reassembled IPv4 (3508 bytes)

IPv4 Fragments (ip.fragments), 3,508 bytes

Packets: 491 · Displayed: 30 (6.1%) Profile: Default

We can also check individual fragments and can verify the same data is transmitted in each fragment.

task1.pcapng.gz

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

ip.addr == 163.172.208.7

Source	Destination	Protocol	Length	Info
192.168.78.227	163.172.208.7	IPv4	1516	Fragmented IP protocol (proto=ICMP 1, off=0, ID=b6ee) [Reasse...
192.168.78.227	163.172.208.7	IPv4	1516	Fragmented IP protocol (proto=ICMP 1, off=1480, ID=b6ee) [Rea...
192.168.78.227	163.172.208.7	ICMP	584	Echo (ping) request id=0x0001, seq=1/256, ttl=64 (reply in 2...
163.172.208.7	192.168.78.227	IPv4	1516	Fragmented IP protocol (proto=ICMP 1, off=0, ID=d8f5) [Reasse...
163.172.208.7	192.168.78.227	IPv4	1516	Fragmented IP protocol (proto=ICMP 1, off=1480, ID=d8f5) [Rea...
163.172.208.7	192.168.78.227	ICMP	584	Echo (ping) reply id=0x0001, seq=1/256, ttl=53 (request in...
192.168.78.227	163.172.208.7	IPv4	1516	Fragmented IP protocol (proto=ICMP 1, off=0, ID=b73a) [Reasse...
192.168.78.227	163.172.208.7	IPv4	1516	Fragmented IP protocol (proto=ICMP 1, off=1480, ID=b73a) [Rea...
192.168.78.227	163.172.208.7	ICMP	584	Echo (ping) request id=0x0001, seq=2/512, ttl=64 (reply in 2...
163.172.208.7	192.168.78.227	IPv4	1516	Fragmented IP protocol (proto=ICMP 1, off=0, ID=d9c0) [Reasse...
163.172.208.7	192.168.78.227	IPv4	1516	Fragmented IP protocol (proto=ICMP 1, off=1480, ID=d9c0) [Rea...
163.172.208.7	192.168.78.227	ICMP	584	Echo (ping) reply id=0x0001, seq=2/512, ttl=53 (request in...
192.168.78.227	163.172.208.7	IPv4	1516	Fragmented IP protocol (proto=ICMP 1, off=0, ID=b7fa) [Reasse...
192.168.78.227	163.172.208.7	IPv4	1516	Fragmented IP protocol (proto=ICMP 1, off=1480, ID=b7fa) [Rea...
192.168.78.227	163.172.208.7	ICMP	584	Echo (ping) request id=0x0001, seq=3/768, ttl=64 (reply in 3...
163.172.208.7	192.168.78.227	IPv4	1516	Fragmented IP protocol (proto=ICMP 1, off=0, ID=da24) [Reasse...
163.172.208.7	192.168.78.227	IPv4	1516	Fragmented IP protocol (proto=ICMP 1, off=1480, ID=da24) [Rea...
163.172.208.7	192.168.78.227	ICMP	584	Echo (ping) reply id=0x0001, seq=3/768, ttl=53 (request in...
192.168.78.227	163.172.208.7	IPv4	1516	Fragmented IP protocol (proto=ICMP 1, off=0, ID=b8cb) [Reasse...

Time to Live: 64
 Protocol: ICMP (1)
 Header Checksum: 0x1af3 [validation disabled]
 [Header checksum status: Unverified]
 Source Address: 192.168.78.227
 Destination Address: 163.172.208.7
 [Reassembled IPv4 in frame: 239]

Data (1480 bytes)
 Data: 080075c9000100013435aa5f0000000066a2070000000000101112131415161718191a1b...
 [Length: 1480]

0000 00 04 00 01 00 06 70 85 c2 3c 36 3a 43 b0 08 00p...<6:C...
 0010 45 00 05 dc b6 ee 20 00 40 01 1a f3 c0 a8 4e e3 E.....@.....N.
 0020 a3 ac d0 07 08 00 75 c9 00 01 00 01 34 35 aa 5fu.....45..
 0030 00 00 00 00 66 a2 07 00 00 00 00 00 10 11 12 13f.....
 0040 14 15 16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23!""#
 0050 24 25 26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 32 33-./0123
 0060 34 35 36 37 38 39 3a 3b 3c 3d 3e 3f 40 41 42 43 44 45 46 47 48 49 4a 4b 4c 4d 4e 4f 50 51 52 53 54 55 56 57 58 59 5a 5b 5c 5d 5e 5f 60 61 62 63 64 65 66 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75 76 77 78 79 7a 7b 7c 7d 7e 7f 80 81 82 83 84 85 86 87 88 89 8a 8b 8c 8d 8e 8f 90 91 92 93 94 95 96 97 98 99 9a 9b 9c 9d 9e 9f a0 a1 a2 a3 a4 a5 a6 a7 a8 a9 aa ab ac ad ae af b0 b1 b2 b3 b4 b5 b6 b7 b8 b9 ba bb bc bd be bf c0 c1 c2 c3 c4 c5 c6 c7 c8 c9 ca cb cc cd ce cf d0 d1 d2 d3 d4 d5 d6 d7 d8 d9 da db dc dd de df e0 e1 e2 e3 e4 e5 e6 e7 e8 e9 ea eb ec ed ee ef f0 f1 f2 f3 f4 f5 f6 f7 f8 f9 fa fb fc fd fe ff

task1.pcapng.gz C:\Users\visha\Documents... Packets: 491 · Displayed: 30 (6.1%) Profile: Default

We can add the payload data in these fragments to get the final size of request packet sent. Adding the three fragments we get $1480 + 1480 + 548 = 3508$ bytes.

The final size of request is 3508 bytes at the IP level which is also shown in Reassembled IPv4 length.

Similarly for reply packet,

task1.pcapng.gz

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

ip.addr == 163.172.208.7

Source	Destination	Protocol	Length	Info
192.168.78.227	163.172.208.7	IPv4	1516	Fragmented IP protocol (proto=ICMP 1, off=0, ID=b6ee) [Reasse...
192.168.78.227	163.172.208.7	IPv4	1516	Fragmented IP protocol (proto=ICMP 1, off=1480, ID=b6ee) [Rea...
192.168.78.227	163.172.208.7	ICMP	584	Echo (ping) request id=0x0001, seq=1/256, ttl=64 (reply in 2...
163.172.208.7	192.168.78.227	IPv4	1516	Fragmented IP protocol (proto=ICMP 1, off=0, ID=d8f5) [Reasse...
163.172.208.7	192.168.78.227	IPv4	1516	Fragmented IP protocol (proto=ICMP 1, off=1480, ID=d8f5) [Rea...
163.172.208.7	192.168.78.227	ICMP	584	Echo (ping) reply id=0x0001, seq=1/256, ttl=53 (request in...
192.168.78.227	163.172.208.7	IPv4	1516	Fragmented IP protocol (proto=ICMP 1, off=0, ID=b73a) [Reasse...
192.168.78.227	163.172.208.7	IPv4	1516	Fragmented IP protocol (proto=ICMP 1, off=1480, ID=b73a) [Rea...
192.168.78.227	163.172.208.7	ICMP	584	Echo (ping) request id=0x0001, seq=2/512, ttl=64 (reply in 2...
163.172.208.7	192.168.78.227	IPv4	1516	Fragmented IP protocol (proto=ICMP 1, off=0, ID=d9c0) [Reasse...
163.172.208.7	192.168.78.227	IPv4	1516	Fragmented IP protocol (proto=ICMP 1, off=1480, ID=d9c0) [Rea...
163.172.208.7	192.168.78.227	ICMP	584	Echo (ping) reply id=0x0001, seq=2/512, ttl=53 (request in...
192.168.78.227	163.172.208.7	IPv4	1516	Fragmented IP protocol (proto=ICMP 1, off=0, ID=b7fa) [Reasse...
192.168.78.227	163.172.208.7	IPv4	1516	Fragmented IP protocol (proto=ICMP 1, off=1480, ID=b7fa) [Rea...
192.168.78.227	163.172.208.7	ICMP	584	Echo (ping) request id=0x0001, seq=3/768, ttl=64 (reply in 3...
163.172.208.7	192.168.78.227	IPv4	1516	Fragmented IP protocol (proto=ICMP 1, off=0, ID=da24) [Reasse...
163.172.208.7	192.168.78.227	IPv4	1516	Fragmented IP protocol (proto=ICMP 1, off=1480, ID=da24) [Rea...
163.172.208.7	192.168.78.227	ICMP	584	Echo (ping) reply id=0x0001, seq=3/768, ttl=53 (request in...
192.168.78.227	163.172.208.7	IPv4	1516	Fragmented IP protocol (proto=ICMP 1, off=0, ID=b8cb) [Reasse...

[3 IPv4 Fragments (3508 bytes): #251(1480), #252(1480), #253(548)]

[Frame: 251, payload: 0-1479 (1480 bytes)]

[Frame: 252, payload: 1480-2959 (1480 bytes)]

[Frame: 253, payload: 2960-3507 (548 bytes)]

[Fragment count: 3]

[Reassembled IPv4 length: 3508]

[Reassembled IPv4 data: 00007dc9000100013435aa5f0000000066a207000000000101112131415161718191a1b...]

Internet Control Message Protocol

Type: 0 (Echo (ping) reply)

Code: 0

0000 00 00 7d c9 00 01 00 01 34 35 aa 5f 00 00 00 00 ..}.....45.....

0010 66 a2 07 00 00 00 00 00 10 11 12 13 14 15 16 17 f.....

0020 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25 26 27!"#\$%&'

0030 28 29 2a 2b 2c 2d 2e 2f 30 31 32 33 34 35 36 37 ()*+,-./ 01234567

0040 38 39 3a 3b 3c 3d 3e 3f 40 41 42 43 44 45 46 47 89:;<=> @ABCDEFGH

0050 48 49 4a 4b 4c 4d 4e 4f 50 51 52 53 54 55 56 57 HIJKLMNOPQRSTUVWXYZ

Frame (584 bytes) Reassembled IPv4 (3508 bytes)

IPv4 Fragments (ip.fragments), 3,508 bytes

Packets: 491 · Displayed: 30 (6.1%) Profile: Default

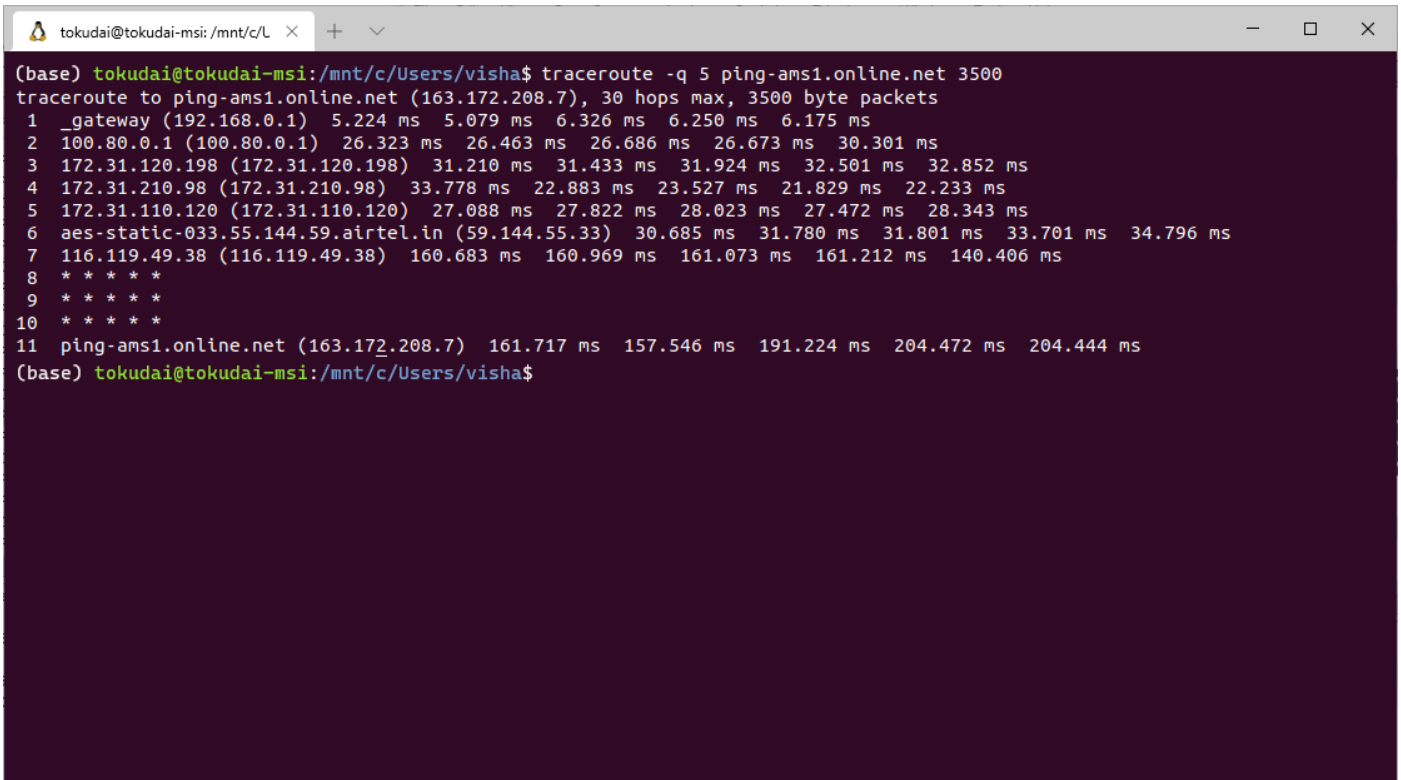
We can add the fragment sizes to get 1480 + 1480 + 548 bytes of data at IP level.

- In the saved file used for analysis, export only those IP packets involved in either direction (in communication) using appropriate Wireshark display filters and save it to another file to upload during submission.

File saved as task1_filtered.pcapng.gz

Task 2

1. What is the IP address of **ping-ams1.online.net**?



```
(base) tokudai@tokudai-msi:/mnt/c/L X + v
(base) tokudai@tokudai-msi:/mnt/c/Users/visha$ traceroute -q 5 ping-ams1.online.net 3500
traceroute to ping-ams1.online.net (163.172.208.7), 30 hops max, 3500 byte packets
 1  _gateway (192.168.0.1)  5.224 ms  5.079 ms  6.326 ms  6.250 ms  6.175 ms
 2  100.80.0.1 (100.80.0.1)  26.323 ms  26.463 ms  26.686 ms  26.673 ms  30.301 ms
 3  172.31.120.198 (172.31.120.198)  31.210 ms  31.433 ms  31.924 ms  32.501 ms  32.852 ms
 4  172.31.210.98 (172.31.210.98)  33.778 ms  22.883 ms  23.527 ms  21.829 ms  22.233 ms
 5  172.31.110.120 (172.31.110.120)  27.088 ms  27.822 ms  28.023 ms  27.472 ms  28.343 ms
 6  aes-static-033.55.144.59.airtel.in (59.144.55.33)  30.685 ms  31.780 ms  31.801 ms  33.701 ms  34.796 ms
 7  116.119.49.38 (116.119.49.38)  160.683 ms  160.969 ms  161.073 ms  161.212 ms  140.406 ms
 8  * * * * *
 9  * * * * *
10  * * * * *
11  ping-ams1.online.net (163.172.208.7)  161.717 ms  157.546 ms  191.224 ms  204.472 ms  204.444 ms
(base) tokudai@tokudai-msi:/mnt/c/Users/visha$
```

From the traceroute screenshot given above; we can see the IP address for ping-ams1.online.net is 163.172.208.7

2. How many hops are involved in finding the route to this **ping-ams1.online.net**?

There are 11 hops involved in finding the route to ping-ams1.online.net where 11th hop is the destination we intend to reach.

3. How many total IP packets are exchanged in the communication to get the final traceroute output of **ping-ams1.online.net** ? How many of them are sent from client to remote machine (server/router) ? How many of them are sent from the remote machine (hop/server/router) to the local client ? Tabulate this with an entry for a router/server and the client too.

A total of 205 packets are exchanged to get the final output of traceroute. We do this by searching all the packets sent to / received from ping-ams1.online.net using *ip.addr == 162.172.208.7* filter. This is shown in the screenshot given below.

The image shows a Wireshark packet capture window titled 'task2.pcapng.gz'. The top menu bar includes File, Edit, View, Go, Capture, Analyze, Statistics, Telephony, Wireless, Tools, and Help. Below the menu is a toolbar with various icons. A filter bar at the top shows 'ip.addr == 163.172.208.7'. The main packet list displays 67 packets, all from source 192.168.0.109 to destination 163.172.208.7. The packets are a mix of IPv4 and UDP, with many being fragmented. Packet 105 is selected, and its details are shown in the lower pane. The details pane shows the following structure:

- > Frame 105: 184 bytes on wire (1472 bits), 184 bytes captured (1472 bits) on interface any, id 0
- > Linux cooked capture v1
- > Internet Protocol Version 4, Src: 172.31.120.198, Dst: 192.168.0.109
- > Internet Control Message Protocol
- > Data (112 bytes)

The bottom pane shows the raw packet data in hexadecimal and ASCII. The ASCII column shows the following text:

```

.....X..N....
E...@...Y...x
...m.....E...
2M.....m....
.....4 @ABCDEFGH
HIJKLMNO PQRSTUW
XYZ[\]^_`abcdefg
hijklmno pqrstuvw

```

At the bottom right, a status bar indicates 'Packets: 345 · Displayed: 205 (59.4%)' and 'Profile: Default'.

- Why and how does the hop/router involved send the response to the packet sent by your client machine ?

The IP protocol has a TTL (time to live) field in header of IPv4 packet which describes how many hops the packet can take before being discarded. When a packet reaches end of TTL, it is dropped and a response is sent from the dropping router/hop to the client which sent the packet.

For traceroute, we increase the TTL by 1 for every next hop. When the packet is dropped the, the router will send a response which will contains its IP address in source field. This way we can know the hops through which the packet passes.

- Which upper layer protocol is used in sending the packet from local client to remote machine ? Which upper layer protocol is used in sending the packet from remote server/router to local client ? Identify within the protocol the type/name of this message (within the protocol) from server to client?

Within the header, the value in the upper layer protocol field from client to server is IP,UDP,Payload. The value in the upper layer protocol field from server to client is IP,ICMP,Payload.

6. Which fields in the IP datagram always change from one datagram to the next within this series of IP packets sent by your host/client ? Which fields stay constant? Which of the fields must stay constant? Which fields must change? Why?

The fields that always change in IP datagram are:

Identification – All IP packets must have a different ID.

Time to live – Traceroute increments the TTL for each subsequent packet to find the next hop.

Header checksum – Since the header changes for every packet, so the checksum needs to change for each packet.

The fields that change for every packet are:

Version – Since we are using IPv4 for for all packets, the version should remain same for all packets.

Header length – Since all the packets are ICMP packets, the header length always remains constant.

Source IP – Since all packets are being sent from our client, the source IP will remain same for all packets sent from our system.

Destination IP – Since all packets are being sent to final destination we need to reach, albeit with different TTL, the destination IP address should remain same for all packets.

Upper Layer Protocol – Since the entire conversation uses ICMP packets, the upper layer protocol should remain same for all packets.

7. Describe the pattern you see in the values in the Identification field of the IP datagram both from client to server and hop/router/server to your client?

We observe that the Identification flag under Internet Protocol version 4 under Internet control message protocol is incremented by 1 for every response. When the response comes from a new hop, the value may increase by more than 1.

task2.pcapng.gz

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

ip.addr == 163.172.208.7

No.	Time	Source	Destination	Protocol	Length	Info
84	2.509214195	192.168.0.109	163.172.208.7	IPv4	1516	Fragmented IP protocol (proto=UDP 17, off=0, ID=3246)
85	2.509223272	192.168.0.109	163.172.208.7	IPv4	1516	Fragmented IP protocol (proto=UDP 17, off=1480, ID=3246)
86	2.509230539	192.168.0.109	163.172.208.7	UDP	556	44657 → 33447 Len=3472
87	2.509285960	192.168.0.109	163.172.208.7	IPv4	1516	Fragmented IP protocol (proto=UDP 17, off=0, ID=3246)
88	2.509295106	192.168.0.109	163.172.208.7	IPv4	1516	Fragmented IP protocol (proto=UDP 17, off=1480, ID=3246)
89	2.509302494	192.168.0.109	163.172.208.7	UDP	556	33481 → 33448 Len=3472
90	2.509363728	192.168.0.109	163.172.208.7	IPv4	1516	Fragmented IP protocol (proto=UDP 17, off=0, ID=3246)
91	2.509372882	192.168.0.109	163.172.208.7	IPv4	1516	Fragmented IP protocol (proto=UDP 17, off=1480, ID=3246)
92	2.509380293	192.168.0.109	163.172.208.7	UDP	556	49717 → 33449 Len=3472
93	2.513231108	192.168.0.1	192.168.0.109	ICMP	72	Time-to-live exceeded (Time to live exceeded in transit)
94	2.513231760	192.168.0.1	192.168.0.109	ICMP	72	Time-to-live exceeded (Time to live exceeded in transit)
97	2.514558428	192.168.0.1	192.168.0.109	ICMP	72	Time-to-live exceeded (Time to live exceeded in transit)
98	2.514558921	192.168.0.1	192.168.0.109	ICMP	72	Time-to-live exceeded (Time to live exceeded in transit)
99	2.514559058	192.168.0.1	192.168.0.109	ICMP	72	Time-to-live exceeded (Time to live exceeded in transit)
100	2.534782661	100.80.0.1	192.168.0.109	ICMP	72	Time-to-live exceeded (Time to live exceeded in transit)
101	2.534993874	100.80.0.1	192.168.0.109	ICMP	72	Time-to-live exceeded (Time to live exceeded in transit)
102	2.535290761	100.80.0.1	192.168.0.109	ICMP	72	Time-to-live exceeded (Time to live exceeded in transit)
103	2.535355553	100.80.0.1	192.168.0.109	ICMP	72	Time-to-live exceeded (Time to live exceeded in transit)
104	2.539086907	100.80.0.1	192.168.0.109	ICMP	72	Time-to-live exceeded (Time to live exceeded in transit)
105	2.540129623	172.31.120.198	192.168.0.109	ICMP	184	Time-to-live exceeded (Time to live exceeded in transit)
106	2.540467819	172.31.120.198	192.168.0.109	ICMP	184	Time-to-live exceeded (Time to live exceeded in transit)
107	2.541053498	172.31.120.198	192.168.0.109	ICMP	184	Time-to-live exceeded (Time to live exceeded in transit)
108	2.541705433	172.31.120.198	192.168.0.109	ICMP	184	Time-to-live exceeded (Time to live exceeded in transit)

Unused: 00000000

Internet Protocol Version 4, Src: 192.168.0.109, Dst: 163.172.208.7

0100 = Version: 4

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

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

Total Length: 1500

Identification: 0x3246 (12870)

Flags: 0x20, more fragments

Fragment Offset: 0

> Time to Live: 1

0000 00 00 00 01 00 06 58 d9 d5 ea 4e c0 00 00 08 00X..N....

0010 45 00 00 38 5a 67 00 00 40 01 9e 9f c0 a8 00 01 E..8Zg..@.....

0020 c0 a8 00 6d 0b 00 3c 39 00 00 00 00 45 00 05 dc ...m...<9....E...

0030 32 46 20 00 01 11 2d 02 c0 a8 00 6d a3 ac d0 07 2f.....m....

0040 e8 24 82 9d 0d 98 40 6c ..\$....@l

Identification (ip.id), 2 bytes

Packets: 345 · Displayed: 205 (59.4%)

Profile: Default

8. Make a table with an entry for each request sent from your client/host, listing what is the value in the IP Identification field and the TTL field for the request and respective response (Include entries if the packet is fragmented). Do these values remain unchanged for all of the replies sent to your computer by the respective (hop) router? If yes why? If not why?

A few values are shown below as it was not possible to show all values due to the large number of entries that will be made.

Request Identification	Request TTL	Response Identification	Response TTL
12867	1	23140	64
12874	2	0	254
12880	3	0	249
12898	4	59409	251

We observe that for a specific hop, the identification number keeps incrementing by 1 for every new request and response whereas the TTL remains the same for request. The TTL remains same as we need to get response from the same hop every time.

For response from a single hop, the identification keeps increasing by 1 whereas the TTL remains the same for all packets. The identification keeps changing so that the other side knows that it is a new packet and not the same packet transmitted again due to congestion or loss of packet.

9. Calculate the average RTT for each request sent by traceroute w.r.t its respective response (from the related hop) using the different IP fields and Wireshark display filters. There shall be a proof of screenshot(s) showing this calculation for at least 1 packet with respective response(s). Plot a graph of hop name/IP address which sent the response versus the RTT using the calculations done.

We use filter *ip.addr == 192.168.0.109 && ip.addr == 163.172.208.7 && (udp || icmp)* to filter out the packets that are sent to server and the ICMP response received from different hosts.

Hop	Send time for each packet	Response time for each packet	RTT for each packet	Average RTT
192.168.0.1	2.508091115, 2.508183923, 2.508262170, 2.5083335949, 2.508411003	2.513231108, 2.513231760, 2.514558428, 2.514558921, 2.514559058	0.005139993, 0.005047837, 0.006296258, 0.0062253261, 0.006148055	0.00577149382
100.80.0.1	2.508484924, 2.508557272, 2.508629571, 2.508725571, 2.508826666	2.534782661, 2.534993874, 2.535290761, 2.535355553, 2.539086907	0.026297737, 0.026436602, 0.02666119, 0.026629982, 0.030260241	0.0272571504
172.31.120.198	2.508965721, 2.509074783, 2.509157851, 2.509230539, 2.509302494	2.540129623, 2.540467819, 2.541053498, 2.541705433, 2.542128576	0.031163902, 0.031393036, 0.031895647, 0.032474894, 0.032826082	0.0319507122
172.31.210.98	2.509380293, 2.566824860, 2.566897609, 3.023102009, 3.023180429	2.543129071, 2.589636522, 2.590400886, 3.044864773, 3.045387225	0.033748778, 0.022811662, 0.023503277, 0.021762764, 0.022206796	0.0248066554
172.31.110.120	3.023252596, 3.023321026, 3.023393692, 3.023464267, 3.023534957	3.050315340, 3.050909835, 3.051120016, 3.051391002, 3.051850891	0.027062744, 0.027588809, 0.027726324, 0.027926735, 0.028315934	0.0277241092

59.144.55.33	3.023604144, 3.023677099, 3.023745330, 3.023814871, 3.023886436	3.054263926, 3.055429171, 3.055521678, 3.057491170, 3.058656346	0.030659782, 0.031752072, 0.031776348, 0.033676299, 0.03476991	0.0325268822
116.119.49.38	3.023958083, 3.024028478, 3.024205662, 3.024277839, 3.045294679	3.184613414, 3.184971264, 3.185246120, 3.185463351, 3.185583429	0.160655331, 0.160942786, 0.161040458, 0.161185512, 0.14028875	0.1568225674
163.172.208.7	3.278266581, 3.440277891, 3.598109224, 3.789742012, 3.994534308	3.439953895, 3.597734655, 3.789255632, 3.994101766, 4.198802957	0.161687314, 0.157456764, 0.191146408, 0.204359754, 0.204268649	0.1837837778

task2.pcapng.gz

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

ip.src == 192.168.0.109

No.	Time	Source	Destination	Protocol	Length	Info
47	2.508091115	192.168.0.109	163.172.208.7	UDP	556	48870 → 33434 Len=3472
50	2.508183923	192.168.0.109	163.172.208.7	UDP	556	52235 → 33435 Len=3472
53	2.508262170	192.168.0.109	163.172.208.7	UDP	556	49824 → 33436 Len=3472
56	2.508335949	192.168.0.109	163.172.208.7	UDP	556	59428 → 33437 Len=3472
59	2.508411003	192.168.0.109	163.172.208.7	UDP	556	34325 → 33438 Len=3472
62	2.508484924	192.168.0.109	163.172.208.7	UDP	556	33895 → 33439 Len=3472
65	2.508557272	192.168.0.109	163.172.208.7	UDP	556	55667 → 33440 Len=3472
68	2.508629571	192.168.0.109	163.172.208.7	UDP	556	42857 → 33441 Len=3472
71	2.508725571	192.168.0.109	163.172.208.7	UDP	556	42631 → 33442 Len=3472
74	2.508826666	192.168.0.109	163.172.208.7	UDP	556	39926 → 33443 Len=3472
77	2.508965721	192.168.0.109	163.172.208.7	UDP	556	36122 → 33444 Len=3472
80	2.509074783	192.168.0.109	163.172.208.7	UDP	556	41629 → 33445 Len=3472
83	2.509157851	192.168.0.109	163.172.208.7	UDP	556	43422 → 33446 Len=3472
86	2.509230539	192.168.0.109	163.172.208.7	UDP	556	44657 → 33447 Len=3472
89	2.509302494	192.168.0.109	163.172.208.7	UDP	556	33481 → 33448 Len=3472
92	2.509380293	192.168.0.109	163.172.208.7	UDP	556	49717 → 33449 Len=3472
93	2.513231108	192.168.0.1	192.168.0.109	ICMP	72	Time-to-live exceeded (Time to live exceeded in tra
94	2.513231760	192.168.0.1	192.168.0.109	ICMP	72	Time-to-live exceeded (Time to live exceeded in tra
97	2.514558428	192.168.0.1	192.168.0.109	ICMP	72	Time-to-live exceeded (Time to live exceeded in tra
98	2.514558921	192.168.0.1	192.168.0.109	ICMP	72	Time-to-live exceeded (Time to live exceeded in tra
99	2.514559058	192.168.0.1	192.168.0.109	ICMP	72	Time-to-live exceeded (Time to live exceeded in tra
100	2.534782661	100.80.0.1	192.168.0.109	ICMP	72	Time-to-live exceeded (Time to live exceeded in tra
101	2.534993874	100.80.0.1	192.168.0.109	ICMP	72	Time-to-live exceeded (Time to live exceeded in tra
102	2.535290761	100.80.0.1	192.168.0.109	ICMP	72	Time-to-live exceeded (Time to live exceeded in tra
103	2.535355553	100.80.0.1	192.168.0.109	ICMP	72	Time-to-live exceeded (Time to live exceeded in tra
104	2.539086907	100.80.0.1	192.168.0.109	ICMP	72	Time-to-live exceeded (Time to live exceeded in tra
105	2.540129623	172.31.120.198	192.168.0.109	ICMP	184	Time-to-live exceeded (Time to live exceeded in tra
106	2.540467819	172.31.120.198	192.168.0.109	ICMP	184	Time-to-live exceeded (Time to live exceeded in tra
107	2.541053498	172.31.120.198	192.168.0.109	ICMP	184	Time-to-live exceeded (Time to live exceeded in tra
108	2.541705433	172.31.120.198	192.168.0.109	ICMP	184	Time-to-live exceeded (Time to live exceeded in tra
109	2.542128576	172.31.120.198	192.168.0.109	ICMP	184	Time-to-live exceeded (Time to live exceeded in tra
110	2.543129071	172.31.210.98	192.168.0.109	ICMP	184	Time-to-live exceeded (Time to live exceeded in tra
117	2.566824860	192.168.0.109	163.172.208.7	UDP	556	46252 → 33450 Len=3472
120	2.566897609	192.168.0.109	163.172.208.7	UDP	556	57072 → 33451 Len=3472

Frame 74: 556 bytes on wire (4448 bits), 556 bytes captured (4448 bits) on interface any, id 0

- Interface id: 0 (any)
 - Encapsulation type: Linux cooked-mode capture v1 (25)
 - Arrival Time: Nov 13, 2020 16:50:28.820635602 India Standard Time
 - [Time shift for this packet: 0.000000000 seconds]
 - Epoch Time: 1605266428.820635602 seconds
 - [Time delta from previous captured frame: 0.000099909 seconds]
 - [Time delta from previous displayed frame: 0.000101095 seconds]
 - [Time since reference or first frame: 2.508826666 seconds]
 - Frame Number: 74
 - Frame Length: 556 bytes (4448 bits)
 - Capture Length: 556 bytes (4448 bits)
 - [Frame is marked: False]

0000 00 04 00 01 00 06 f0 03 8c 07 e3 dd 00 00 08 00

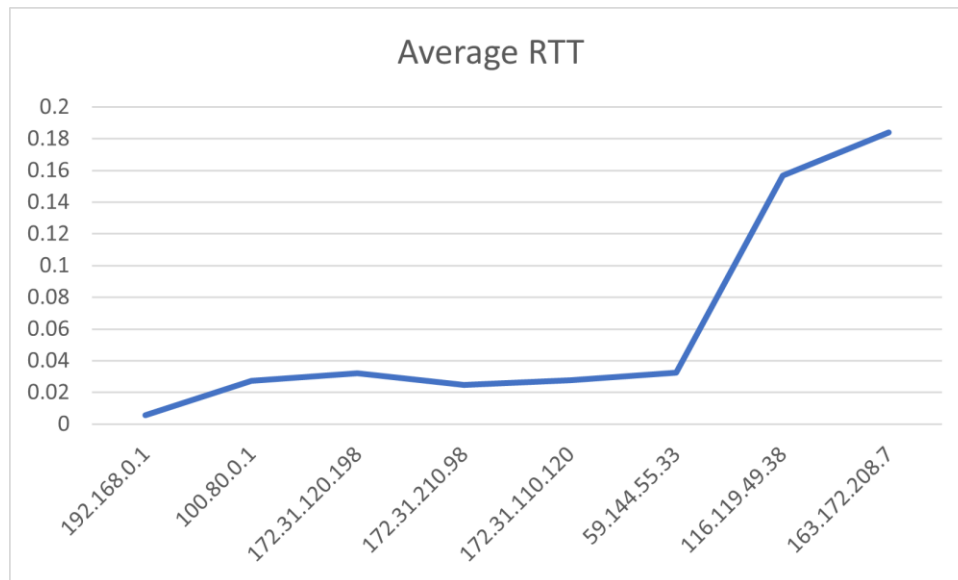
Frame (556 bytes) Reassembled IPv4 (3480 bytes)

Time relative to time reference or first frame (frame.time_relative) | Packets: 345 · Displayed: 95 (27.5%) | Profile: Default

We see that for packet 74, we get a response at packet 104 (shown by dotted line in Wireshark). We see the time difference is $2.539086907 - 2.508826666 = 0$.

Similarly we can find the RTT for each packet for each hop and calculate the average RTT.

The graph for Hop vs Average RTT is given below:



10. Pick any packet from client towards server. Has this IP datagram been fragmented?

We pick packet 117 from the capture which is from our host to server. We can check the Internet Protocol version 4 field in Wireshark where it shows the number of fragments of the packet, their lengths as well as the fragments which are combined to form a single packet.

task2.pcapng.gz

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

ip.addr == 163.172.208.7 && ip.src == 192.168.0.109

No.	Time	Source	Destination	Protocol	Length	Info
103	2.535355553	100.80.0.1	192.168.0.109	ICMP	72	Time-to-live exceeded (Time to live exceeded in tr
104	2.539086907	100.80.0.1	192.168.0.109	ICMP	72	Time-to-live exceeded (Time to live exceeded in tr
105	2.540129623	172.31.120.198	192.168.0.109	ICMP	184	Time-to-live exceeded (Time to live exceeded in tr
106	2.540467819	172.31.120.198	192.168.0.109	ICMP	184	Time-to-live exceeded (Time to live exceeded in tr
107	2.541053498	172.31.120.198	192.168.0.109	ICMP	184	Time-to-live exceeded (Time to live exceeded in tr
108	2.541705433	172.31.120.198	192.168.0.109	ICMP	184	Time-to-live exceeded (Time to live exceeded in tr
109	2.542128576	172.31.120.198	192.168.0.109	ICMP	184	Time-to-live exceeded (Time to live exceeded in tr
110	2.543129071	172.31.210.98	192.168.0.109	ICMP	184	Time-to-live exceeded (Time to live exceeded in tr
115	2.566778273	192.168.0.109	163.172.208.7	IPv4	1516	Fragmented IP protocol (proto=UDP 17, off=0, ID=32
116	2.566816990	192.168.0.109	163.172.208.7	IPv4	1516	Fragmented IP protocol (proto=UDP 17, off=1480, ID
117	2.566824860	192.168.0.109	163.172.208.7	UDP	556	46252 → 33450 Len=3472
118	2.566883856	192.168.0.109	163.172.208.7	IPv4	1516	Fragmented IP protocol (proto=UDP 17, off=0, ID=32
119	2.566891761	192.168.0.109	163.172.208.7	IPv4	1516	Fragmented IP protocol (proto=UDP 17, off=1480, ID
120	2.566897609	192.168.0.109	163.172.208.7	UDP	556	57072 → 33451 Len=3472
123	2.589636522	172.31.210.98	192.168.0.109	ICMP	184	Time-to-live exceeded (Time to live exceeded in tr
124	2.590400886	172.31.210.98	192.168.0.109	ICMP	184	Time-to-live exceeded (Time to live exceeded in tr
144	3.023056996	192.168.0.109	163.172.208.7	IPv4	1516	Fragmented IP protocol (proto=UDP 17, off=0, ID=32
145	3.023093015	192.168.0.109	163.172.208.7	IPv4	1516	Fragmented IP protocol (proto=UDP 17, off=1480, ID
146	3.023102009	192.168.0.109	163.172.208.7	UDP	556	55026 → 33452 Len=3472
147	3.023165130	192.168.0.109	163.172.208.7	IPv4	1516	Fragmented IP protocol (proto=UDP 17, off=0, ID=32
148	3.023173820	192.168.0.109	163.172.208.7	IPv4	1516	Fragmented IP protocol (proto=UDP 17, off=1480, ID
149	3.023180429	192.168.0.109	163.172.208.7	UDP	556	50523 → 33453 Len=3472
150	3.023237538	192.168.0.109	163.172.208.7	IPv4	1516	Fragmented IP protocol (proto=UDP 17, off=0, ID=32

Protocol: UDP (17)
Header Checksum: 0x4c35 [validation disabled]
[Header checksum status: Unverified]
Source Address: 192.168.0.109
Destination Address: 163.172.208.7

▼ [3 IPv4 Fragments (3480 bytes): #115(1480), #116(1480), #117(520)]

- [Frame: 115, payload: 0-1479 (1480 bytes)]
- [Frame: 116, payload: 1480-2959 (1480 bytes)]
- [Frame: 117, payload: 2960-3479 (520 bytes)]

[Fragment count: 3]

```

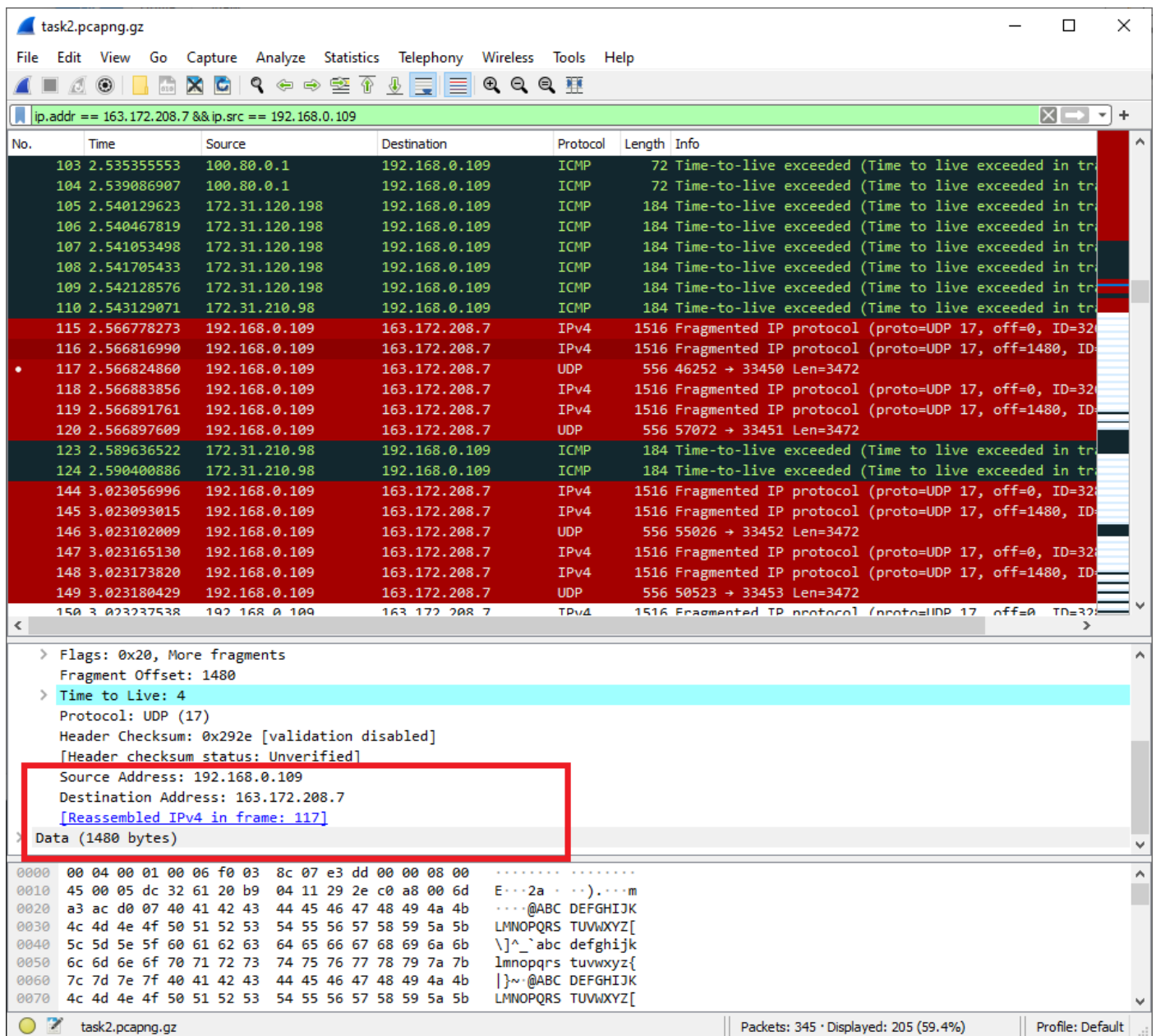
0000  00 04 00 01 00 06 f0 03 8c 07 e3 dd 00 00 08 00  ....2a.r...L5...m
0010  45 00 02 1c 32 61 01 72 04 11 4c 35 c0 a8 00 6d  ....HIJK LMNOPQRS
0020  a3 ac d0 07 48 49 4a 4b 4c 4d 4e 4f 50 51 52 53  ....TUVWXYZ[ \]^_`abc
0030  54 55 56 57 58 59 5a 5b 5c 5d 5e 5f 60 61 62 63  defghijk lmnopqrs
0040  64 65 66 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73  tuvvwxyz{ }~@ABC
0050  74 75 76 77 78 79 7a 7b 7c 7d 7e 7f 40 41 42 43

```

Frame (556 bytes) Reassembled IPv4 (3480 bytes)

task2.pcapng.gz Packets: 345 · Displayed: 205 (59.4%) Profile: Default

If the packet is not a end of fragment packet, such as packet 116, wireshark shows the frame in which the fragment is reassembled to form a packet.



11. Explain how you determine whether or not the datagram has been fragmented and where does the fragmentation end.

As explained in previous answer, we can check Internet Protocol version 4 filed in Wireshark to see if the datagram is fragmented. For a datagram that is not ending datagram, we can see the datagram number where the datagrams are reassembled to form a single packet.

To check where which frame is the last one in fragments, we can check MF flag. When More Fragments flag is 0 we know that there are no more fragments left.

12. In the saved file used for analysis, export only those IP packets involved in either direction (in communication) using appropriate Wireshark display filters and save it to another file to upload during submission.

File is uploaded as task2_filtered.pcapng.gz

Task 3

1. Comment on your understanding of differences between traceroute and ping command executed for **ping-ams1.online.net** (in **Task1 and Task2**) using the respective wireshark traces. List the IP fields which indicate the differences in the behaviour of execution. Place at least 1 screenshot of wireshark from respective traces showing relevant display filters as a proof of explaining the difference.

The following observations were made from the ping and traceroute.

Ping command works over the IP layer. This command work on ECHO-Request and ECHO-Reply. This command is used to find whether a host is available or not. For this client send the ECHO request using ICMP packet and the destination host will reply with an ECHO packet if it is available.

Traceroute is used to find the path and number of hops in between source and destination. This protocol uses ICMP, IPV4 and UDP protocol for this mechanism. This process uses various messages like TTL exceeded, port unreachable etc to find the hosts that act as hop between the client and server.

The IP fields which indicate the difference are:

For traceroute, we use TTL filed to set the number of hops. By setting the number of hops to 1 and increasing it by 1 for every next hop, we can find all the hops from which the packet passes. This is achieved as when the TTL gets to 0, the hop which drops the packet will send a ICMP response saying TTL exceeded back to the host. We can check the IP address from where the response came and find the hops.

task2.pcapng.gz

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

ip.src == 192.168.0.109

No.	Time	Source	Destination	Protocol	Length	Info
47	2.508091115	192.168.0.109	163.172.208.7	UDP	556	48870 → 33434 Len=3472
50	2.508183923	192.168.0.109	163.172.208.7	UDP	556	52235 → 33435 Len=3472
53	2.508262170	192.168.0.109	163.172.208.7	UDP	556	49824 → 33436 Len=3472
56	2.508335949	192.168.0.109	163.172.208.7	UDP	556	59428 → 33437 Len=3472
59	2.508411003	192.168.0.109	163.172.208.7	UDP	556	34325 → 33438 Len=3472
62	2.508484924	192.168.0.109	163.172.208.7	UDP	556	33895 → 33439 Len=3472
65	2.508557272	192.168.0.109	163.172.208.7	UDP	556	55667 → 33440 Len=3472
68	2.508629571	192.168.0.109	163.172.208.7	UDP	556	42857 → 33441 Len=3472
71	2.508725571	192.168.0.109	163.172.208.7	UDP	556	42631 → 33442 Len=3472
74	2.508826666	192.168.0.109	163.172.208.7	UDP	556	39926 → 33443 Len=3472
77	2.508965721	192.168.0.109	163.172.208.7	UDP	556	36122 → 33444 Len=3472
80	2.509074783	192.168.0.109	163.172.208.7	UDP	556	41629 → 33445 Len=3472
83	2.509157851	192.168.0.109	163.172.208.7	UDP	556	43422 → 33446 Len=3472
86	2.509230539	192.168.0.109	163.172.208.7	UDP	556	44657 → 33447 Len=3472
89	2.509302494	192.168.0.109	163.172.208.7	UDP	556	33481 → 33448 Len=3472
92	2.509380293	192.168.0.109	163.172.208.7	UDP	556	49717 → 33449 Len=3472
93	2.513231108	192.168.0.1	192.168.0.109	ICMP	72	Time-to-live exceeded (Time to live exceeded in tra
94	2.513231760	192.168.0.1	192.168.0.109	ICMP	72	Time-to-live exceeded (Time to live exceeded in tra
97	2.514558428	192.168.0.1	192.168.0.109	ICMP	72	Time-to-live exceeded (Time to live exceeded in tra
98	2.514558921	192.168.0.1	192.168.0.109	ICMP	72	Time-to-live exceeded (Time to live exceeded in tra
99	2.514559058	192.168.0.1	192.168.0.109	ICMP	72	Time-to-live exceeded (Time to live exceeded in tra
100	2.534782661	100.80.0.1	192.168.0.109	ICMP	72	Time-to-live exceeded (Time to live exceeded in tra
101	2.534993874	100.80.0.1	192.168.0.109	ICMP	72	Time-to-live exceeded (Time to live exceeded in tra
102	2.535290761	100.80.0.1	192.168.0.109	ICMP	72	Time-to-live exceeded (Time to live exceeded in tra
103	2.535355553	100.80.0.1	192.168.0.109	ICMP	72	Time-to-live exceeded (Time to live exceeded in tra
104	2.539086907	100.80.0.1	192.168.0.109	ICMP	72	Time-to-live exceeded (Time to live exceeded in tra
105	2.540129623	172.31.120.198	192.168.0.109	ICMP	184	Time-to-live exceeded (Time to live exceeded in tra
106	2.540467819	172.31.120.198	192.168.0.109	ICMP	184	Time-to-live exceeded (Time to live exceeded in tra
107	2.541053498	172.31.120.198	192.168.0.109	ICMP	184	Time-to-live exceeded (Time to live exceeded in tra
108	2.541705433	172.31.120.198	192.168.0.109	ICMP	184	Time-to-live exceeded (Time to live exceeded in tra
109	2.542128576	172.31.120.198	192.168.0.109	ICMP	184	Time-to-live exceeded (Time to live exceeded in tra
110	2.543129071	172.31.210.98	192.168.0.109	ICMP	184	Time-to-live exceeded (Time to live exceeded in tra
117	2.566824860	192.168.0.109	163.172.208.7	UDP	556	46252 → 33450 Len=3472
120	2.566897609	192.168.0.109	163.172.208.7	UDP	556	57072 → 33451 Len=3472

> Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
 Total Length: 540
 Identification: 0x3243 (12867)
 Flags: 0x01
 0... = Reserved bit: Not set
 .0.. = Don't fragment: Not set
 ..0. = More fragments: Not set
 Fragment Offset: 2960
 Time to Live: 1
 [Expert Info (Note/Sequence): "Time To Live" only 1]
 Protocol: UDP (17)
 Header Checksum: 0x4f53 [validation disabled]
 [Header checksum status: Unverified]

0000 be e6 82 9a 0d 98 69 ad 40 41 42 43 44 45 46 47i. @ABCDEFG

Frame (556 bytes) Reassembled IPv4 (3480 bytes)

The reassembled payload (ip.reassembled.data), 3,480 bytes

Packets: 345 · Displayed: 95 (27.5%) Profile: Default

For Ping requests, we set the Protocol in IPv4 to 1 and Type in ICMP to 8 for ping request or 0 for ping reply.

task1_filtered.pcapng.gz

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
12	1.188666032	163.172.208.7	192.168.78.227	ICMP	584	Echo (ping) reply id=0x0001, seq=2/5
13	2.002558098	192.168.78.227	163.172.208.7	IPv4	1516	Fragmented IP protocol (proto=ICMP 1, o
14	2.002561775	192.168.78.227	163.172.208.7	IPv4	1516	Fragmented IP protocol (proto=ICMP 1, o
15	2.002562523	192.168.78.227	163.172.208.7	ICMP	584	Echo (ping) request id=0x0001, seq=3/7
16	2.189817689	163.172.208.7	192.168.78.227	IPv4	1516	Fragmented IP protocol (proto=ICMP 1, o
17	2.189818398	163.172.208.7	192.168.78.227	IPv4	1516	Fragmented IP protocol (proto=ICMP 1, o
18	2.189818475	163.172.208.7	192.168.78.227	ICMP	584	Echo (ping) reply id=0x0001, seq=3/7
19	3.003754493	192.168.78.227	163.172.208.7	IPv4	1516	Fragmented IP protocol (proto=ICMP 1, o
20	3.003757554	192.168.78.227	163.172.208.7	IPv4	1516	Fragmented IP protocol (proto=ICMP 1, o
21	3.003758047	192.168.78.227	163.172.208.7	ICMP	584	Echo (ping) request id=0x0001, seq=4/10
22	3.183084055	163.172.208.7	192.168.78.227	IPv4	1516	Fragmented IP protocol (proto=ICMP 1, o
23	3.183084388	163.172.208.7	192.168.78.227	IPv4	1516	Fragmented IP protocol (proto=ICMP 1, o
24	3.183084426	163.172.208.7	192.168.78.227	ICMP	584	Echo (ping) reply id=0x0001, seq=4/10
25	4.004793597	192.168.78.227	163.172.208.7	IPv4	1516	Fragmented IP protocol (proto=ICMP 1, o
26	4.004797144	192.168.78.227	163.172.208.7	IPv4	1516	Fragmented IP protocol (proto=ICMP 1, o
27	4.004797658	192.168.78.227	163.172.208.7	ICMP	584	Echo (ping) request id=0x0001, seq=5/10
28	4.184783308	163.172.208.7	192.168.78.227	IPv4	1516	Fragmented IP protocol (proto=ICMP 1, o
29	4.184783726	163.172.208.7	192.168.78.227	IPv4	1516	Fragmented IP protocol (proto=ICMP 1, o
30	4.184783763	163.172.208.7	192.168.78.227	ICMP	584	Echo (ping) reply id=0x0001, seq=5/10

Protocol: ICMP (1)
 Header checksum: 0x3e10 [validation disabled]
 [Header checksum status: Unverified]
 Source Address: 192.168.78.227
 Destination Address: 163.172.208.7
 > [3 IPv4 Fragments (3508 bytes): #13(1480), #14(1480), #15(548)]

Internet Control Message Protocol
 Type: 8 (Echo (ping) request)
 Code: 0
 Checksum: 0x7bbd [correct]

0070 d4 d5 d6 d7 d8 d9 da db dc dd de df e0 e1 e2 e3
 0080 e4 e5 e6 e7 e8 e9 ea eb ec ed ee ef f0 f1 f2 f3
 0090 f4 f5 f6 f7 f8 f9 fa fb fc fd fe ff 00 01 02 03
 00a0 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13
 00b0 14 15 16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 !"#
 00c0 24 25 26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 32 33 \$%&'()*+,-./0123

Frame (584 bytes) Reassembled IPv4 (3508 bytes)

Protocol (ip.proto), 1 byte

Packets: 30 · Displayed: 30 (100.0%) Profile: Default

I certify that this assignment/report is my own work, based on my personal study and/or research and that I have acknowledged all material and sources used in its preparation, whether they be books, articles, reports, lecture notes, and any other kind of document, electronic or personal communication. I also certify that this assignment/report has not previously been submitted for assessment in any other course, except where specific permission has been granted from all course instructors involved, or at any other time in this course, and that I have not copied in part or whole or otherwise plagiarised the work of other students and/or persons. I pledge to uphold the principles of honesty and responsibility at CSE@IITH. In addition, I understand my responsibility to report honour violations by other students if I become aware of it.