Name: *Muhammad Rizwan Khalid*
Registration No: *180459*
IP address: *10.7.44.166*

```
Wireless LAN adapter Wi-Fi:

   Connection-specific DNS Suffix  . : nust.edu.pk
   Link-local IPv6 Address . . . . . : fe80::41ab:dcf1:1dd2:f4b2%13
   IPv4 Address. . . . . . . . . . . : 10.7.44.166
   Subnet Mask . . . . . . . . . . . : 255.255.252.0
   Default Gateway . . . . . . . . . : 10.7.44.1
```

Lab Title: Introduction to the Network Layer

1.0 Objective of this lab:

In this lab, we'll investigate the IP protocol, focusing on the IP datagram. We'll do so by analyzing a trace of IP datagrams sent and received by the execution of the traceroute program. We'll investigate the various fields in the IP datagram, and study IP fragmentation in detail.

2.0 Instructions:

- *Read carefully before starting the lab.*
- *These exercises are to be done individually.*
- *You are supposed to provide the answers to the questions listed in this document (attach screen shots) and upload the completed report to your course's LMS site.*
- *Avoid plagiarism by copying from the Internet or from your peers. You may refer to source/ text but you must paraphrase the original work. Your submitted work should be written by yourself.*

4. IP fragmentation

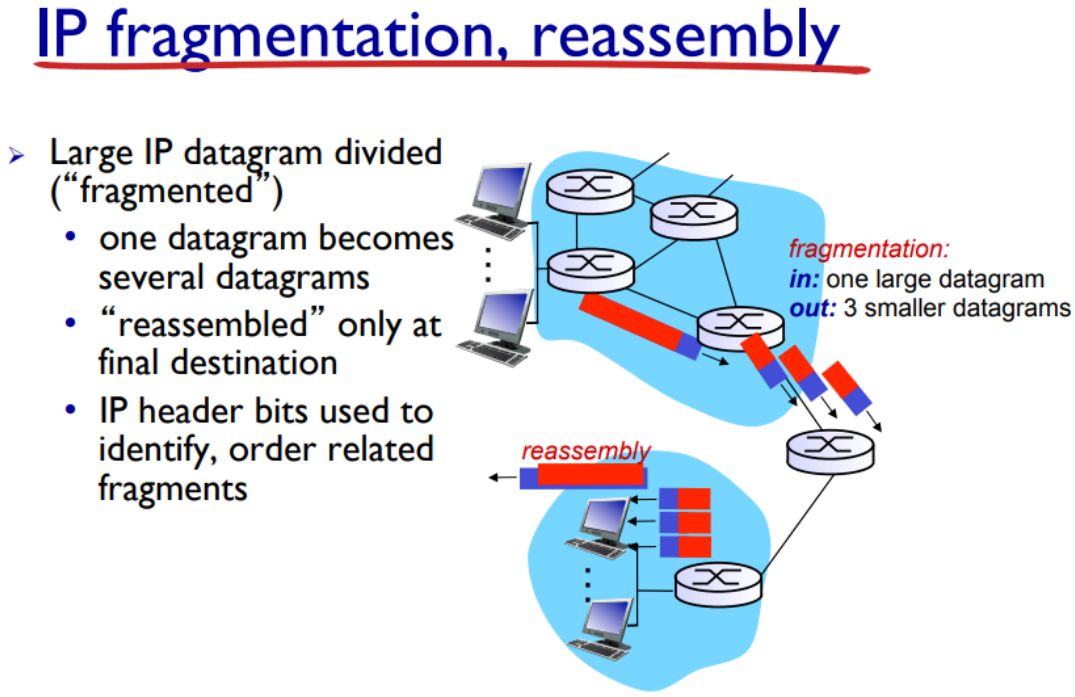
4.1 Background:

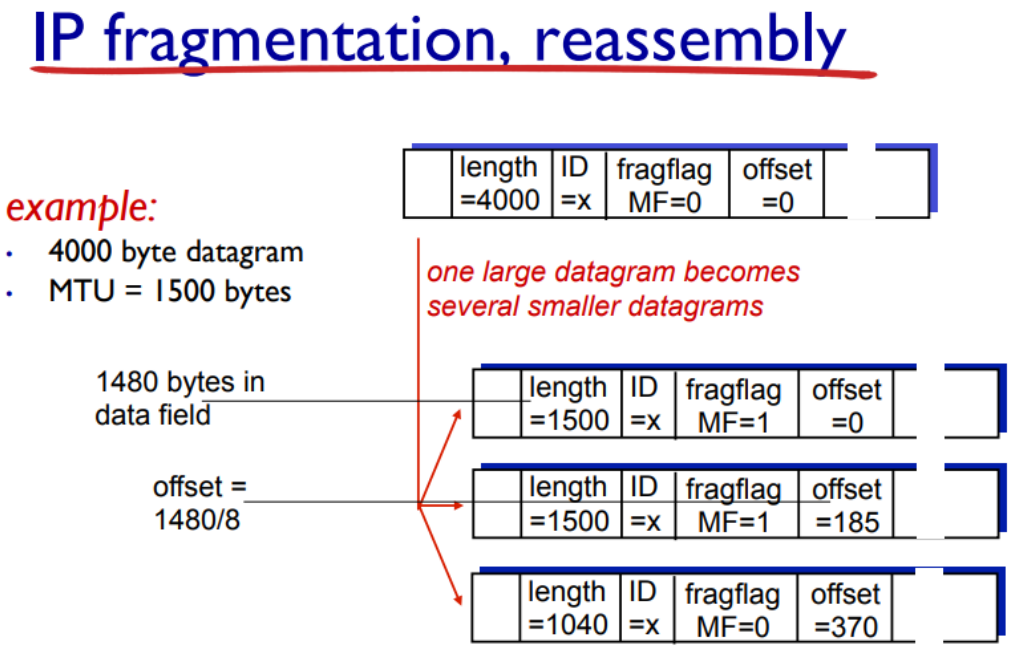
IP fragmentation is an Internet Protocol (IP) process that breaks datagrams into smaller pieces (fragments), so that packets may be formed that can pass through a link with a smaller maximum transmission unit (MTU) than the original datagram size. The fragments are reassembled by the receiving host[1].

A maximum transmission unit (MTU) is the largest size packet or frame, specified in octets (eight-bit bytes), that can be sent in a packet- or frame-based network such as the Internet. The Internet's Transmission Control Protocol (TCP) uses the MTU to determine the maximum size of each packet in any transmission. Too large an MTU size may mean retransmissions if the packet encounters a router that can't handle that large a packet. Too small an MTU size means relatively more header overhead and more acknowledgements that have to be sent and handled. Most computer operating systems provide a default MTU value that is suitable for most users. In general,

Internet users should follow the advice of their Internet service provider (ISP) about whether to change the default value and what to change it to[2].

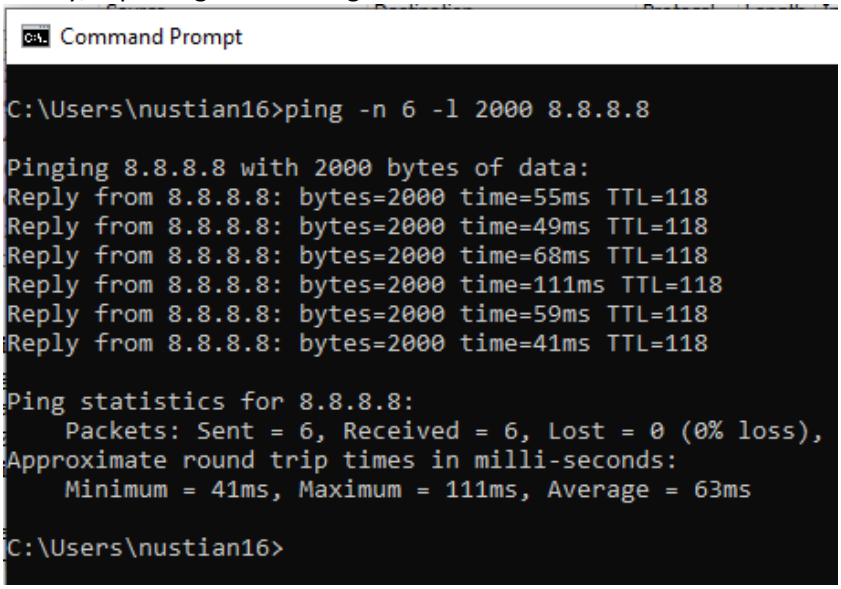
In this part of the lab, we will try to find out what happens when IP fragments a datagram by increasing the size of a datagram until fragmentation occurs. We will ping with options (use the –s option on MacOS, -l on windows) to set the size of data to be carried in the ICMP echo request message. Note that the default packet size is 64 bytes in MacOS and 32 Bytes in Windows. Once you have send a series of packets with the increasing data sizes, IP will start fragmenting packets that it cannot handle.

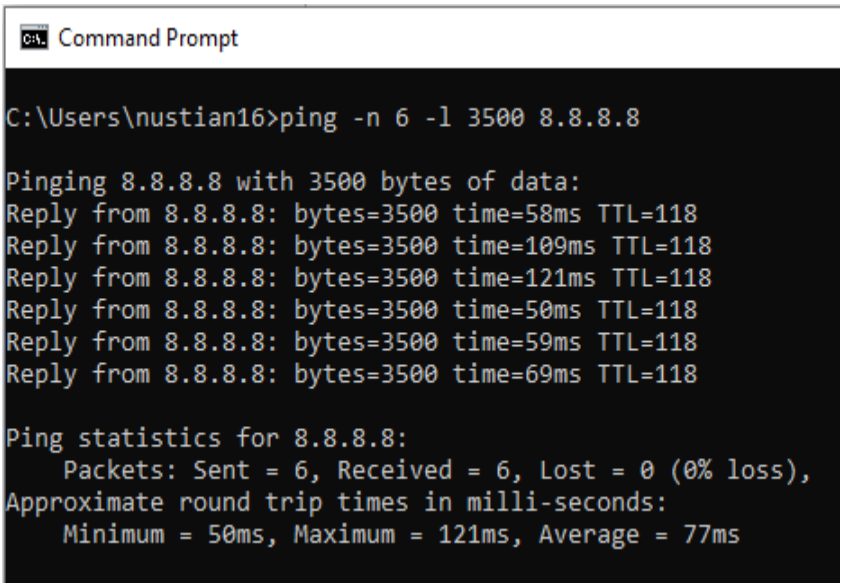




4.2 Steps for performing this part of the lab:

- Startup Wireshark and begin packet capture (Capture->Start) and then press OK on the Wireshark Packet Capture Options screen (we'll not need to select any options here).
- Startup ping with default values to the target destination as 8.8.8.8
- Stop the wireshark capture when you receive at least 6 replies from target destination.
- Next, repeat the above steps by sending a set of ICMP requests with a length of 2000.
- Finally, repeat again with length value set as 3500





Now answer the following questions.

1. **Which datagram size has caused fragmentation and why? Which host/router has fragmented the original datagram? How many fragments have been created in this case?**

   In the two cases, fragments will be framed in light of the fact that in Ethernet/Wi-fi size of the packet which can be sent on the connection is constrained to 1500 bytes. As 1500 is

smaller than 2000 and 3500 which I am attempting to send that is the reason packet get fragmented. In every packet, we need to duplicate a header of a primary bundle before it gets divided. These packets are divided at the host machine as a machine is sending the packet as indicated by some convention characterized by the OS. If there should arise an occurrence of a packet of 2000 bytes 2 pieces will be fragmented whereas in the event of 3500 bytes 3 sections will be framed.

For packet of size 2000 bytes:

```
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 548
    Identification: 0xbdc6 (48582)
  > Flags: 0x00b9
    Time to live: 128
    Protocol: ICMP (1)
    Header checksum: 0x339d [validation disabled]
    [Header checksum status: Unverified]
    Source: 10.7.44.166
    Destination: 8.8.8.8
  v [2 IPv4 Fragments (2008 bytes): #163(1480), #164(528)]
      [Frame: 163, payload: 0-1479 (1480 bytes)]
      [Frame: 164, payload: 1480-2007 (528 bytes)]
      [Fragment count: 2]
      [Reassembled IPv4 length: 2008]
      [Reassembled IPv4 data: 08007b6400010013616263646566768696a6b6c6d6e6f70...]
```

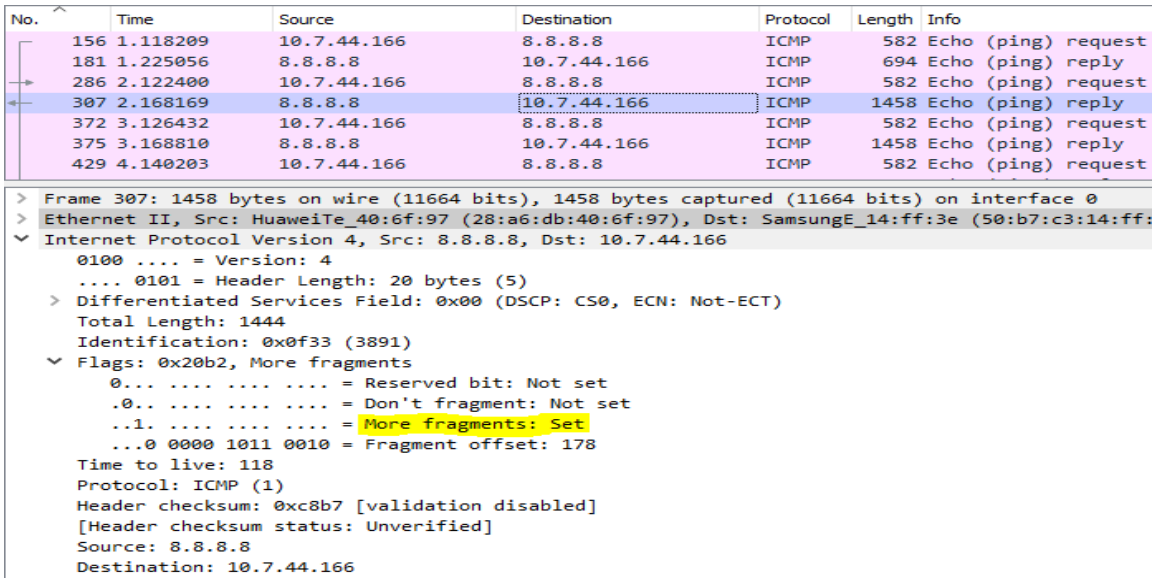
For packet of size 3500 bytes

```
> Frame 156: 582 bytes on wire (4656 bits), 582 bytes captured (4656 bits) c
> Ethernet II, Src: SamsungE_14:ff:3e (50:b7:c3:14:ff:3e), Dst: HuaweiTe_40:
v Internet Protocol Version 4, Src: 10.7.44.166, Dst: 8.8.8.8
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 568
    Identification: 0xbe6f (48751)
  > Flags: 0x0172
    Time to live: 128
    Protocol: ICMP (1)
    Header checksum: 0x3227 [validation disabled]
    [Header checksum status: Unverified]
    Source: 10.7.44.166
    Destination: 8.8.8.8
  v [3 IPv4 Fragments (3508 bytes): #154(1480), #155(1480), #156(548)]
      [Frame: 154, payload: 0-1479 (1480 bytes)]
      [Frame: 155, payload: 1480-2959 (1480 bytes)]
      [Frame: 156, payload: 2960-3507 (548 bytes)]
```

2. **Did the reply from the destination 8.8.8.8. For 3500 byte datagram also get fragmented? Why and why not?**

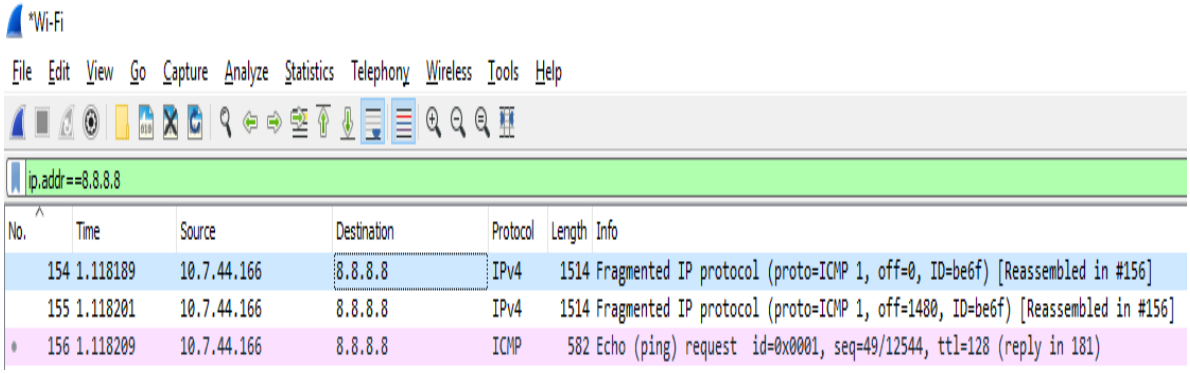
Answer:

3500-byte datagram gets divided in light of the fact that the length of a packet that can be sent on the wifi is restricted to 1500 bytes. As here I am sending a packet of 3500 bytes which is more prominent than 1500 that is the reason packet get divided. In every packet, we need to duplicate a header of a primary packet before it gets divided. Discontinuity is done as such that information can come to the destination with no misfortune even the connection limit is less.



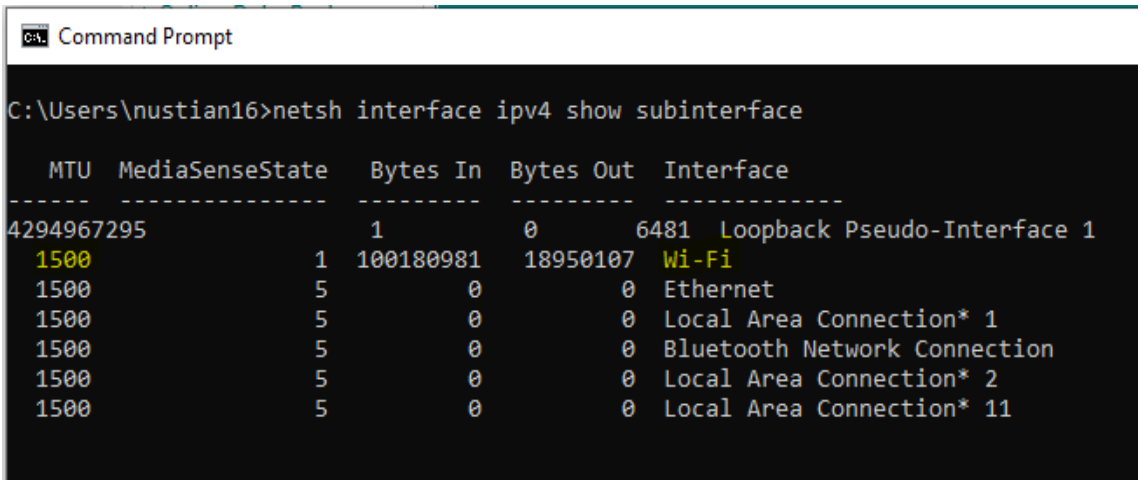
3. **Give the ID, length, MF and offset values for all the fragments of any single datagram with size 3500 bytes?**

ID	Length	MF	Offset
Be6f	**1514**	**1**	**0**
Be6f	**1514**	**1**	**1480**
Be6f	**582**	**0**	**2960**



4. **Has fragmentation of fragments occurred when datagram of size 3500 has been used? Why and why not?**
   Fragmentation of fragments is not occurred since the connection set up to the IP has the least limit at the Ethernet switch i.e., 1500 and in the path to destination there is no switch with a limit not as much as that.

5. **What is the least MTU value along the complete path to destination? Which is the bottleneck link?**
   The minimum value of MTU along the total path to a destination is 1500 bytes in light of the fact that my PC is associated with Wi-fi and its MTU is 1500. Physical layer is the bottleneck connect as MTU is characterized by physical layer.



Conclusion:

Basically in this lab we practiced and observed the concept of ip fragmentation. Wifi, in my case can send the packet of size upto 1500 bytes. When we try to send the packet of size greater than 1500 bytes the packets get fragmented. We used wireshark to capture ICMP packets. We send the 2000 bytes and 3500 bytes packets. In case of 2000 bytes packets, there were 2 fragments, while in case of 3500 bytes packets there were 3 fragments.

References

[1] https://en.wikipedia.org/wiki/IP_fragmentation
[2] https://searchnetworking.techtarget.com/definition/maximum-transmission-unit