

## ASSIGNMENT 5

### Wireshark for Network Layer

Submitted By,  
**SHRUSTI**  
**CS22MTECH11017**

**Task1: Ping to destination (using [www.zoho.com](http://www.zoho.com) and size as 5329)**

Start the Wireshark packet sniffer and start capturing. Open a terminal. Execute  
`ping -s <size><server-name> -c 5`

Stop the wireshark capture and save the file for further analysis.

Answer the following using the captured trace file.

**1. What does the above ping command do ?**

**Solution :**

PING command is used to check the network connectivity between host and server/host. This command takes as input the IP address or the URL and sends a data packet to the specified address and gets a response from the server/host, this time is recorded which is called latency.

Ping uses ICMP(Internet Control Message Protocol) to send an ICMP echo message to the specified host if that host is available then it sends an ICMP reply message.

**In Windows, instead of -s and -c options we use options -l and -n respectively.**

Options	Explanation
-l <i>size</i>	Use this option to set the size in bytes of the echo request packet from 32 to 65,527. Here, we have used the size as 5329. The ping command will send a 32-byte echo request if you don't use the -l option.
-n <i>count</i>	This option sets the number of ICMP Echo Requests to send, from 1 to 4294967295. Here, we have used count as 5

The ping command will send 4 by default if -n isn't used.

```
C:\WINDOWS\system32>ping -l 5329 www.zoho.com -n 5

Pinging gsite.zohocdn.com [204.141.42.155] with 5329 bytes of data:
Reply from 204.141.42.155: bytes=5329 time=252ms TTL=46
Reply from 204.141.42.155: bytes=5329 time=251ms TTL=46
Reply from 204.141.42.155: bytes=5329 time=251ms TTL=46
Reply from 204.141.42.155: bytes=5329 time=255ms TTL=46
Reply from 204.141.42.155: bytes=5329 time=251ms TTL=46

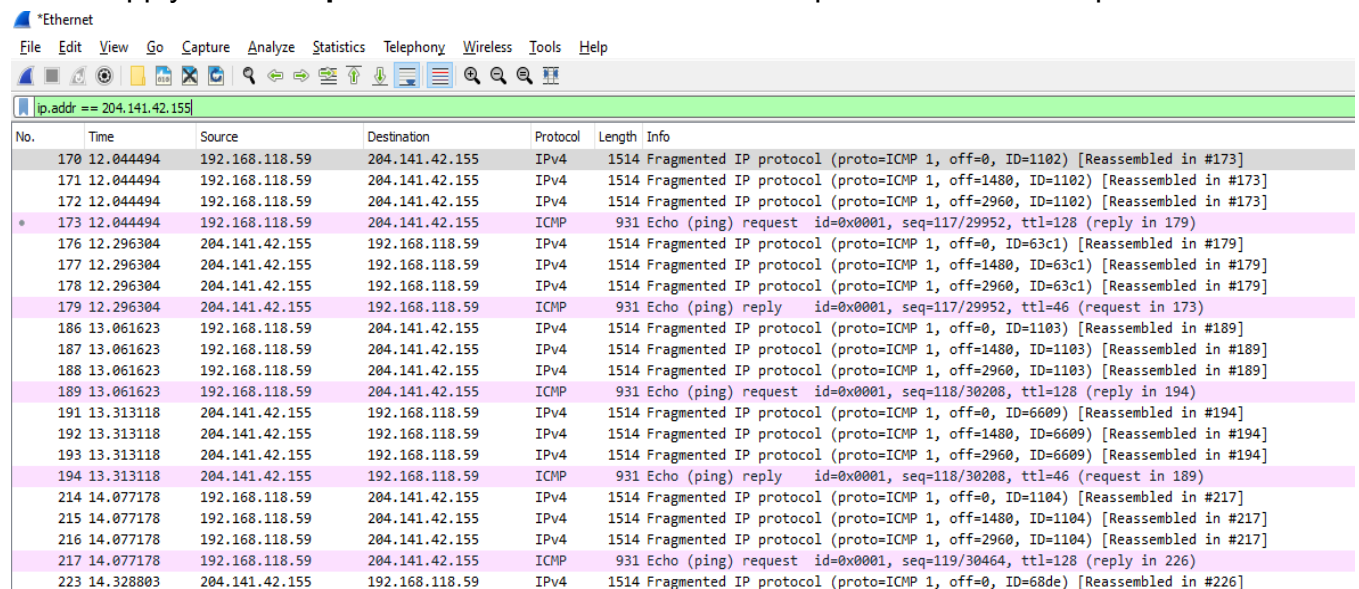
Ping statistics for 204.141.42.155:
    Packets: Sent = 5, Received = 5, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 251ms, Maximum = 255ms, Average = 252ms
```

## 2. How many total IP packets are exchanged in the communication between your host and the remote server ?

### Solution:

From the ping command we can see that the IP address of [www.zoho.com](http://www.zoho.com) is 204.141.42.155

Hence apply the filter **ip.addr == 204.141.42.155** to filter packets from our capture.



No.	Time	Source	Destination	Protocol	Length	Info
170	12.044494	192.168.118.59	204.141.42.155	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=0, ID=1102) [Reassembled in #173]
171	12.044494	192.168.118.59	204.141.42.155	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=1480, ID=1102) [Reassembled in #173]
172	12.044494	192.168.118.59	204.141.42.155	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=2960, ID=1102) [Reassembled in #173]
173	12.044494	192.168.118.59	204.141.42.155	ICMP	931	Echo (ping) request id=0x0001, seq=117/29952, ttl=128 (reply in 179)
176	12.296304	204.141.42.155	192.168.118.59	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=0, ID=63c1) [Reassembled in #179]
177	12.296304	204.141.42.155	192.168.118.59	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=1480, ID=63c1) [Reassembled in #179]
178	12.296304	204.141.42.155	192.168.118.59	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=2960, ID=63c1) [Reassembled in #179]
179	12.296304	204.141.42.155	192.168.118.59	ICMP	931	Echo (ping) reply id=0x0001, seq=117/29952, ttl=46 (request in 173)
186	13.061623	192.168.118.59	204.141.42.155	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=0, ID=1103) [Reassembled in #189]
187	13.061623	192.168.118.59	204.141.42.155	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=1480, ID=1103) [Reassembled in #189]
188	13.061623	192.168.118.59	204.141.42.155	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=2960, ID=1103) [Reassembled in #189]
189	13.061623	192.168.118.59	204.141.42.155	ICMP	931	Echo (ping) request id=0x0001, seq=118/30208, ttl=128 (reply in 194)
191	13.313118	204.141.42.155	192.168.118.59	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=0, ID=6609) [Reassembled in #194]
192	13.313118	204.141.42.155	192.168.118.59	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=1480, ID=6609) [Reassembled in #194]
193	13.313118	204.141.42.155	192.168.118.59	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=2960, ID=6609) [Reassembled in #194]
194	13.313118	204.141.42.155	192.168.118.59	ICMP	931	Echo (ping) reply id=0x0001, seq=118/30208, ttl=46 (request in 189)
214	14.077178	192.168.118.59	204.141.42.155	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=0, ID=1104) [Reassembled in #217]
215	14.077178	192.168.118.59	204.141.42.155	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=1480, ID=1104) [Reassembled in #217]
216	14.077178	192.168.118.59	204.141.42.155	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=2960, ID=1104) [Reassembled in #217]
217	14.077178	192.168.118.59	204.141.42.155	ICMP	931	Echo (ping) request id=0x0001, seq=119/30464, ttl=128 (reply in 226)
223	14.328803	204.141.42.155	192.168.118.59	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=0, ID=68de) [Reassembled in #226]

Number of IP packets exchanged in the communication between your host and the remote server are : **40 packets**.

This can be checked in Statistics → Conversations

Wireshark · Conversations · Ethernet										
Ethernet · 1		IPv4 · 1		IPv6	TCP	UDP				
Address A	Address B	Packets	Bytes	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration	Bits/s A → B
192.168.118.59	204.141.42.155	40	54 k	20	27 k	20	27 k	12.044494	4.3301	50 k

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

**Solution :**

The size of each ping request sent from your host to the remote server is : **5329 bytes** this is the same as the packet size defined by us in the ping command.

ip.addr == 204.141.42.155						
No.	Time	Source	Destination	Protocol	Length	Info
170	12.044494	192.168.118.59	204.141.42.155	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=0, ID=1102) [Reassembled in #173]
171	12.044494	192.168.118.59	204.141.42.155	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=1480, ID=1102) [Reassembled in #173]
172	12.044494	192.168.118.59	204.141.42.155	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=2960, ID=1102) [Reassembled in #173]
173	12.044494	192.168.118.59	204.141.42.155	ICMP	931	Echo (ping) request id=0x0001, seq=117/29952, ttl=128 (reply in 179)
176	12.296304	204.141.42.155	192.168.118.59	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=0, ID=63c1) [Reassembled in #179]
177	12.296304	204.141.42.155	192.168.118.59	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=1480, ID=63c1) [Reassembled in #179]
178	12.296304	204.141.42.155	192.168.118.59	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=2960, ID=63c1) [Reassembled in #179]
179	12.296304	204.141.42.155	192.168.118.59	ICMP	931	Echo (ping) reply id=0x0001, seq=117/29952, ttl=46 (request in 173)
186	13.061623	192.168.118.59	204.141.42.155	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=0, ID=1103) [Reassembled in #189]
187	13.061623	192.168.118.59	204.141.42.155	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=1480, ID=1103) [Reassembled in #189]
188	13.061623	192.168.118.59	204.141.42.155	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=2960, ID=1103) [Reassembled in #189]
189	13.061623	192.168.118.59	204.141.42.155	ICMP	931	Echo (ping) request id=0x0001, seq=118/30208, ttl=128 (reply in 194)
191	13.313118	204.141.42.155	192.168.118.59	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=0, ID=6609) [Reassembled in #194]
192	13.313118	204.141.42.155	192.168.118.59	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=1480, ID=6609) [Reassembled in #194]
193	13.313118	204.141.42.155	192.168.118.59	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=2960, ID=6609) [Reassembled in #194]
194	13.313118	204.141.42.155	192.168.118.59	ICMP	931	Echo (ping) reply id=0x0001, seq=118/30208, ttl=46 (request in 189)
Protocol: ICMP (1) Header Checksum: 0x0000 [validation disabled] [Header checksum status: Unverified] Source Address: 192.168.118.59 Destination Address: 204.141.42.155 > [4 IPv4 Fragments (5337 bytes): #170(1480), #171(1480), #172(1480), #173(897)] ✓ Internet Control Message Protocol Type: 8 (Echo (ping) request) Code: 0 Checksum: 0x5b61 [correct] [Checksum Status: Good] Identifier (BE): 1 (0x0001) Identifier (LE): 256 (0x0100) Sequence Number (BE): 117 (0x0075) Sequence Number (LE): 29952 (0x7500) [Response frame: 179] > Data (5329 bytes)						

4. 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.

**Solution:**

ip.addr == 204.141.42.155 && icmp						
No.	Time	Source	Destination	Protocol	Length	Info
173	12.044494	192.168.118.59	204.141.42.155	ICMP	931	Echo (ping) request id=0x0001, seq=117/29952, ttl=128 (reply in 179)
179	12.296304	204.141.42.155	192.168.118.59	ICMP	931	Echo (ping) reply id=0x0001, seq=117/29952, ttl=46 (request in 173)
189	13.061623	192.168.118.59	204.141.42.155	ICMP	931	Echo (ping) request id=0x0001, seq=118/30208, ttl=128 (reply in 194)
194	13.313118	204.141.42.155	192.168.118.59	ICMP	931	Echo (ping) reply id=0x0001, seq=118/30208, ttl=46 (request in 189)
217	14.077178	192.168.118.59	204.141.42.155	ICMP	931	Echo (ping) request id=0x0001, seq=119/30464, ttl=128 (reply in 226)
226	14.328803	204.141.42.155	192.168.118.59	ICMP	931	Echo (ping) reply id=0x0001, seq=119/30464, ttl=46 (request in 217)
236	15.106903	192.168.118.59	204.141.42.155	ICMP	931	Echo (ping) request id=0x0001, seq=120/30720, ttl=128 (reply in 242)
242	15.362242	204.141.42.155	192.168.118.59	ICMP	931	Echo (ping) reply id=0x0001, seq=120/30720, ttl=46 (request in 236)
289	16.123129	192.168.118.59	204.141.42.155	ICMP	931	Echo (ping) request id=0x0001, seq=121/30976, ttl=128 (reply in 306)
306	16.374593	204.141.42.155	192.168.118.59	ICMP	931	Echo (ping) reply id=0x0001, seq=121/30976, ttl=46 (request in 289)

An ICMP packet is a request or response packet can be identified by the column “Info”.

Details of the packet can be seen by opening the packet.

Wireshark · Packet 173 · task1.pcapng

- > Frame 173: 931 bytes on wire (7448 bits), 931 bytes captured (7448 bits) on interface \Device\NPF\_{FF8771DA-851B-4CFA-815D-B733132483A2}, id 0
- > Ethernet II, Src: Dell\_af:f7:5a (20:47:47:af:f7:5a), Dst: HewlettP\_a9:8b:f9 (5c:8a:38:a9:8b:f9)
- ▼ Internet Protocol Version 4, Src: 192.168.118.59, Dst: 204.141.42.155
  - 0100 .... = Version: 4
  - .... 0101 = Header Length: 20 bytes (5)
  - > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
  - Total Length: 917
  - Identification: 0x1102 (4354)
  - > Flags: 0x02
  - ...1 0001 0101 1000 = Fragment Offset: 4440
  - Time to Live: 128
  - Protocol: ICMP (1)
  - Header Checksum: 0x0000 [validation disabled]
  - [Header checksum status: Unverified]
  - Source Address: 192.168.118.59
  - Destination Address: 204.141.42.155
  - > [4 IPv4 Fragments (5337 bytes): #170(1480), #171(1480), #172(1480), #173(897)]
- ▼ Internet Control Message Protocol
  - Type: 8 (Echo (ping) request)
  - Code: 0
  - Checksum: 0x5b61 [correct]
  - [Checksum Status: Good]
  - Identifier (BE): 1 (0x0001)
  - Identifier (LE): 256 (0x0100)
  - Sequence Number (BE): 117 (0x0075)
  - Sequence Number (LE): 29952 (0x7500)
  - [\[Response frame: 179\]](#)
  - > Data (5329 bytes)

Details of fragments

Packet no.	Fragmented (Yes/No)	#fragments	Details of each individual fragment			Total actual length of data in packet
173 (ICMP request)	Yes	4 (packet # 170, 171, 172,173)				1480 + 1480 + 1480+ 897 bytes  = 5337 bytes
			Fragment no.	Time (in ms)	Fragment length (bytes)	
			170	12.044494	1480	
			171	12.044494	1480	
			172	12.044494	1480	
			173	12.044494	897	
179 (ICMP reply)	Yes	4 (packet #176, 177,178, 179,)				1480 + 1480 + 1480+ 897 bytes  = 5337 bytes
			Fragment no.	Time (in ms)	Fragment length (bytes)	
			176	12.296304	1480	
			177	12.296304	1480	
			178	12.296304	1480	
			179	12.296304	897	
189 (ICMP request)	Yes	4 (packet # 186, 187,188, 189)				1480 + 1480 + 1480+ 897 bytes  = 5337 bytes
			Fragment no.	Time (in ms)	Fragment length (bytes)	
			186	13.061623	1480	
			187	13.061623	1480	
			188	13.061623	1480	
			189	13.061623	897	
194 (ICMP reply)	Yes	4 (packet # 191,192,				1480 + 1480 + 1480+ 897 bytes
			Fragment	Time	Fragment	

		193, 194)	<table><tr><th>no.</th><th>(in ms)</th><th>length (bytes)</th></tr><tr><td>191</td><td>13.313118</td><td>1480</td></tr><tr><td>192</td><td>13.313118</td><td>1480</td></tr><tr><td>193</td><td>13.313118</td><td>1480</td></tr><tr><td>194</td><td>13.313118</td><td>897</td></tr></table>	no.	(in ms)	length (bytes)	191	13.313118	1480	192	13.313118	1480	193	13.313118	1480	194	13.313118	897	= 5337 bytes
no.	(in ms)	length (bytes)																	
191	13.313118	1480																	
192	13.313118	1480																	
193	13.313118	1480																	
194	13.313118	897																	
217 (ICMP request)	Yes	4 (packet # 214, 215, 216, 217)	<table><tr><th>Fragment no.</th><th>Time (in ms)</th><th>Fragment length (bytes)</th></tr><tr><td>214</td><td>14.077178</td><td>1480</td></tr><tr><td>215</td><td>14.077178</td><td>1480</td></tr><tr><td>216</td><td>14.077178</td><td>1480</td></tr><tr><td>217</td><td>14.077178</td><td>897</td></tr></table>	Fragment no.	Time (in ms)	Fragment length (bytes)	214	14.077178	1480	215	14.077178	1480	216	14.077178	1480	217	14.077178	897	1480 + 1480 + 1480+ 897 bytes  = 5337 bytes
Fragment no.	Time (in ms)	Fragment length (bytes)																	
214	14.077178	1480																	
215	14.077178	1480																	
216	14.077178	1480																	
217	14.077178	897																	
226 (ICMP reply)	Yes	4 (packet # 223, 224, 225, 226)	<table><tr><th>Fragment no.</th><th>Time (in ms)</th><th>Fragment length (bytes)</th></tr><tr><td>223</td><td>14.328803</td><td>1480</td></tr><tr><td>224</td><td>14.328803</td><td>1480</td></tr><tr><td>225</td><td>14.328803</td><td>1480</td></tr><tr><td>226</td><td>14.328803</td><td>897</td></tr></table>	Fragment no.	Time (in ms)	Fragment length (bytes)	223	14.328803	1480	224	14.328803	1480	225	14.328803	1480	226	14.328803	897	1480 + 1480 + 1480+ 897 bytes  = 5337 bytes
Fragment no.	Time (in ms)	Fragment length (bytes)																	
223	14.328803	1480																	
224	14.328803	1480																	
225	14.328803	1480																	
226	14.328803	897																	
236 (ICMP request)	Yes	4 (packet # 233, 234, 235, 236)	<table><tr><th>Fragment no.</th><th>Time (in ms)</th><th>Fragment length (bytes)</th></tr><tr><td>233</td><td>15.106903</td><td>1480</td></tr><tr><td>234</td><td>15.106903</td><td>1480</td></tr></table>	Fragment no.	Time (in ms)	Fragment length (bytes)	233	15.106903	1480	234	15.106903	1480	1480 + 1480 + 1480+ 897 bytes  = 5337 bytes						
Fragment no.	Time (in ms)	Fragment length (bytes)																	
233	15.106903	1480																	
234	15.106903	1480																	

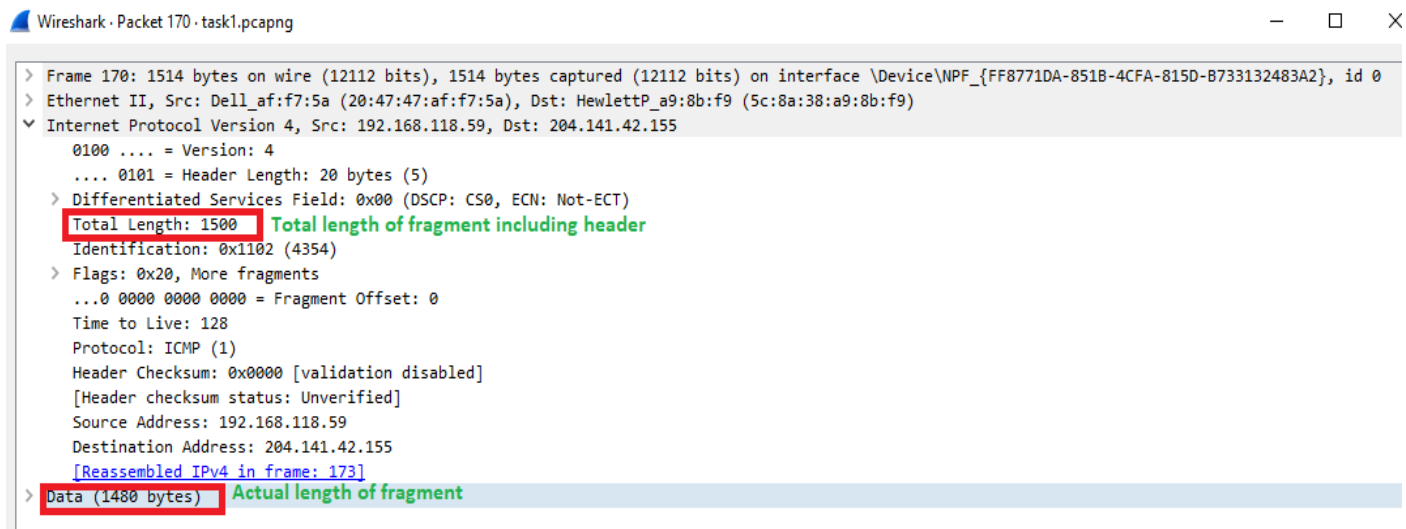
			<table><tr><td>235</td><td>15.106903</td><td>1480</td></tr><tr><td>236</td><td>15.106903</td><td>897</td></tr></table>	235	15.106903	1480	236	15.106903	897										
235	15.106903	1480																	
236	15.106903	897																	
242 (ICMP reply)	Yes	4 (packet # 239, 240, 241, 242)	<table><tr><th>Fragment no.</th><th>Time (in ms)</th><th>Fragment length (bytes)</th></tr><tr><td>239</td><td>15.362242</td><td>1480</td></tr><tr><td>240</td><td>15.362242</td><td>1480</td></tr><tr><td>241</td><td>15.362242</td><td>1480</td></tr><tr><td>242</td><td>15.362242</td><td>897</td></tr></table>	Fragment no.	Time (in ms)	Fragment length (bytes)	239	15.362242	1480	240	15.362242	1480	241	15.362242	1480	242	15.362242	897	1480 + 1480 + 1480+ 897 bytes  = 5337 bytes
Fragment no.	Time (in ms)	Fragment length (bytes)																	
239	15.362242	1480																	
240	15.362242	1480																	
241	15.362242	1480																	
242	15.362242	897																	
289 (ICMP request)	Yes	4 (packet # 286, 287, 288, 289)	<table><tr><th>Fragment no.</th><th>Time (in ms)</th><th>Fragment length (bytes)</th></tr><tr><td>286</td><td>16.123129</td><td>1480</td></tr><tr><td>287</td><td>16.123129</td><td>1480</td></tr><tr><td>288</td><td>16.123129</td><td>1480</td></tr><tr><td>289</td><td>16.123129</td><td>897</td></tr></table>	Fragment no.	Time (in ms)	Fragment length (bytes)	286	16.123129	1480	287	16.123129	1480	288	16.123129	1480	289	16.123129	897	1480 + 1480 + 1480+ 897 bytes  = 5337 bytes
Fragment no.	Time (in ms)	Fragment length (bytes)																	
286	16.123129	1480																	
287	16.123129	1480																	
288	16.123129	1480																	
289	16.123129	897																	
306 (ICMP reply)	Yes	4 (packet # 303, 304, 305, 306)	<table><tr><th>Fragment no.</th><th>Time (in ms)</th><th>Fragment length (bytes)</th></tr><tr><td>303</td><td>16.374593</td><td>1480</td></tr><tr><td>304</td><td>16.374593</td><td>1480</td></tr><tr><td>305</td><td>16.374593</td><td>1480</td></tr><tr><td>306</td><td>16.374593</td><td>897</td></tr></table>	Fragment no.	Time (in ms)	Fragment length (bytes)	303	16.374593	1480	304	16.374593	1480	305	16.374593	1480	306	16.374593	897	1480 + 1480 + 1480+ 897 bytes  = 5337 bytes
Fragment no.	Time (in ms)	Fragment length (bytes)																	
303	16.374593	1480																	
304	16.374593	1480																	
305	16.374593	1480																	
306	16.374593	897																	

**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 ?**

**Solution :**

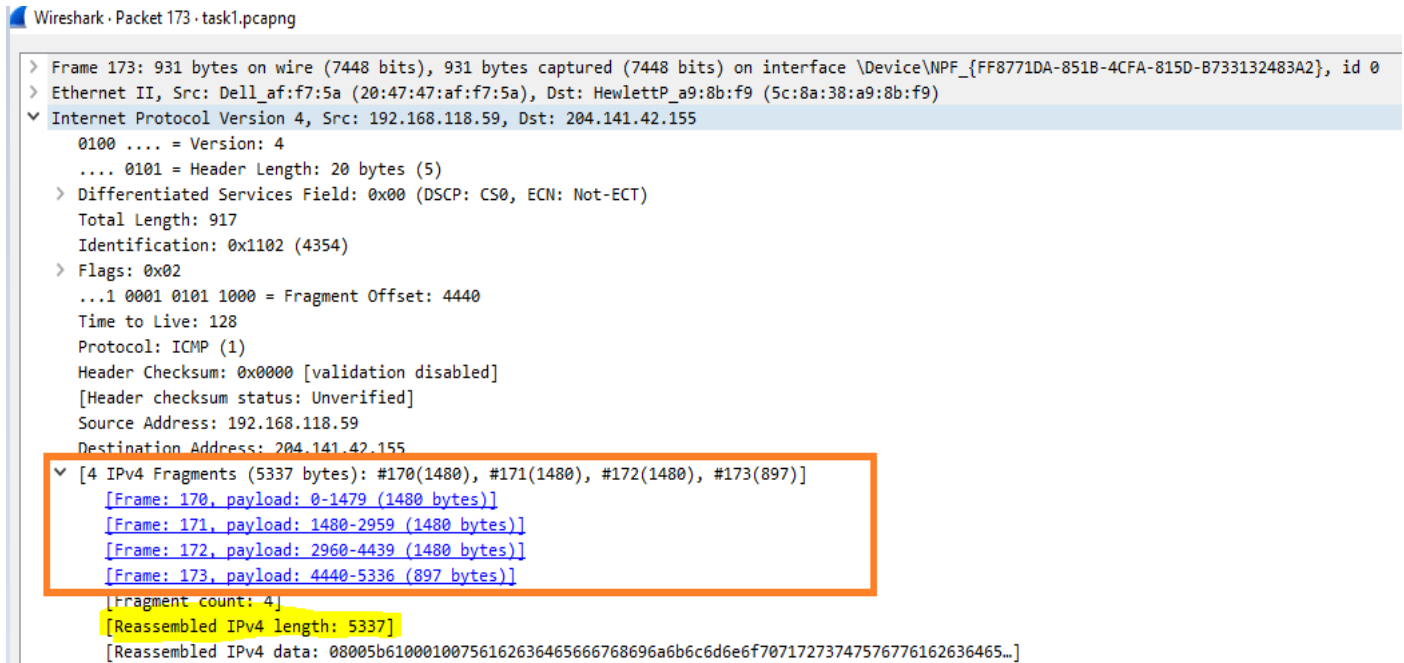
i) Total length of fragment is **1500 bytes** and actual data ( length after removing headers) is **1480 bytes** in individual fragments can be found by opening the packet and observing the IPV4 details.

For example, the length of actual data for packet 170 is shown below. Similarly we can check the length of actual data for other packets as well.



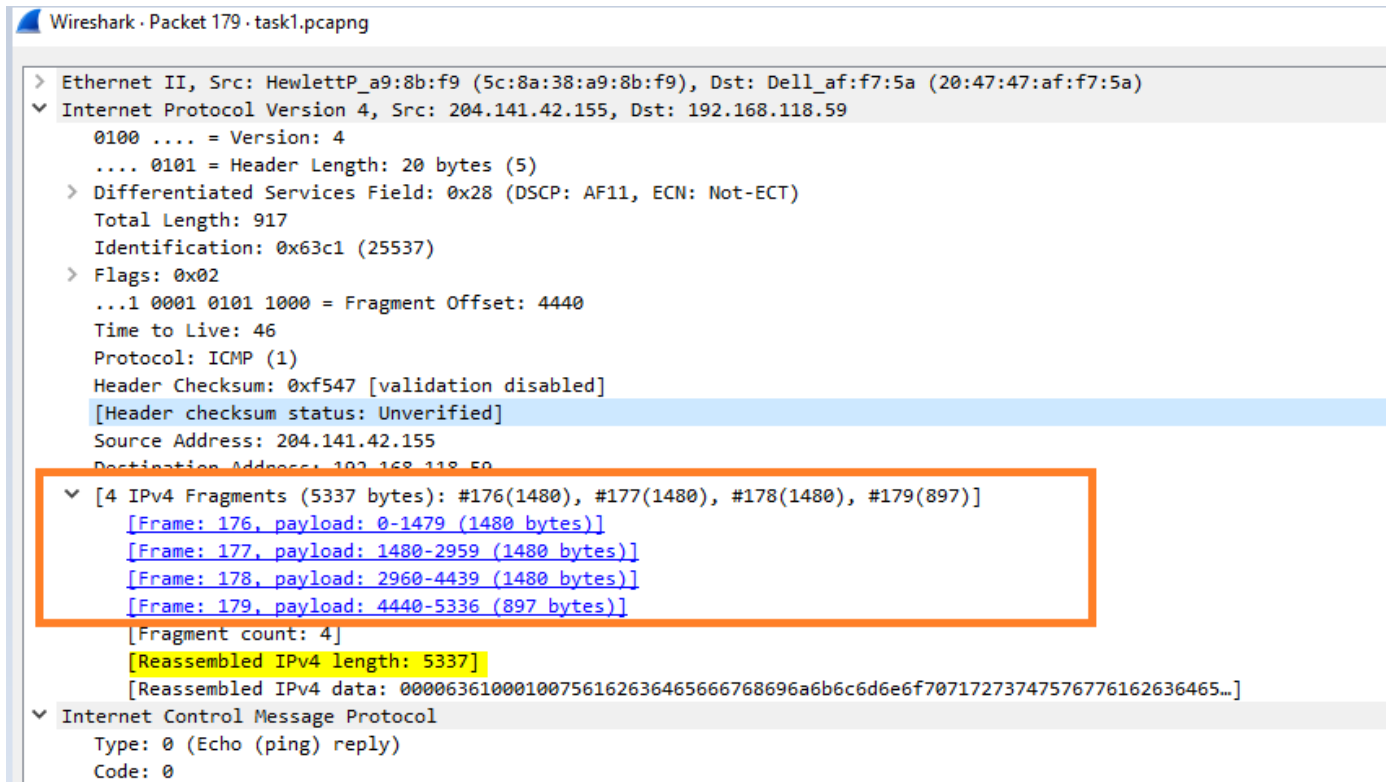
ii) The total/final length of the respective ping request and response at IP level can be checked by opening the **IPV4 of the last packet of fragment**.





The above image shows the fragment length for ping request.  
Adding all the payloads of fragments gives us the total/final length.  
Therefore, adding  $1480 + 1480 + 1480 + 897 = 5337$  bytes  
The total/final length can be seen in the **Reassembled IPV4 length** also.

Similarly we can find total/final length for ping response also.



Adding all the payloads of fragments gives us the total/final length.  
Therefore, adding  $1480 + 1480 + 1480 + 897 = 5337$  bytes  
The total/final length can be seen in the **Reassembled IPv4 length** also.

6. 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.

**Solution:**

The file filtered by IP address `ip.addr == 204.141.42.155` is saved as **task1\_filtered.pcapng** and is attached in the zip file.

## Task2: Traceroute to destination (using meter.net and size as 5329)

You will require traceroute software to execute this experiment. (If you don't have it already, install it using `sudo apt-get install traceroute`).

Start the Wireshark packet sniffer and start capturing. Open a terminal.

Execute `traceroute -q 5 <server-name> <size>`

Stop the wireshark capture and save the file for further analysis.

Answer the following using the captured trace file.

**Note:** As discussed, traceroute is not working in my system when I specify the size, hence I've used my friend's system to capture the file.

**1. What is the IP address of <server-name>? Note that you should use the traceroute command as above to find the IP Address of the server-name. Use nslookup for verification only. Appropriate screenshots shall be provided to confirm this. If you use nslookup only, then your answer will not be evaluated.**

**Solution:**

On execution of `traceroute -q 5 meter.net 5329`, the IP address obtained is

```
girish@Girishs-MacBook-Air ~ % traceroute -I -q 5 meter.net 5329
traceroute: Warning: meter.net has multiple addresses; using 104.26.4.16
traceroute to meter.net (104.26.4.16), 64 hops max, 5329 byte packets
 1 * * * * *
 2 192.168.41.41 (192.168.41.41) 12.066 ms 6.161 ms 4.364 ms 5.026 ms 10.336 ms
 3 103.232.241.70 (103.232.241.70) 4.091 ms 7.766 ms 10.768 ms 4.613 ms 6.456 ms
 4 noc-cr-in.comp.iith.ac.in (103.232.241.2) 6.640 ms 6.063 ms 6.206 ms 6.463 ms 14.012 ms
 5 noc-cn-in.comp.iith.ac.in (10.119.254.121) 3.589 ms 3.588 ms 3.461 ms 3.661 ms 3.790 ms
 6 10.160.24.5 (10.160.24.5) 5.790 ms 4.915 ms 7.975 ms 10.602 ms 11.433 ms
 7 10.255.222.33 (10.255.222.33) 5.160 ms 13.074 ms 4.777 ms 5.308 ms 4.951 ms
 8 115.247.100.29 (115.247.100.29) 6.395 ms 6.610 ms 6.534 ms 5.636 ms 6.596 ms
 9 * * * * *
10 49.44.220.145 (49.44.220.145) 36.131 ms 37.228 ms 36.421 ms 40.558 ms 36.048 ms
11 104.26.4.16 (104.26.4.16) 38.164 ms 36.882 ms 37.609 ms 36.639 ms 36.818 ms
girish@Girishs-MacBook-Air ~ %
```

IP address of meter.net verified using nslookup.

```
C:\WINDOWS\system32>nslookup meter.net
Server: dns2.iith.ac.in
Address: 192.168.36.53

Non-authoritative answer:
Name: meter.net
Addresses: 104.26.4.16
           104.26.5.16
           172.67.72.15
```

We can observe that the IP address obtained using traceroute and nslookup matches i.e; **104.26.4.16**

**2. How many hops are involved in finding the route to this server ? Use Wireshark for verifying your answer.**

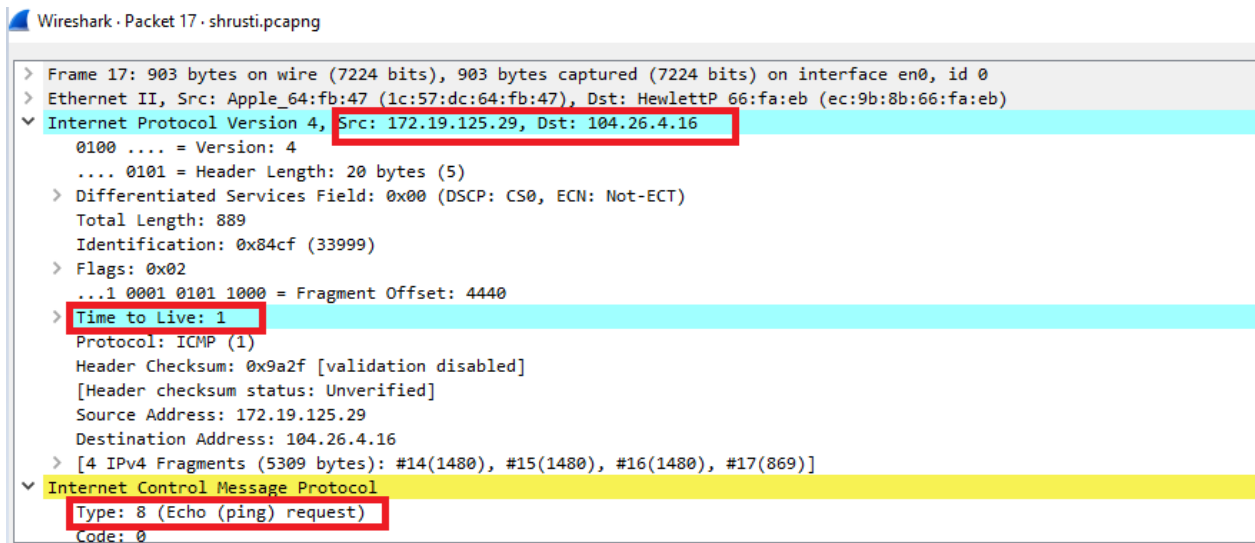
**Solution:**

**11 hops** are involved in finding the route to meter.net and in the 11th hop we reach our destination.

**Using Wireshark:**

Observe the TTL of the last request packet to know the number of hops involved in finding the route to the destination.

First request packet has TTL=1



Last request packet has TTL =11

```

> Frame 361: 903 bytes on wire (7224 bits), 903 bytes captured (7224 bits) on interface en0, id 0
> Ethernet II, Src: Apple_64:fb:47 (1c:57:dc:64:fb:47), Dst: HewlettP_66:fa:eb (ec:9b:8b:66:fa:eb)
▼ Internet Protocol Version 4, Src: 172.19.125.29, Dst: 104.26.4.16
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
    > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
        Total Length: 889
        Identification: 0x8505 (34053)
    > Flags: 0x02
        ...1 0001 0101 1000 = Fragment Offset: 4440
        Time to Live: 11
        Protocol: ICMP (1)
        Header Checksum: 0x8ff9 [validation disabled]
        [Header checksum status: Unverified]
        Source Address: 172.19.125.29
        Destination Address: 104.26.4.16
    > [4 IPv4 Fragments (5309 bytes): #358(1480), #359(1480), #360(1480), #361(869)]
▼ Internet Control Message Protocol
    Type: 8 (Echo (ping) request)
    Code: 0
    Checksum: 0x72fa [correct]
    [Checksum Status: Good]

```

3. How many total IP packets are exchanged in the communication to get the final traceroute output of <server-name>? 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.

**Solution :**

ip.addr == 104.26.4.16						
No.	Time	Source	Destination	Protocol	Length	Info
14	2.601554	172.19.125.29	104.26.4.16	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=0, ID=84cf) [Reassembled in #17]
15	2.601604	172.19.125.29	104.26.4.16	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=1480, ID=84cf) [Reassembled in #17]
16	2.601613	172.19.125.29	104.26.4.16	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=2960, ID=84cf) [Reassembled in #17]
17	2.601621	172.19.125.29	104.26.4.16	ICMP	903	Echo (ping) request id=0x84ce, seq=1/256, ttl=1 (no response found!)
25	7.602083	172.19.125.29	104.26.4.16	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=0, ID=84d0) [Reassembled in #28]
26	7.602137	172.19.125.29	104.26.4.16	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=1480, ID=84d0) [Reassembled in #28]
27	7.602219	172.19.125.29	104.26.4.16	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=2960, ID=84d0) [Reassembled in #28]
28	7.602234	172.19.125.29	104.26.4.16	ICMP	903	Echo (ping) request id=0x84ce, seq=2/512, ttl=1 (no response found!)
39	12.607265	172.19.125.29	104.26.4.16	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=0, ID=84d1) [Reassembled in #42]
40	12.607397	172.19.125.29	104.26.4.16	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=1480, ID=84d1) [Reassembled in #42]
41	12.607401	172.19.125.29	104.26.4.16	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=2960, ID=84d1) [Reassembled in #42]
42	12.607495	172.19.125.29	104.26.4.16	ICMP	903	Echo (ping) request id=0x84ce, seq=3/768, ttl=1 (no response found!)
50	17.610866	172.19.125.29	104.26.4.16	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=0, ID=84d2) [Reassembled in #53]
51	17.611030	172.19.125.29	104.26.4.16	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=1480, ID=84d2) [Reassembled in #53]
52	17.611094	172.19.125.29	104.26.4.16	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=2960, ID=84d2) [Reassembled in #53]
53	17.611116	172.19.125.29	104.26.4.16	ICMP	903	Echo (ping) request id=0x84ce, seq=4/1024, ttl=1 (no response found!)
58	22.613067	172.19.125.29	104.26.4.16	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=0, ID=84d3) [Reassembled in #61]
59	22.613143	172.19.125.29	104.26.4.16	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=1480, ID=84d3) [Reassembled in #61]
60	22.613146	172.19.125.29	104.26.4.16	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=2960, ID=84d3) [Reassembled in #61]
61	22.613149	172.19.125.29	104.26.4.16	ICMP	903	Echo (ping) request id=0x84ce, seq=5/1280, ttl=1 (no response found!)
63	27.618214	172.19.125.29	104.26.4.16	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=0, ID=84d4) [Reassembled in #66]

```

> Frame 14: 1514 bytes on wire (12112 bits), 1514 bytes captured (12112 bits) on interface en0, id 0
> Ethernet II, Src: Apple_64:fb:47 (1c:57:dc:64:fb:47), Dst: HewlettP_66:fa:eb (ec:9b:8b:66:fa:eb)
> Internet Protocol Version 4, Src: 172.19.125.29, Dst: 104.26.4.16
> Data (1480 bytes)

```

Apply the filter ***ip.addr == 104.26.4.16*** to filter out the packets from meter.net.  
 Observing the above screenshot we can say that the total IP packets exchanged in the communication to get the final traceroute output of meter.net is **280 packets**.

The number of packets exchanged in the communication can be found under Statistics  
 → Conversations as shown below.

Wireshark · Conversations · shrusti.pcapng

Ethernet · 2	IPv4 · 9	IPv6	TCP	UDP							
Address A	Address B	Packets	Bytes	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration	Bits/s A → B	Bits/s B → A
10.119.254.121	172.19.125.29	5	350	5	350	0	0	27.736463	0.0151	185 k	
10.160.24.5	172.19.125.29	5	910	5	910	0	0	27.757378	0.0359	202 k	
10.255.222.33	172.19.125.29	5	550	5	550	0	0	27.798507	0.0293	150 k	
49.44.220.145	172.19.125.29	5	350	5	350	0	0	52.918281	0.1831	15 k	
103.232.241.2	172.19.125.29	5	350	5	350	0	0	27.699246	0.0336	83 k	
103.232.241.70	172.19.125.29	5	2950	5	2950	0	0	27.661851	0.0307	767 k	
115.247.100.29	172.19.125.29	5	550	5	550	0	0	27.834186	0.0262	167 k	
172.19.125.29	104.26.4.16	240	326 k	220	299 k	20	27 k	2.601554	50.7073	47 k	
192.168.41.41	172.19.125.29	5	2950	5	2950	0	0	27.629926	0.0279	847 k	

Number of packets sent from client(172.19.125.29) to remote machine (server/router) (104.26.4.16) is : **240 packets**

Number of packets sent from the different remote machine (hop/server/router) to the local client is : **5+5+5+5+5+5+5+5 = 40 packets**

Router/hop	#packets
10.119.254.121	5
10.160.24.5	5
10.255.222.33	5
49.44.220.145	5
103.232.241.2	5
103.232.241.70	5
115.247.100.29	5
192.168.41.41	5

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

**Solution :**

In the header of the IPV4 protocol called **TTL(Time to Live)**.

The number of hops a packet travels before being discarded by a network is known as the time to live (TTL) or hop limit. The maximum range for packets is indicated by TTL values.

The datagram's TTL field is set by the sender and is reduced by each router along the path to its destination.

Traceroute simply sends a packet to the destination, at first it sets the TTL to 1, meaning the packet is allowed to take only 1 hop. The router decrements the TTL and notices now it is zero.

When TTL goes to zero, the router does 2 things: it (i) drops the packet (NOT forwarding it at all), and (ii)sends an ICMP packet BACK to the source IP address of the dropped packet. This ICMP message says that the "TTL Expired in Transit." The source IP address of the ICMP message is the address of the router itself, so traceroute can now see the IP address of the router at hop 1.

**5. 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 ?**

**Solution :**

The upper layer protocol used in sending packets from local client to remote machine is **ICMP** as shown below.



ip.addr == 104.26.4.16

No.	Time	Source	Destination	Protocol	Length	Info
14	2.601554	172.19.125.29	104.26.4.16	IPv4		1514 Fragmented IP protocol (proto=ICMP 1, off=0, ID=84cf) [Reassembled in #17]
15	2.601604	172.19.125.29	104.26.4.16	IPv4		1514 Fragmented IP protocol (proto=ICMP 1, off=1480, ID=84cf) [Reassembled in #17]
16	2.601613	172.19.125.29	104.26.4.16	IPv4		1514 Fragmented IP protocol (proto=ICMP 1, off=2960, ID=84cf) [Reassembled in #17]
17	2.601621	172.19.125.29	104.26.4.16	ICMP		903 Echo (ping) request id=0x84ce, seq=1/256, ttl=1 (no response found!)
25	7.602083	172.19.125.29	104.26.4.16	IPv4		1514 Fragmented IP protocol (proto=ICMP 1, off=0, ID=84d0) [Reassembled in #28]
26	7.602137	172.19.125.29	104.26.4.16	IPv4		1514 Fragmented IP protocol (proto=ICMP 1, off=1480, ID=84d0) [Reassembled in #28]
27	7.602219	172.19.125.29	104.26.4.16	IPv4		1514 Fragmented IP protocol (proto=ICMP 1, off=2960, ID=84d0) [Reassembled in #28]
28	7.602234	172.19.125.29	104.26.4.16	ICMP		903 Echo (ping) request id=0x84ce, seq=2/512, ttl=1 (no response found!)
39	12.607265	172.19.125.29	104.26.4.16	IPv4		1514 Fragmented IP protocol (proto=ICMP 1, off=0, ID=84d1) [Reassembled in #42]
40	12.607397	172.19.125.29	104.26.4.16	IPv4		1514 Fragmented IP protocol (proto=ICMP 1, off=1480, ID=84d1) [Reassembled in #42]
41	12.607401	172.19.125.29	104.26.4.16	IPv4		1514 Fragmented IP protocol (proto=ICMP 1, off=2960, ID=84d1) [Reassembled in #42]
42	12.607405	172.19.125.29	104.26.4.16	ICMP		903 Echo (ping) request id=0x84ce, seq=3/768, ttl=1 (no response found!)
50	17.610866	172.19.125.29	104.26.4.16	IPv4		1514 Fragmented IP protocol (proto=ICMP 1, off=0, ID=84d2) [Reassembled in #53]
51	17.611030	172.19.125.29	104.26.4.16	IPv4		1514 Fragmented IP protocol (proto=ICMP 1, off=1480, ID=84d2) [Reassembled in #53]
52	17.611094	172.19.125.29	104.26.4.16	IPv4		1514 Fragmented IP protocol (proto=ICMP 1, off=2960, ID=84d2) [Reassembled in #53]
53	17.611116	172.19.125.29	104.26.4.16	ICMP		903 Echo (ping) request id=0x84ce, seq=4/1024, ttl=1 (no response found!)
58	22.613067	172.19.125.29	104.26.4.16	IPv4		1514 Fragmented IP protocol (proto=ICMP 1, off=0, ID=84d3) [Reassembled in #61]
59	22.613143	172.19.125.29	104.26.4.16	IPv4		1514 Fragmented IP protocol (proto=ICMP 1, off=1480, ID=84d3) [Reassembled in #61]

> Frame 17: 903 bytes on wire (7224 bits), 903 bytes captured (7224 bits) on interface en0, id 0  
 > Ethernet II, Src: Apple\_64:fb:47 (1c:57:dc:64:fb:47), Dst: HewlettP\_66:fa:eb (ec:9b:8b:66:fa:eb)  
 > Internet Protocol Version 4, Src: 172.19.125.29, Dst: 104.26.4.16  
 0100 .... = Version: 4  
 .... 0101 = Header Length: 20 bytes (5)  
 > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)  
 Total Length: 889  
 Identification: 0x84cf (33999)  
 > Flags: 0x02  
 ...1 0001 0101 1000 = Fragment Offset: 4440  
 > Time to Live: 1  
 Protocol: ICMP (1)  
 Header Checksum: 0x9a2f [validation disabled]  
 [Header checksum status: Unverified]

Similarly, the upper layer protocol used in sending the packet from remote server/router to local client is also **ICMP**.

ip.addr == 104.26.4.16

No.	Time	Source	Destination	Protocol	Length	Info
315	53.025124	172.19.125.29	104.26.4.16	IPv4		1514 Fragmented IP protocol (proto=ICMP 1, off=1480, ID=84ff) [Reassembled in #317]
316	53.025194	172.19.125.29	104.26.4.16	IPv4		1514 Fragmented IP protocol (proto=ICMP 1, off=2960, ID=84ff) [Reassembled in #317]
317	53.025213	172.19.125.29	104.26.4.16	ICMP		903 Echo (ping) request id=0x84ce, seq=49/12544, ttl=10 (no response found!)
318	53.065255	49.44.220.145	172.19.125.29	ICMP		70 Time-to-live exceeded (Time to live exceeded in transit)
319	53.065596	172.19.125.29	104.26.4.16	IPv4		1514 Fragmented IP protocol (proto=ICMP 1, off=0, ID=8500) [Reassembled in #322]
320	53.065649	172.19.125.29	104.26.4.16	IPv4		1514 Fragmented IP protocol (proto=ICMP 1, off=1480, ID=8500) [Reassembled in #322]
321	53.065729	172.19.125.29	104.26.4.16	IPv4		1514 Fragmented IP protocol (proto=ICMP 1, off=2960, ID=8500) [Reassembled in #322]
322	53.065747	172.19.125.29	104.26.4.16	ICMP		903 Echo (ping) request id=0x84ce, seq=50/12800, ttl=10 (no response found!)
323	53.101349	49.44.220.145	172.19.125.29	ICMP		70 Time-to-live exceeded (Time to live exceeded in transit)
324	53.101675	172.19.125.29	104.26.4.16	IPv4		1514 Fragmented IP protocol (proto=ICMP 1, off=0, ID=8501) [Reassembled in #327]
325	53.101754	172.19.125.29	104.26.4.16	IPv4		1514 Fragmented IP protocol (proto=ICMP 1, off=1480, ID=8501) [Reassembled in #327]
326	53.101758	172.19.125.29	104.26.4.16	IPv4		1514 Fragmented IP protocol (proto=ICMP 1, off=2960, ID=8501) [Reassembled in #327]
327	53.101766	172.19.125.29	104.26.4.16	ICMP		903 Echo (ping) request id=0x84ce, seq=51/13056, ttl=11 (reply in 331)
328	53.139465	104.26.4.16	172.19.125.29	IPv4		1514 Fragmented IP protocol (proto=ICMP 1, off=0, ID=e1b7) [Reassembled in #331]
329	53.139467	104.26.4.16	172.19.125.29	IPv4		1514 Fragmented IP protocol (proto=ICMP 1, off=1480, ID=e1b7) [Reassembled in #331]
330	53.139511	104.26.4.16	172.19.125.29	IPv4		1514 Fragmented IP protocol (proto=ICMP 1, off=2960, ID=e1b7) [Reassembled in #331]
331	53.139513	104.26.4.16	172.19.125.29	ICMP		903 Echo (ping) reply id=0x84ce, seq=51/13056, ttl=53 (request in 327)
334	53.161152	172.19.125.29	104.26.4.16	IPv4		1514 Fragmented IP protocol (proto=ICMP 1, off=0, ID=8502) [Reassembled in #337]

> Frame 331: 903 bytes on wire (7224 bits), 903 bytes captured (7224 bits) on interface en0, id 0  
 > Ethernet II, Src: HewlettP\_50:49:7d (78:48:59:50:49:7d), Dst: Apple\_64:fb:47 (1c:57:dc:64:fb:47)  
 > Internet Protocol Version 4, Src: 104.26.4.16, Dst: 172.19.125.29  
 0100 .... = Version: 4  
 .... 0101 = Header Length: 20 bytes (5)  
 > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)  
 Total Length: 889  
 Identification: 0xe1b7 (57783)  
 > Flags: 0x02  
 ...1 0001 0101 1000 = Fragment Offset: 4440  
 Time to Live: 53  
 Protocol: ICMP (1)  
 Header Checksum: 0x0947 [validation disabled]  
 [Header checksum status: Unverified]



**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?**

**Solution:**

The fields that change or must change are:

- A. **Identification** : All IP packets have different IDs.
- B. **TTL (Time to live)** : Traceroute works in the way of sending the 1st Echo packet with a Time-To-Live of 1 and subsequently will increment each additional Time-To-Live packet by 1 until the destination responds or the maximum Time-To-Live is reached.
- C. **Header Checksum** : Since the router changes the IPv4 header (it decrements the TTL), it needs to calculate a new value for the checksum.

The fields that stay constant or must stay constant are :

- A. **Version** : Since we are using IPV4, it should remain the same for all packets.
- B. **Header length** : Since all these are ICMP packets
- C. **Source IP**: Since all packets are sent from same source
- D. **Destination IP** : Since we are sending to the same destination
- E. **Type of Service** : Since all packets are ICMP, they use the same Type of Service class.
- F. **Upper Layer Protocol** : Since these are ICMP 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 ?**

**Solution :**

**Client to server :**

ip.addr == 104.26.4.16						
Time	Source	Destination	Protocol	Length	Identification	Info
2.601554	172.19.125.29	104.26.4.16	IPv4	1514	0x84cf (33999)	Fragmented IP protocol (proto=ICMP 1, off=0, ID=84cf) [Reassembled in #17]
2.601604	172.19.125.29	104.26.4.16	IPv4	1514	0x84cf (33999)	Fragmented IP protocol (proto=ICMP 1, off=1480, ID=84cf) [Reassembled in #17]
2.601613	172.19.125.29	104.26.4.16	IPv4	1514	0x84cf (33999)	Fragmented IP protocol (proto=ICMP 1, off=2960, ID=84cf) [Reassembled in #17]
2.601621	172.19.125.29	104.26.4.16	ICMP	903	0x84cf (33999)	Echo (ping) request id=0x84ce, seq=1/256, ttl=1 (no response found!)
7.602083	172.19.125.29	104.26.4.16	IPv4	1514	0x84d0 (34000)	Fragmented IP protocol (proto=ICMP 1, off=0, ID=84d0) [Reassembled in #28]
7.602137	172.19.125.29	104.26.4.16	IPv4	1514	0x84d0 (34000)	Fragmented IP protocol (proto=ICMP 1, off=1480, ID=84d0) [Reassembled in #28]
7.602219	172.19.125.29	104.26.4.16	IPv4	1514	0x84d0 (34000)	Fragmented IP protocol (proto=ICMP 1, off=2960, ID=84d0) [Reassembled in #28]
7.602234	172.19.125.29	104.26.4.16	ICMP	903	0x84d0 (34000)	Echo (ping) request id=0x84ce, seq=2/512, ttl=1 (no response found!)
12.607265	172.19.125.29	104.26.4.16	IPv4	1514	0x84d1 (34001)	Fragmented IP protocol (proto=ICMP 1, off=0, ID=84d1) [Reassembled in #42]
12.607397	172.19.125.29	104.26.4.16	IPv4	1514	0x84d1 (34001)	Fragmented IP protocol (proto=ICMP 1, off=1480, ID=84d1) [Reassembled in #42]
12.607401	172.19.125.29	104.26.4.16	IPv4	1514	0x84d1 (34001)	Fragmented IP protocol (proto=ICMP 1, off=2960, ID=84d1) [Reassembled in #42]
12.607405	172.19.125.29	104.26.4.16	ICMP	903	0x84d1 (34001)	Echo (ping) request id=0x84ce, seq=3/768, ttl=1 (no response found!)
17.610866	172.19.125.29	104.26.4.16	IPv4	1514	0x84d2 (34002)	Fragmented IP protocol (proto=ICMP 1, off=0, ID=84d2) [Reassembled in #53]
17.611030	172.19.125.29	104.26.4.16	IPv4	1514	0x84d2 (34002)	Fragmented IP protocol (proto=ICMP 1, off=1480, ID=84d2) [Reassembled in #53]
17.611094	172.19.125.29	104.26.4.16	IPv4	1514	0x84d2 (34002)	Fragmented IP protocol (proto=ICMP 1, off=2960, ID=84d2) [Reassembled in #53]
17.611116	172.19.125.29	104.26.4.16	ICMP	903	0x84d2 (34002)	Echo (ping) request id=0x84ce, seq=4/1024, ttl=1 (no response found!)
22.613067	172.19.125.29	104.26.4.16	IPv4	1514	0x84d3 (34003)	Fragmented IP protocol (proto=ICMP 1, off=0, ID=84d3) [Reassembled in #61]
22.613143	172.19.125.29	104.26.4.16	IPv4	1514	0x84d3 (34003)	Fragmented IP protocol (proto=ICMP 1, off=1480, ID=84d3) [Reassembled in #61]
22.613146	172.19.125.29	104.26.4.16	IPv4	1514	0x84d3 (34003)	Fragmented IP protocol (proto=ICMP 1, off=2960, ID=84d3) [Reassembled in #61]
22.613149	172.19.125.29	104.26.4.16	ICMP	903	0x84d3 (34003)	Echo (ping) request id=0x84ce, seq=5/1280, ttl=1 (no response found!)
27.618214	172.19.125.29	104.26.4.16	IPv4	1514	0x84d4 (34004)	Fragmented IP protocol (proto=ICMP 1, off=0, ID=84d4) [Reassembled in #66]
27.618350	172.19.125.29	104.26.4.16	IPv4	1514	0x84d4 (34004)	Fragmented IP protocol (proto=ICMP 1, off=1480, ID=84d4) [Reassembled in #66]
27.618355	172.19.125.29	104.26.4.16	IPv4	1514	0x84d4 (34004)	Fragmented IP protocol (proto=ICMP 1, off=2960, ID=84d4) [Reassembled in #66]
27.618359	172.19.125.29	104.26.4.16	ICMP	903	0x84d4 (34004)	Echo (ping) request id=0x84ce, seq=6/1536, ttl=2 (no response found!)

We can see that the fragments of each request have the same identification number.

**For eg :** We can see that the first request has 4 IPV4 fragments and all their Identification value is same.

The IP **Identification** field (ID) is used to re-associate fragmented packets (they will have the same ID).

Also, observing the field “**Identification**” in the below screenshot, we can see that for each ICMP Request packet the value of Identification is increasing.

ip.addr == 104.26.4.16 && icmp						
Time	Source	Destination	Protocol	Length	Identification	Info
2.601621	172.19.125.29	104.26.4.16	ICMP	903	0x84cf (33999)	Echo (ping) request id=0x84ce, seq=1/256, ttl=1 (no response found!)
7.602234	172.19.125.29	104.26.4.16	ICMP	903	0x84d0 (34000)	Echo (ping) request id=0x84ce, seq=2/512, ttl=1 (no response found!)
12.607405	172.19.125.29	104.26.4.16	ICMP	903	0x84d1 (34001)	Echo (ping) request id=0x84ce, seq=3/768, ttl=1 (no response found!)
17.611116	172.19.125.29	104.26.4.16	ICMP	903	0x84d2 (34002)	Echo (ping) request id=0x84ce, seq=4/1024, ttl=1 (no response found!)
22.613149	172.19.125.29	104.26.4.16	ICMP	903	0x84d3 (34003)	Echo (ping) request id=0x84ce, seq=5/1280, ttl=1 (no response found!)
27.618359	172.19.125.29	104.26.4.16	ICMP	903	0x84d4 (34004)	Echo (ping) request id=0x84ce, seq=6/1536, ttl=2 (no response found!)
27.629926	192.168.41.41	172.19.125.29	ICMP	590	0x54b5 (21685),0x84d4 (34004)	Time-to-live exceeded (Time to live exceeded in transit)
27.632129	172.19.125.29	104.26.4.16	ICMP	903	0x84d5 (34005)	Echo (ping) request id=0x84ce, seq=7/1792, ttl=2 (no response found!)
27.637993	192.168.41.41	172.19.125.29	ICMP	590	0x54b6 (21686),0x84d5 (34005)	Time-to-live exceeded (Time to live exceeded in transit)
27.638206	172.19.125.29	104.26.4.16	ICMP	903	0x84d6 (34006)	Echo (ping) request id=0x84ce, seq=8/2048, ttl=2 (no response found!)
27.642383	192.168.41.41	172.19.125.29	ICMP	590	0x54b7 (21687),0x84d6 (34006)	Time-to-live exceeded (Time to live exceeded in transit)
27.642577	172.19.125.29	104.26.4.16	ICMP	903	0x84d7 (34007)	Echo (ping) request id=0x84ce, seq=9/2304, ttl=2 (no response found!)
27.647433	192.168.41.41	172.19.125.29	ICMP	590	0x54b8 (21688),0x84d7 (34007)	Time-to-live exceeded (Time to live exceeded in transit)
27.647601	172.19.125.29	104.26.4.16	ICMP	903	0x84d8 (34008)	Echo (ping) request id=0x84ce, seq=10/2560, ttl=2 (no response found!)
27.657779	192.168.41.41	172.19.125.29	ICMP	590	0x54b9 (21689),0x84d8 (34008)	Time-to-live exceeded (Time to live exceeded in transit)
27.657963	172.19.125.29	104.26.4.16	ICMP	903	0x84d9 (34009)	Echo (ping) request id=0x84ce, seq=11/2816, ttl=3 (no response found!)
27.661851	103.232.241.70	172.19.125.29	ICMP	590	0xaae2 (43746),0x84d9 (34009)	Time-to-live exceeded (Time to live exceeded in transit)
27.663154	172.19.125.29	104.26.4.16	ICMP	903	0x84da (34010)	Echo (ping) request id=0x84ce, seq=12/3072, ttl=3 (no response found!)
27.670660	103.232.241.70	172.19.125.29	ICMP	590	0xaae3 (43747),0x84da (34010)	Time-to-live exceeded (Time to live exceeded in transit)
27.670935	172.19.125.29	104.26.4.16	ICMP	903	0x84db (34011)	Echo (ping) request id=0x84ce, seq=13/3328, ttl=3 (no response found!)
27.681496	103.232.241.70	172.19.125.29	ICMP	590	0xaaee (43748),0x84db (34011)	Time-to-live exceeded (Time to live exceeded in transit)
27.681683	172.19.125.29	104.26.4.16	ICMP	903	0x84dc (34012)	Echo (ping) request id=0x84ce, seq=14/3584, ttl=3 (no response found!)

**Server/hop/router to client :**

Time	Source	Destination	Protocol	Length	Identification	Info
53.139513	104.26.4.16	172.19.125.29	ICMP	903	0xe1b7 (57783)	Echo (ping) reply id=0x84ce, seq=51/13056, ttl=53 (request in 327)
53.197737	104.26.4.16	172.19.125.29	ICMP	903	0xe1e1 (57825)	Echo (ping) reply id=0x84ce, seq=52/13312, ttl=53 (request in 337)
53.235377	104.26.4.16	172.19.125.29	ICMP	903	0xe1eb (57835)	Echo (ping) reply id=0x84ce, seq=53/13568, ttl=53 (request in 345)
53.272059	104.26.4.16	172.19.125.29	ICMP	903	0xe1f2 (57842)	Echo (ping) reply id=0x84ce, seq=54/13824, ttl=53 (request in 353)
53.308903	104.26.4.16	172.19.125.29	ICMP	903	0xe20b (57867)	Echo (ping) reply id=0x84ce, seq=55/14080, ttl=53 (request in 361)

Similarly, in ICMP Reply packets also the Identification field is increasing for each fragment.

**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).**

**Solution :**

We have many ICMP Echo requests for a hop which are further fragmented into 4 fragments.

Below table shows the details of only a few echo requests and few echo replies as it is difficult to show detail of every packet.

Packet #	Source IP	Destination IP	Fragments #	Details of Individual Fragment		
ICMP Request #17	172.19.125.29	104.26.4.16	4 (#14, #15, #16, #17)  No response found			
				Fragment #	ID field value	TTL
				14	33999	1
				15	33999	1
				16	33999	1
				17	33999	1
ICMP Request #28	172.19.125.29	104.26.4.16	4 (#25, #26, #27, #28)  No response found			
				Fragment #	ID field value	TTL

				<table><tr><td>25</td><td>34000</td><td>1</td></tr><tr><td>26</td><td>34000</td><td>1</td></tr><tr><td>27</td><td>34000</td><td>1</td></tr><tr><td>28</td><td>34000</td><td>1</td></tr></table>	25	34000	1	26	34000	1	27	34000	1	28	34000	1			
25	34000	1																	
26	34000	1																	
27	34000	1																	
28	34000	1																	
Time to Live Exceeded #67	192.168.41.41	172.19.125.29	No fragments	<table><tr><td>Fragment #</td><td>ID field value</td><td>TTL</td></tr><tr><td>67</td><td>21685</td><td>255</td></tr></table>	Fragment #	ID field value	TTL	67	21685	255									
Fragment #	ID field value	TTL																	
67	21685	255																	
ICMP Request #91	172.19.125.29	104.26.4.16	4 (#88, #89, #90, #91)  No response found	<table><tr><td>Fragment #</td><td>ID field value</td><td>TTL</td></tr><tr><td>88</td><td>34009</td><td>3</td></tr><tr><td>89</td><td>34009</td><td>3</td></tr><tr><td>90</td><td>34009</td><td>3</td></tr><tr><td>91</td><td>34009</td><td>3</td></tr></table>	Fragment #	ID field value	TTL	88	34009	3	89	34009	3	90	34009	3	91	34009	3
Fragment #	ID field value	TTL																	
88	34009	3																	
89	34009	3																	
90	34009	3																	
91	34009	3																	
Time to Live Exceeded #92	103.232.241.70	172.19.125.29	No fragments	<table><tr><td>Fragment #</td><td>ID field value</td><td>TTL</td></tr><tr><td>92</td><td>34009</td><td>63</td></tr></table>	Fragment #	ID field value	TTL	92	34009	63									
Fragment #	ID field value	TTL																	
92	34009	63																	
Time to Live Exceeded #142	10.119.254.122	172.19.125.29	No fragments	<table><tr><td>Fragment #</td><td>ID field value</td><td>TTL</td></tr><tr><td>142</td><td>34019</td><td>252</td></tr></table>	Fragment #	ID field value	TTL	142	34019	252									
Fragment #	ID field value	TTL																	
142	34019	252																	
Time to	10.160.24.5	172.19.125.	No fragments																

Live Exceeded #167		29		<table><tr><td>Fragment #</td><td>ID field value</td><td>TTL</td></tr><tr><td>167</td><td>34024</td><td>249</td></tr></table>			Fragment #	ID field value	TTL	167	34024	249									
				Fragment #	ID field value	TTL															
				167	34024	249															
ICMP request #353 (last request fragment)	172.19.125.29	104.26.4.16	4(#350, #351, #352, #353)	<table><tr><td>Fragment #</td><td>ID field value</td><td>TTL</td></tr><tr><td>350</td><td>34052</td><td>11</td></tr><tr><td>351</td><td>34052</td><td>11</td></tr><tr><td>352</td><td>34052</td><td>11</td></tr><tr><td>353</td><td>34052</td><td>11</td></tr></table>			Fragment #	ID field value	TTL	350	34052	11	351	34052	11	352	34052	11	353	34052	11
				Fragment #	ID field value	TTL															
				350	34052	11															
				351	34052	11															
				352	34052	11															
				353	34052	11															
ICMP Reply #357	104.26.4.16	172.19.125.29	4 (#354, #355, #356, #357)	<table><tr><td>Fragment #</td><td>ID field value</td><td>TTL</td></tr><tr><td>354</td><td>57842</td><td>53</td></tr><tr><td>355</td><td>57842</td><td>53</td></tr><tr><td>356</td><td>57842</td><td>53</td></tr><tr><td>357</td><td>57842</td><td>53</td></tr></table>			Fragment #	ID field value	TTL	354	57842	53	355	57842	53	356	57842	53	357	57842	53
				Fragment #	ID field value	TTL															
				354	57842	53															
				355	57842	53															
				356	57842	53															
				357	57842	53															
ICMP Reply #365	104.26.4.16	172.19.125.29	4 (#362, #363, #364, #365)	<table><tr><td>Fragment #</td><td>ID field value</td><td>TTL</td></tr><tr><td>362</td><td>57867</td><td>53</td></tr><tr><td>363</td><td>57867</td><td>53</td></tr><tr><td>364</td><td>57867</td><td>53</td></tr><tr><td>365</td><td>57867</td><td>53</td></tr></table>			Fragment #	ID field value	TTL	362	57867	53	363	57867	53	364	57867	53	365	57867	53
				Fragment #	ID field value	TTL															
				362	57867	53															
				363	57867	53															
				364	57867	53															
				365	57867	53															

**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 ?**

The **Identification field changes** for all the ICMP TTL-exceeded replies because the Identification field is a unique value.

When two or more IP datagrams have the same value it means they belong to the same fragment.

**TTL remains unchanged** because TTL for the hop/router is always the same.

**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.**

**Solution :**

```
girish@Girishs-MacBook-Air ~ % traceroute -I -q 5 meter.net 5329
traceroute: Warning: meter.net has multiple addresses; using 104.26.4.16
traceroute to meter.net (104.26.4.16), 64 hops max, 5329 byte packets
 1 * * * * *
 2 192.168.41.41 (192.168.41.41) 12.066 ms 6.161 ms 4.364 ms 5.026 ms 10.336 ms
 3 103.232.241.70 (103.232.241.70) 4.091 ms 7.766 ms 10.768 ms 4.613 ms 6.456 ms
 4 noc-cr-in.comp.iith.ac.in (103.232.241.2) 6.640 ms 6.063 ms 6.206 ms 6.463 ms 14.012 ms
 5 noc-cn-in.comp.iith.ac.in (10.119.254.121) 3.589 ms 3.588 ms 3.461 ms 3.661 ms 3.790 ms
 6 10.160.24.5 (10.160.24.5) 5.790 ms 4.915 ms 7.975 ms 10.602 ms 11.433 ms
 7 10.255.222.33 (10.255.222.33) 5.160 ms 13.074 ms 4.777 ms 5.308 ms 4.951 ms
 8 115.247.100.29 (115.247.100.29) 6.395 ms 6.610 ms 6.534 ms 5.636 ms 6.596 ms
 9 * * * * *
10 49.44.220.145 (49.44.220.145) 36.131 ms 37.228 ms 36.421 ms 40.558 ms 36.048 ms
11 104.26.4.16 (104.26.4.16) 38.164 ms 36.882 ms 37.609 ms 36.639 ms 36.818 ms
girish@Girishs-MacBook-Air ~ %
```

**Number of hops = 11**

**Number of packets sent for each hop =5**

**Hop#1 (TTL = 1)**

Since \* \* \* \* \* was shown as the response by the traceroute command, we cannot calculate average RTT for this hop.

**Hop#2 : 192.168.41.41 (TTL = 2)**

Time	Source	Destination	Protocol	Length	Identification	Frame	Info
22.613149	172.19.125.29	104.26.4.16	ICMP		903 0x84d3 (34003)	✓	Echo (ping) request id=0x84ce, seq=5/1280, ttl=1 (no response found!)
27.618214	172.19.125.29	104.26.4.16	IPv4		1514 0x84d4 (34004)	✓	Fragmented IP protocol (proto=ICMP 1, off=0, ID=84d4) [Reassembled in #66]
27.618350	172.19.125.29	104.26.4.16	IPv4		1514 0x84d4 (34004)	✓	Fragmented IP protocol (proto=ICMP 1, off=1480, ID=84d4) [Reassembled in #6]
27.618355	172.19.125.29	104.26.4.16	IPv4		1514 0x84d4 (34004)	✓	Fragmented IP protocol (proto=ICMP 1, off=2960, ID=84d4) [Reassembled in #6]
27.618359	172.19.125.29	104.26.4.16	ICMP		903 0x84d4 (34004)	✓	Echo (ping) request id=0x84ce, seq=6/1536, ttl=2 (no response found!)
27.629926	192.168.41.41	172.19.125.29	ICMP		590 0x54b5 (21685),0x84d4 (34004)	✓	Time-to-live exceeded (Time to live exceeded in transit)
27.632003	172.19.125.29	104.26.4.16	IPv4		1514 0x84d5 (34005)	✓	Fragmented IP protocol (proto=ICMP 1, off=0, ID=84d5) [Reassembled in #71]
27.632048	172.19.125.29	104.26.4.16	IPv4		1514 0x84d5 (34005)	✓	Fragmented IP protocol (proto=ICMP 1, off=1480, ID=84d5) [Reassembled in #7]
27.632111	172.19.125.29	104.26.4.16	IPv4		1514 0x84d5 (34005)	✓	Fragmented IP protocol (proto=ICMP 1, off=2960, ID=84d5) [Reassembled in #7]
27.632129	172.19.125.29	104.26.4.16	ICMP		903 0x84d5 (34005)	✓	Echo (ping) request id=0x84ce, seq=7/1792, ttl=2 (no response found!)
27.637993	192.168.41.41	172.19.125.29	ICMP		590 0x54b6 (21686),0x84d5 (34005)	✓	Time-to-live exceeded (Time to live exceeded in transit)
27.638133	172.19.125.29	104.26.4.16	IPv4		1514 0x84d6 (34006)	✓	Fragmented IP protocol (proto=ICMP 1, off=0, ID=84d6) [Reassembled in #76]
27.638164	172.19.125.29	104.26.4.16	IPv4		1514 0x84d6 (34006)	✓	Fragmented IP protocol (proto=ICMP 1, off=1480, ID=84d6) [Reassembled in #7]
27.638189	172.19.125.29	104.26.4.16	IPv4		1514 0x84d6 (34006)	✓	Fragmented IP protocol (proto=ICMP 1, off=2960, ID=84d6) [Reassembled in #7]
27.638206	172.19.125.29	104.26.4.16	ICMP		903 0x84d6 (34006)	✓	Echo (ping) request id=0x84ce, seq=8/2048, ttl=2 (no response found!)
27.642383	192.168.41.41	172.19.125.29	ICMP		590 0x54b7 (21687),0x84d6 (34006)	✓	Time-to-live exceeded (Time to live exceeded in transit)
27.642505	172.19.125.29	104.26.4.16	IPv4		1514 0x84d7 (34007)	✓	Fragmented IP protocol (proto=ICMP 1, off=0, ID=84d7) [Reassembled in #81]
27.642534	172.19.125.29	104.26.4.16	IPv4		1514 0x84d7 (34007)	✓	Fragmented IP protocol (proto=ICMP 1, off=1480, ID=84d7) [Reassembled in #8]
27.642559	172.19.125.29	104.26.4.16	IPv4		1514 0x84d7 (34007)	✓	Fragmented IP protocol (proto=ICMP 1, off=2960, ID=84d7) [Reassembled in #8]
27.642577	172.19.125.29	104.26.4.16	ICMP		903 0x84d7 (34007)	✓	Echo (ping) request id=0x84ce, seq=9/2304, ttl=2 (no response found!)
27.647433	192.168.41.41	172.19.125.29	ICMP		590 0x54b8 (21688),0x84d7 (34007)	✓	Time-to-live exceeded (Time to live exceeded in transit)
27.647533	172.19.125.29	104.26.4.16	IPv4		1514 0x84d8 (34008)	✓	Fragmented IP protocol (proto=ICMP 1, off=0, ID=84d8) [Reassembled in #86]
27.647557	172.19.125.29	104.26.4.16	IPv4		1514 0x84d8 (34008)	✓	Fragmented IP protocol (proto=ICMP 1, off=1480, ID=84d8) [Reassembled in #8]
27.647577	172.19.125.29	104.26.4.16	IPv4		1514 0x84d8 (34008)	✓	Fragmented IP protocol (proto=ICMP 1, off=2960, ID=84d8) [Reassembled in #8]
27.647601	172.19.125.29	104.26.4.16	ICMP		903 0x84d8 (34008)	✓	Echo (ping) request id=0x84ce, seq=10/2560, ttl=2 (no response found!)
27.657779	192.168.41.41	172.19.125.29	ICMP		590 0x54b9 (21689),0x84d8 (34008)	✓	Time-to-live exceeded (Time to live exceeded in transit)
27.657896	172.19.125.29	104.26.4.16	IPv4		1514 0x84d9 (34009)	✓	Fragmented IP protocol (proto=ICMP 1, off=0, ID=84d9) [Reassembled in #91]
27.657923	172.19.125.29	104.26.4.16	IPv4		1514 0x84d9 (34009)	✓	Fragmented IP protocol (proto=ICMP 1, off=1480, ID=84d9) [Reassembled in #9]
27.657947	172.19.125.29	104.26.4.16	IPv4		1514 0x84d9 (34009)	✓	Fragmented IP protocol (proto=ICMP 1, off=2960, ID=84d9) [Reassembled in #9]
27.657963	172.19.125.29	104.26.4.16	ICMP		903 0x84d9 (34009)	✓	Echo (ping) request id=0x84ce, seq=11/2816, ttl=3 (no response found!)
27.661851	103.232.241.70	172.19.125.29	ICMP		590 0xaae2 (43746),0x84d9 (34009)	✓	Time-to-live exceeded (Time to live exceeded in transit) activate Windows.

**RTT = (Time of TTL exceeded packet - Time of 1st fragment before TTL exceeded packet is sent)**

**Average RTT = (Sum of all RTT) / number of packets**

Packet #	RTT
1	27.629926 - 27.618214 = 0.011712 = 11.712ms
2	27.637993 - 27.632003 = 0.00599 = 5.99ms
3	27.642383 - 27.638133 = 0.00425 = 4.25ms
4	27.647433 - 27.642505 = 0.004928 = 4.928ms
5	27.657779 - 27.647533 = 0.010246 = 10.246ms

**Note :** We can verify this RTT with the RTT values obtained through traceroute, they almost match each other.

Average RTT = (11.712 + 5.99 + 4.25 + 4.928 + 10.246)/5 = **7.4252ms**

**Hop#3 : 103.232. 241.70 (TTL =3)**

Packet #	RTT
1	27.661851 - 27.657896 =0.003955 = 3.9955ms

2	$27.670660 - 27.663080 = 0.00758 = 7.58\text{ms}$
3	$27.681496 - 27.670867 = 0.010629 = 10.629\text{ms}$
4	$27.686131 - 27.681618 = 0.004513 = 4.513\text{ms}$
5	$27.692596 - 27.686228 = 0.006368 = 6.368\text{ms}$

Average RTT =  $(3.9955 + 7.58 + 10.629 + 4.513 + 6.368)/5 = \mathbf{6.6171\text{ms}}$

**Hop#4 : 103.232. 241.2 (TTL = 4)**

Packet #	RTT
1	$27.699246 - 27.692699 = 6.547\text{ms}$
2	$27.706156 - 27.700193 = 5.963\text{ms}$
3	$27.712368 - 27.706247 = 6.121\text{ms}$
4	$27.718843 - 27.712468 = 6.375\text{ms}$
5	$27.732841 - 27.718935 = 13.906\text{ms}$

Average RTT =  $(6.547 + 5.963 + 6.121 + 6.375 + 13.906)/5 = \mathbf{7.7824\text{ms}}$

**Hop#5 : 10.119.254.121 (TTL =5)**

Packet #	RTT
1	$27.736463 - 27.732966 = 3.497\text{ms}$
2	$27.740636 - 27.737119 = 3.517\text{ms}$
3	$27.744090 - 27.740719 = 3.371\text{ms}$
4	$27.747782 - 27.744214 = 3.568\text{ms}$
5	$27.757378 - 27.751674 = 5.704\text{ms}$

Average RTT =  $(3.497 + 3.517 + 3.371 + 3.568 + 5.704)/5 = \mathbf{3.9314\text{ms}}$

**Hop#6 : 10.160.24.5 (TTL =6)**



Packet #	RTT
1	27.757378 - 27.751674 = 5.704ms
2	27.763282 - 27.758461 = 4.821ms
3	27.771247 - 27.763373 = 7.874ms
4	27.781857 - 27.771374 = 10.483ms
5	27.793322 - 27.782000 = 11.322ms

Average RTT = (5.704 + 4.821 + 7.874 + 10.483 + 11.322)/5 = **8.0408ms**

**Hop#7 :10.255.222.33 (TTL = 7)**

Packet #	RTT
1	5.062ms
2	12.959ms
3	6.302ms
4	5.235ms
5	4.858ms

Average RTT = (5.062 + 12.959 + 6.302 + 5.235 + 4.858)/5 = **6.8832ms**

**Hop#8 : 115.247.100.29 (TTL = 8)**

Packet #	RTT
1	6.302ms
2	6.454ms
3	6.443ms
4	5.538ms
5	6.458ms

Average RTT = (6.302 + 6.454 + 6.443 + 5.538 + 6.458)/5 = **6.239ms**

**Hop#9 : (TTL = 9)**

Since \* \* \* \* \* was shown as the response by the traceroute command, we cannot calculate average RTT for this hop.

**Hop#10 : 49.44.220.145 (TTL = 10)**

Packet #	RTT
1	35.817ms
2	36.929ms
3	36.117ms
4	40.248ms
5	35.753ms

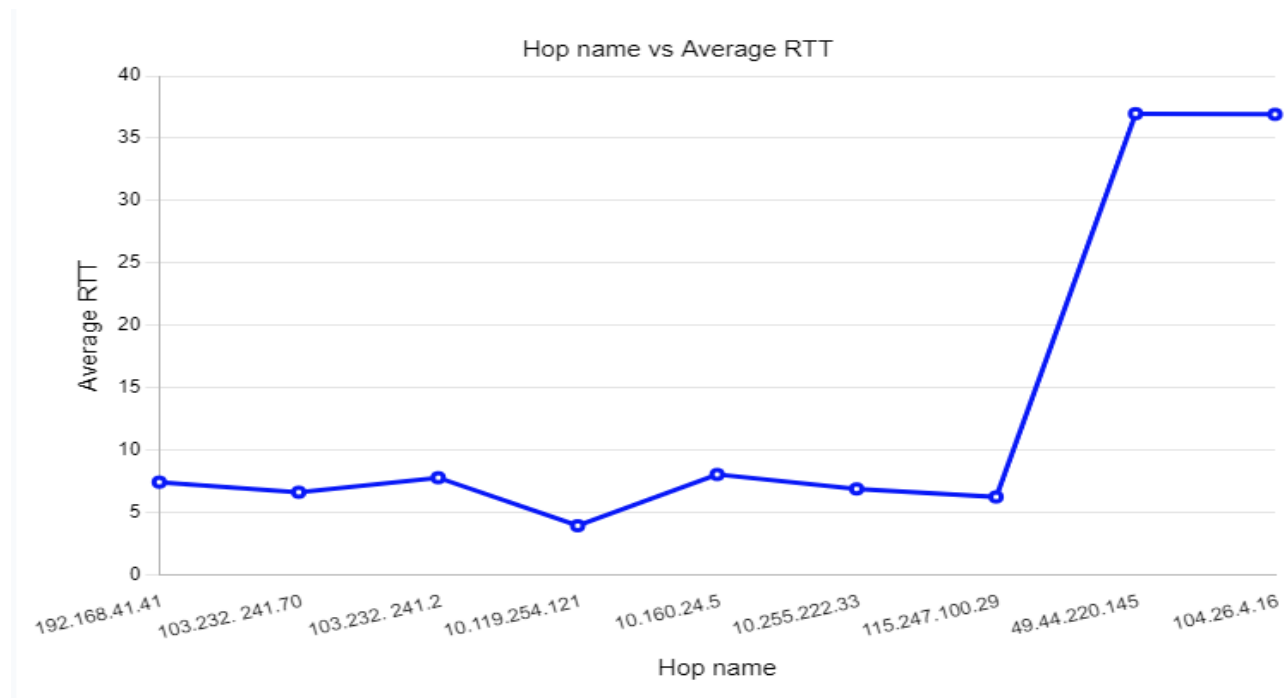
Average RTT =  $(35.817 + 36.929 + 36.117 + 40.248 + 35.753)/5 = 36.9728\text{ms}$

**Hop#11 : 104.26.4.16 ( TTL = 11)**

Packet #	RTT
1	37.838ms
2	36.585ms
3	37.316ms
4	36.324ms
5	36.514ms

Average RTT =  $(37.838 + 36.585 + 37.316 + 36.324 + 36.514)/5 = 36.9154\text{ms}$

**Graph for Hop/Router name vs Average RTT :**



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

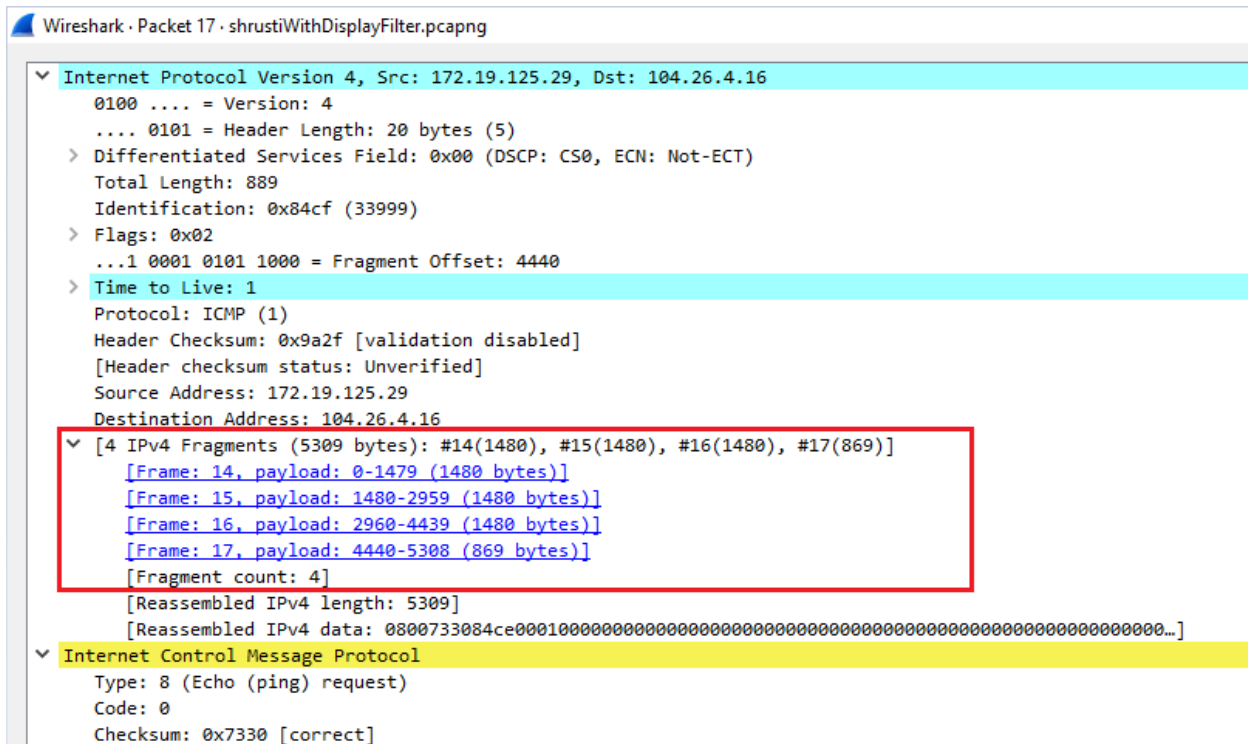
**Explain how you determine whether or not the datagram has been fragmented and where does the fragmentation end.**

**Solution :**

**Case 1: Pick end of the fragment packet**

Consider packet 17 which is from client to server. To know whether the IP datagram is fragmented or not : Open the packet → IPV4 → Fragments

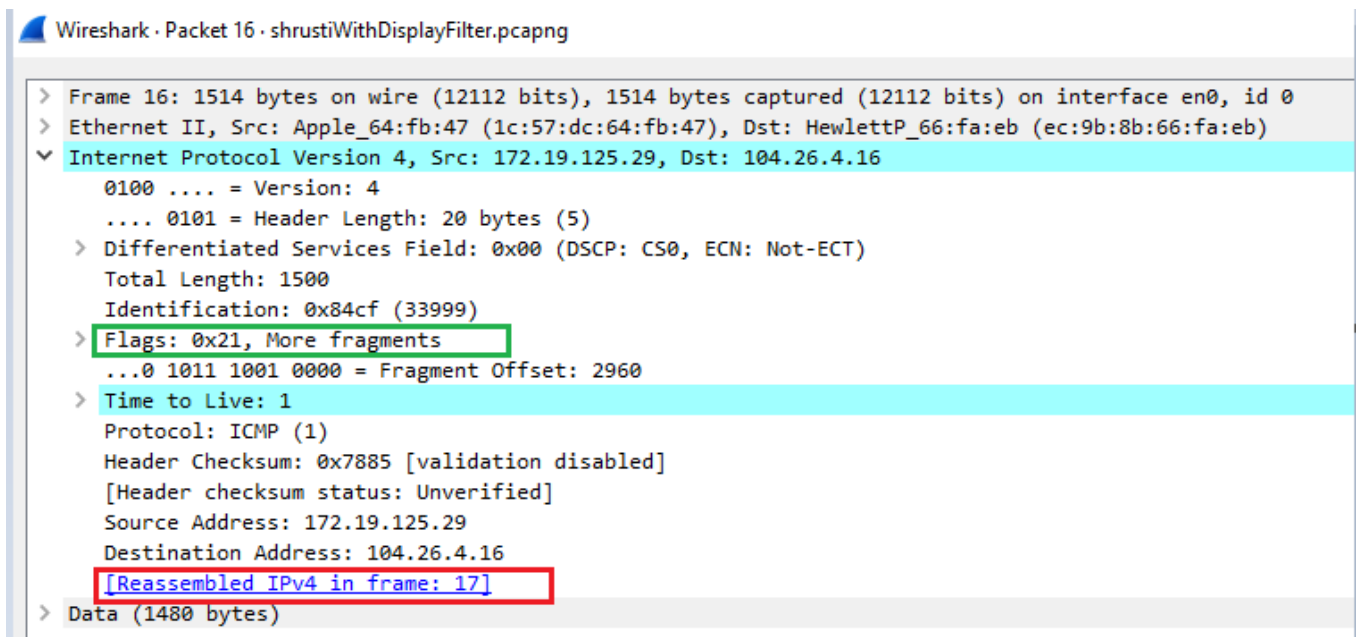
ip.addr == 104.26.4.16						
Time	Source	Destination	Protocol	Length	Identification	Info
2.601554	172.19.125.29	104.26.4.16	IPv4	1514	0x84cf (33999)	Fragmented IP protocol (proto=ICMP 1, off=0, ID=84cf) [Reassembled in #17]
2.601604	172.19.125.29	104.26.4.16	IPv4	1514	0x84cf (33999)	Fragmented IP protocol (proto=ICMP 1, off=1480, ID=84cf) [Reassembled in #17]
2.601613	172.19.125.29	104.26.4.16	IPv4	1514	0x84cf (33999)	Fragmented IP protocol (proto=ICMP 1, off=2960, ID=84cf) [Reassembled in #17]
2.601621	172.19.125.29	104.26.4.16	ICMP	903	0x84cf (33999)	Echo (ping) request id=0x84ce, seq=1/256, ttl=1 (no response found!)
7.602083	172.19.125.29	104.26.4.16	IPv4	1514	0x84d0 (34000)	Fragmented IP protocol (proto=ICMP 1, off=0, ID=84d0) [Reassembled in #28]
7.602137	172.19.125.29	104.26.4.16	IPv4	1514	0x84d0 (34000)	Fragmented IP protocol (proto=ICMP 1, off=1480, ID=84d0) [Reassembled in #28]
7.602219	172.19.125.29	104.26.4.16	IPv4	1514	0x84d0 (34000)	Fragmented IP protocol (proto=ICMP 1, off=2960, ID=84d0) [Reassembled in #28]
7.602234	172.19.125.29	104.26.4.16	ICMP	903	0x84d0 (34000)	Echo (ping) request id=0x84ce, seq=2/512, ttl=1 (no response found!)
12.607265	172.19.125.29	104.26.4.16	IPv4	1514	0x84d1 (34001)	Fragmented IP protocol (proto=ICMP 1, off=0, ID=84d1) [Reassembled in #42]
12.607397	172.19.125.29	104.26.4.16	IPv4	1514	0x84d1 (34001)	Fragmented IP protocol (proto=ICMP 1, off=1480, ID=84d1) [Reassembled in #42]
12.607401	172.19.125.29	104.26.4.16	IPv4	1514	0x84d1 (34001)	Fragmented IP protocol (proto=ICMP 1, off=2960, ID=84d1) [Reassembled in #42]
12.607405	172.19.125.29	104.26.4.16	ICMP	903	0x84d1 (34001)	Echo (ping) request id=0x84ce, seq=3/768, ttl=1 (no response found!)
17.610866	172.19.125.29	104.26.4.16	IPv4	1514	0x84d2 (34002)	Fragmented IP protocol (proto=ICMP 1, off=0, ID=84d2) [Reassembled in #53]
17.611030	172.19.125.29	104.26.4.16	IPv4	1514	0x84d2 (34002)	Fragmented IP protocol (proto=ICMP 1, off=1480, ID=84d2) [Reassembled in #53]
17.611094	172.19.125.29	104.26.4.16	IPv4	1514	0x84d2 (34002)	Fragmented IP protocol (proto=ICMP 1, off=2960, ID=84d2) [Reassembled in #53]
17.611116	172.19.125.29	104.26.4.16	ICMP	903	0x84d2 (34002)	Echo (ping) request id=0x84ce, seq=4/1024, ttl=1 (no response found!)
22.613067	172.19.125.29	104.26.4.16	IPv4	1514	0x84d3 (34003)	Fragmented IP protocol (proto=ICMP 1, off=0, ID=84d3) [Reassembled in #61]
22.613143	172.19.125.29	104.26.4.16	IPv4	1514	0x84d3 (34003)	Fragmented IP protocol (proto=ICMP 1, off=1480, ID=84d3) [Reassembled in #61]
22.613146	172.19.125.29	104.26.4.16	IPv4	1514	0x84d3 (34003)	Fragmented IP protocol (proto=ICMP 1, off=2960, ID=84d3) [Reassembled in #61]
22.613149	172.19.125.29	104.26.4.16	ICMP	903	0x84d3 (34003)	Echo (ping) request id=0x84ce, seq=5/1280, ttl=1 (no response found!)



For the above packet #17 we can observe that there exists 4 IPV4 fragments with packet #14, #15, #16 and #17.

### Case 2: Pick a packet which is not the end of the fragment

Let's consider the packet #16 as it is not the end of fragment packet, we can know it's fragmentation end is at packet #17 by observing the ***"Reassembled IPV4 in frame"***.



```
Wireshark · Packet 16 · shrutiWithDisplayFilter.pcapng
> Frame 16: 1514 bytes on wire (12112 bits), 1514 bytes captured (12112 bits) on interface en0, id 0
> Ethernet II, Src: Apple_64:fb:47 (1c:57:dc:64:fb:47), Dst: HewlettP_66:fa:eb (ec:9b:8b:66:fa:eb)
> Internet Protocol Version 4, Src: 172.19.125.29, Dst: 104.26.4.16
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
> Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
  Total Length: 1500
  Identification: 0x84cf (33999)
> Flags: 0x21, More fragments
  ...0 1011 1001 0000 = Fragment Offset: 2960
> Time to Live: 1
  Protocol: ICMP (1)
  Header Checksum: 0x7885 [validation disabled]
  [Header checksum status: Unverified]
  Source Address: 172.19.125.29
  Destination Address: 104.26.4.16
  [Reassembled IPv4 in frame: 17]
> Data (1480 bytes)
```

The end of the fragmentation can be known by the MF(More Fragments) bit. MF bit is set for all fragments except the last one.

Observe the packet #17 and #16, as #17 is the last fragment “*More fragments* “ is not specified, whereas in packet #16, More fragments is specified(highlighted by red color box) as it is not the last packet of fragmentation.

**11. 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.**

Solution :

The file filtered by IP address `ip.addr == 104.26.4.16` is saved as **task2\_filtered.pcapng** and is attached in the zip file.

## Task3 Ping versus Traceroute

**1. Comment on your understanding of differences between traceroute and ping commands executed for server-name (in Task1 and Task2) using the respective wireshark traces.**

**List the IP fields which indicate the differences in the behavior of execution. Place at least 1 screenshot of wireshark from respective traces showing relevant display filters as a proof of explaining the difference.**

Solution :

**i) Definition :**

**Task1 :**

“Ping” sends a series of ICMP Echo Request packets to the destination, with a maximum TTL (time-to-live) value of 255. If the destination is reachable and configured to respond to ICMP Echo Requests, it responds with an ICMP Echo Reply packet. Ping then measures and displays the RTT (round-trip-time) - the time between sending the ICMP Echo Request and receiving the ICMP Echo Reply.

Here, we have executed a ping command for the server-name “www.zoho.com” with size 5329 where the

Minimum RTT = 251ms

Maximum RTT = 255ms

Average RTT = 252ms

```
C:\WINDOWS\system32>ping -l 5329 www.zoho.com -n 5

Pinging gsite.zohocdn.com [204.141.42.155] with 5329 bytes of data:
Reply from 204.141.42.155: bytes=5329 time=252ms TTL=46
Reply from 204.141.42.155: bytes=5329 time=251ms TTL=46
Reply from 204.141.42.155: bytes=5329 time=251ms TTL=46
Reply from 204.141.42.155: bytes=5329 time=255ms TTL=46
Reply from 204.141.42.155: bytes=5329 time=251ms TTL=46

Ping statistics for 204.141.42.155:
    Packets: Sent = 5, Received = 5, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 251ms, Maximum = 255ms, Average = 252ms
```

**Task2:**

Traceroute also sends ICMP Echo Request messages, but sets the time to live (TTL) in the first packet to just 1, then next to 2, etc, until a reply is received from the destination. As the routers on the path to the destination attempt to forward the ICMP Echo Request message they decrease the TTL at each stop by one, and when the TTL reaches zero, they return an ICMP TTL Exceeded message.

The traceroute program is helpful when the destination is not reachable to figure out where the interruption is.

Here, we have executed the traceroute command for “meter.net” with size 5329 and the below screenshot shows the list of all the routers on the way to the destination.

```

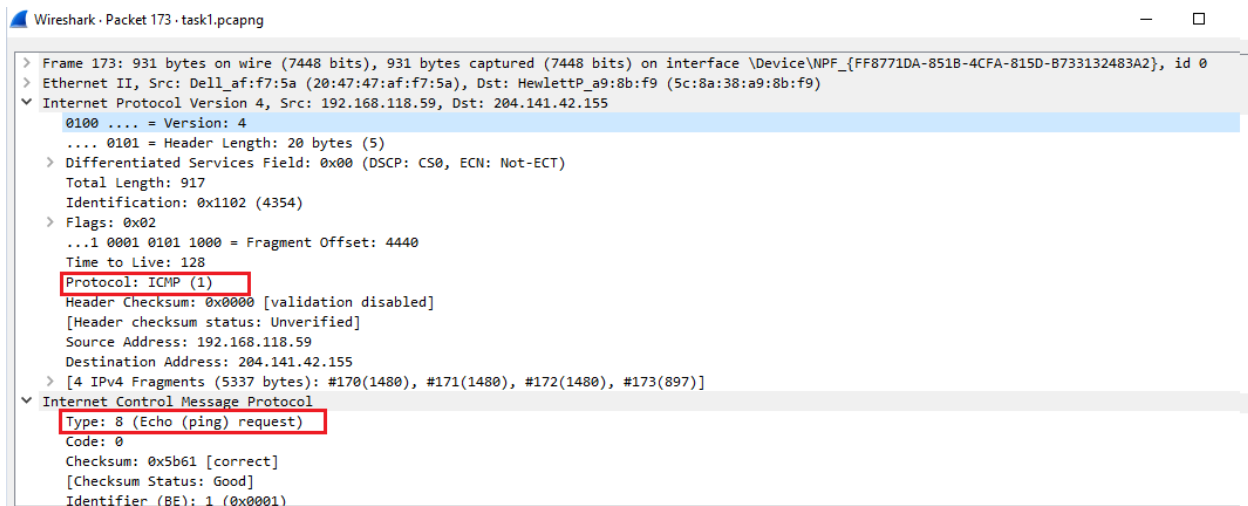
girish@Girishs-MacBook-Air ~ % traceroute -I -q 5 meter.net 5329
traceroute: Warning: meter.net has multiple addresses; using 104.26.4.16
traceroute to meter.net (104.26.4.16), 64 hops max, 5329 byte packets
 1 * * * * *
 2 192.168.41.41 (192.168.41.41) 12.066 ms 6.161 ms 4.364 ms 5.026 ms 10.336 ms
 3 103.232.241.70 (103.232.241.70) 4.091 ms 7.766 ms 10.768 ms 4.613 ms 6.456 ms
 4 noc-cr-in.comp.iith.ac.in (103.232.241.2) 6.640 ms 6.063 ms 6.206 ms 6.463 ms 14.012 ms
 5 noc-cn-in.comp.iith.ac.in (10.119.254.121) 3.589 ms 3.588 ms 3.461 ms 3.661 ms 3.790 ms
 6 10.160.24.5 (10.160.24.5) 5.790 ms 4.915 ms 7.975 ms 10.602 ms 11.433 ms
 7 10.255.222.33 (10.255.222.33) 5.160 ms 13.074 ms 4.777 ms 5.308 ms 4.951 ms
 8 115.247.100.29 (115.247.100.29) 6.395 ms 6.610 ms 6.534 ms 5.636 ms 6.596 ms
 9 * * * * *
10 49.44.220.145 (49.44.220.145) 36.131 ms 37.228 ms 36.421 ms 40.558 ms 36.048 ms
11 104.26.4.16 (104.26.4.16) 38.164 ms 36.882 ms 37.609 ms 36.639 ms 36.818 ms
girish@Girishs-MacBook-Air ~ %

```

ii) The IP fields which indicate the difference in the behavior of the execution are

### Task1:

In Ping, we set IPV4 Protocol ICMP to 1 and **Type 8** for Echo Request and **Type 0** for Echo reply as shown below.



Wireshark · Packet 179 · task1.pcapng

> Frame 179: 931 bytes on wire (7448 bits), 931 bytes captured (7448 bits) on interface \Device\NPF\_{FF8771DA-851B-4CFA-815D-B733132483A2}, id 0

> Ethernet II, Src: HewlettP\_a9:8b:f9 (5c:8a:38:a9:8b:f9), Dst: Dell\_af:f7:5a (20:47:47:af:f7:5a)

▼ Internet Protocol Version 4, Src: 204.141.42.155, Dst: 192.168.118.59

0100 .... = Version: 4

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

> Differentiated Services Field: 0x28 (DSCP: AF11, ECN: Not-ECT)

Total Length: 917

Identification: 0x63c1 (25537)

> Flags: 0x02

...1 0001 0101 1000 = Fragment Offset: 4440

Time to Live: 46

Protocol: ICMP (1)

Header Checksum: 0xf547 [validation disabled]

[Header checksum status: Unverified]

Source Address: 204.141.42.155

Destination Address: 192.168.118.59

> [4 IPv4 Fragments (5337 bytes): #176(1480), #177(1480), #178(1480), #179(897)]

▼ Internet Control Message Protocol

Type: 0 (Echo (ping) reply)

Code: 0

Checksum: 0x6361 [correct]

[Checksum Status: Good]

Identifier (BE): 1 (0x0001)

## Task2:

Whereas in traceroute we use TTL field to set the number of hops. Initially we set TTL =1 and increase it by one for every next hop/router. When TTL =0 the hop sends the TTL Exceeded back to the host, using IP addresses of hops we can check from which hop we got the response.

Below screenshot shows the Request packet where TTL is set to 1 to reach the first hop between source and destination.

Wireshark · Packet 17 · shrutiWithDisplayFilter.pcapng

> Frame 17: 903 bytes on wire (7224 bits), 903 bytes captured (7224 bits) on interface en0, id 0

> Ethernet II, Src: Apple\_64:fb:47 (1c:57:dc:64:fb:47), Dst: HewlettP 66:fa:eb (ec:9b:8b:66:fa:eb)

▼ Internet Protocol Version 4, Src: 172.19.125.29, Dst: 104.26.4.16

0100 .... = Version: 4

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

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

Total Length: 889

Identification: 0x84cf (33999)

> Flags: 0x02

...1 0001 0101 1000 = Fragment Offset: 4440

> Time to Live: 1

Protocol: ICMP (1)

Header Checksum: 0x9a2f [validation disabled]

[Header checksum status: Unverified]

Source Address: 172.19.125.29

Destination Address: 104.26.4.16

> [4 IPv4 Fragments (5309 bytes): #14(1480), #15(1480), #16(1480), #17(869)]

▼ Internet Control Message Protocol

Type: 8 (Echo (ping) request)

## iii) TTL

## Task1:

All ICMP Echo requests have the same TTL i.e; 128.



Time	Source	Destination	Protocol	Length	Identification	Info
12.044494	192.168.118.59	204.141.42.155	IPv4	1514	0x1102 (4354)	Fragmented IP protocol (proto=ICMP 1, off=0, ID=1102) [Reassembled in #173]
12.044494	192.168.118.59	204.141.42.155	IPv4	1514	0x1102 (4354)	Fragmented IP protocol (proto=ICMP 1, off=1480, ID=1102) [Reassembled in #173]
12.044494	192.168.118.59	204.141.42.155	IPv4	1514	0x1102 (4354)	Fragmented IP protocol (proto=ICMP 1, off=2960, ID=1102) [Reassembled in #173]
12.044494	192.168.118.59	204.141.42.155	ICMP	931	0x1102 (4354)	Echo (ping) request id=0x0001, seq=117/29952, ttl=128 (reply in 179)
12.296304	204.141.42.155	192.168.118.59	IPv4	1514	0x63c1 (25537)	Fragmented IP protocol (proto=ICMP 1, off=0, ID=63c1) [Reassembled in #179]
12.296304	204.141.42.155	192.168.118.59	IPv4	1514	0x63c1 (25537)	Fragmented IP protocol (proto=ICMP 1, off=1480, ID=63c1) [Reassembled in #179]
12.296304	204.141.42.155	192.168.118.59	IPv4	1514	0x63c1 (25537)	Fragmented IP protocol (proto=ICMP 1, off=2960, ID=63c1) [Reassembled in #179]
12.296304	204.141.42.155	192.168.118.59	ICMP	931	0x63c1 (25537)	Echo (ping) reply id=0x0001, seq=117/29952, ttl=46 (request in 173)
13.061623	192.168.118.59	204.141.42.155	IPv4	1514	0x1103 (4355)	Fragmented IP protocol (proto=ICMP 1, off=0, ID=1103) [Reassembled in #189]
13.061623	192.168.118.59	204.141.42.155	IPv4	1514	0x1103 (4355)	Fragmented IP protocol (proto=ICMP 1, off=1480, ID=1103) [Reassembled in #189]
13.061623	192.168.118.59	204.141.42.155	IPv4	1514	0x1103 (4355)	Fragmented IP protocol (proto=ICMP 1, off=2960, ID=1103) [Reassembled in #189]
13.061623	192.168.118.59	204.141.42.155	ICMP	931	0x1103 (4355)	Echo (ping) request id=0x0001, seq=118/30208, ttl=128 (reply in 194)
13.313118	204.141.42.155	192.168.118.59	IPv4	1514	0x6609 (26121)	Fragmented IP protocol (proto=ICMP 1, off=0, ID=6609) [Reassembled in #194]
13.313118	204.141.42.155	192.168.118.59	IPv4	1514	0x6609 (26121)	Fragmented IP protocol (proto=ICMP 1, off=1480, ID=6609) [Reassembled in #194]
13.313118	204.141.42.155	192.168.118.59	IPv4	1514	0x6609 (26121)	Fragmented IP protocol (proto=ICMP 1, off=2960, ID=6609) [Reassembled in #194]
13.313118	204.141.42.155	192.168.118.59	ICMP	931	0x6609 (26121)	Echo (ping) reply id=0x0001, seq=118/30208, ttl=46 (request in 189)
14.077178	192.168.118.59	204.141.42.155	IPv4	1514	0x1104 (4356)	Fragmented IP protocol (proto=ICMP 1, off=0, ID=1104) [Reassembled in #217]
14.077178	192.168.118.59	204.141.42.155	IPv4	1514	0x1104 (4356)	Fragmented IP protocol (proto=ICMP 1, off=1480, ID=1104) [Reassembled in #217]
14.077178	192.168.118.59	204.141.42.155	IPv4	1514	0x1104 (4356)	Fragmented IP protocol (proto=ICMP 1, off=2960, ID=1104) [Reassembled in #217]
14.077178	192.168.118.59	204.141.42.155	ICMP	931	0x1104 (4356)	Echo (ping) request id=0x0001, seq=119/30464, ttl=128 (reply in 226)
14.328803	204.141.42.155	192.168.118.59	IPv4	1514	0x68de (26846)	Fragmented IP protocol (proto=ICMP 1, off=0, ID=68de) [Reassembled in #226]
14.328803	204.141.42.155	192.168.118.59	IPv4	1514	0x68de (26846)	Fragmented IP protocol (proto=ICMP 1, off=1480, ID=68de) [Reassembled in #226]
14.328803	204.141.42.155	192.168.118.59	IPv4	1514	0x68de (26846)	Fragmented IP protocol (proto=ICMP 1, off=2960, ID=68de) [Reassembled in #226]
14.328803	204.141.42.155	192.168.118.59	ICMP	931	0x68de (26846)	Echo (ping) reply id=0x0001, seq=119/30464, ttl=46 (request in 217)
15.106903	192.168.118.59	204.141.42.155	IPv4	1514	0x1105 (4357)	Fragmented IP protocol (proto=ICMP 1, off=0, ID=1105) [Reassembled in #236]
15.106903	192.168.118.59	204.141.42.155	IPv4	1514	0x1105 (4357)	Fragmented IP protocol (proto=ICMP 1, off=1480, ID=1105) [Reassembled in #236]
15.106903	192.168.118.59	204.141.42.155	IPv4	1514	0x1105 (4357)	Fragmented IP protocol (proto=ICMP 1, off=2960, ID=1105) [Reassembled in #236]
15.106903	192.168.118.59	204.141.42.155	ICMP	931	0x1105 (4357)	Echo (ping) request id=0x0001, seq=120/30720, ttl=128 (reply in 242)

## Task2:

The TTL values are incrementing from 1,2,3....

Time	Source	Destination	Protocol	Length	Identification	Info
7.602234	172.19.125.29	104.26.4.16	ICMP	903	0x84d0 (34000)	Echo (ping) request id=0x84ce, seq=2/512, ttl=1 (no response found!)
12.607265	172.19.125.29	104.26.4.16	IPv4	1514	0x84d1 (34001)	Fragmented IP protocol (proto=ICMP 1, off=0, ID=84d1) [Reassembled in #42]
12.607397	172.19.125.29	104.26.4.16	IPv4	1514	0x84d1 (34001)	Fragmented IP protocol (proto=ICMP 1, off=1480, ID=84d1) [Reassembled in #42]
12.607401	172.19.125.29	104.26.4.16	IPv4	1514	0x84d1 (34001)	Fragmented IP protocol (proto=ICMP 1, off=2960, ID=84d1) [Reassembled in #42]
12.607405	172.19.125.29	104.26.4.16	ICMP	903	0x84d1 (34001)	Echo (ping) request id=0x84ce, seq=3/768, ttl=1 (no response found!)
17.610866	172.19.125.29	104.26.4.16	IPv4	1514	0x84d2 (34002)	Fragmented IP protocol (proto=ICMP 1, off=0, ID=84d2) [Reassembled in #53]
17.611030	172.19.125.29	104.26.4.16	IPv4	1514	0x84d2 (34002)	Fragmented IP protocol (proto=ICMP 1, off=1480, ID=84d2) [Reassembled in #53]
17.611094	172.19.125.29	104.26.4.16	IPv4	1514	0x84d2 (34002)	Fragmented IP protocol (proto=ICMP 1, off=2960, ID=84d2) [Reassembled in #53]
17.611116	172.19.125.29	104.26.4.16	ICMP	903	0x84d2 (34002)	Echo (ping) request id=0x84ce, seq=4/1024, ttl=1 (no response found!)
22.613067	172.19.125.29	104.26.4.16	IPv4	1514	0x84d3 (34003)	Fragmented IP protocol (proto=ICMP 1, off=0, ID=84d3) [Reassembled in #61]
22.613143	172.19.125.29	104.26.4.16	IPv4	1514	0x84d3 (34003)	Fragmented IP protocol (proto=ICMP 1, off=1480, ID=84d3) [Reassembled in #61]
22.613146	172.19.125.29	104.26.4.16	IPv4	1514	0x84d3 (34003)	Fragmented IP protocol (proto=ICMP 1, off=2960, ID=84d3) [Reassembled in #61]
22.613149	172.19.125.29	104.26.4.16	ICMP	903	0x84d3 (34003)	Echo (ping) request id=0x84ce, seq=5/1280, ttl=1 (no response found!)
27.618214	172.19.125.29	104.26.4.16	IPv4	1514	0x84d4 (34004)	Fragmented IP protocol (proto=ICMP 1, off=0, ID=84d4) [Reassembled in #66]
27.618350	172.19.125.29	104.26.4.16	IPv4	1514	0x84d4 (34004)	Fragmented IP protocol (proto=ICMP 1, off=1480, ID=84d4) [Reassembled in #66]
27.618355	172.19.125.29	104.26.4.16	IPv4	1514	0x84d4 (34004)	Fragmented IP protocol (proto=ICMP 1, off=2960, ID=84d4) [Reassembled in #66]
27.618359	172.19.125.29	104.26.4.16	ICMP	903	0x84d4 (34004)	Echo (ping) request id=0x84ce, seq=6/1536, ttl=2 (no response found!)
27.629926	192.168.41.41	172.19.125.29	ICMP	590	0x54b5 (21685), 0x84d4 (34004)	Time-to-live exceeded (Time to live exceeded in transit)
27.632003	172.19.125.29	104.26.4.16	IPv4	1514	0x84d5 (34005)	Fragmented IP protocol (proto=ICMP 1, off=0, ID=84d5) [Reassembled in #71]
27.632048	172.19.125.29	104.26.4.16	IPv4	1514	0x84d5 (34005)	Fragmented IP protocol (proto=ICMP 1, off=1480, ID=84d5) [Reassembled in #71]
27.632111	172.19.125.29	104.26.4.16	IPv4	1514	0x84d5 (34005)	Fragmented IP protocol (proto=ICMP 1, off=2960, ID=84d5) [Reassembled in #71]
27.632129	172.19.125.29	104.26.4.16	ICMP	903	0x84d5 (34005)	Echo (ping) request id=0x84ce, seq=7/1792, ttl=2 (no response found!)
27.637993	192.168.41.41	172.19.125.29	ICMP	590	0x54b6 (21686), 0x84d5 (34005)	Time-to-live exceeded (Time to live exceeded in transit)
27.638133	172.19.125.29	104.26.4.16	IPv4	1514	0x84d6 (34006)	Fragmented IP protocol (proto=ICMP 1, off=0, ID=84d6) [Reassembled in #76]
27.638164	172.19.125.29	104.26.4.16	IPv4	1514	0x84d6 (34006)	Fragmented IP protocol (proto=ICMP 1, off=1480, ID=84d6) [Reassembled in #76]
27.638189	172.19.125.29	104.26.4.16	IPv4	1514	0x84d6 (34006)	Fragmented IP protocol (proto=ICMP 1, off=2960, ID=84d6) [Reassembled in #76]
27.638206	172.19.125.29	104.26.4.16	ICMP	903	0x84d6 (34006)	Echo (ping) request id=0x84ce, seq=8/2048, ttl=2 (no response found!)
27.642383	192.168.41.41	172.19.125.29	ICMP	590	0x54b7 (21687), 0x84d6 (34006)	Time-to-live exceeded (Time to live exceeded in transit)

## PLAGIARISM STATEMENT

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 plagiarized 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 honor violations by other students if I become aware of it.

Name of the student : **SHRUSTI**

Roll No : **CS22MTECH11017**