180041120
Md Farhan Ishmam

Ans. to Q no. 1

We know ~~header of~~ an IP datagram has the following format

Version	HLen.	Service Type	Total length
4 bits	4 bits	8 bits	16 bits

Identification	Flag	Fragment Offset
16 bits	3 bits	13 bits

TTL	Protocol	Checksum
8 bits	8 bits	16 bits

Source IP
32 Bits

Destination IP
32 bit

Option + Paddling
(0-40) bytes

Data

Given,

	HLEN		Total Length	
4 [5]	00	[00	54)	
00	03	00	00	
TTL [20]	06	[00	00]	Checksum
7C	4E	03	02	
B4	0E	0F	02	

a) No, there are no options.

b) No, packet is not fragmented

c) The size of data =

$$\text{total length} - \text{header length}$$

header length is $(5 \times \frac{8}{4}) = \cancel{40} 20$ bytes

total length $(54)_{16} = 84$ bytes

So, size of data $= 84 - 20 = 64$ bytes.

d) No, checksum is not used.

e) The router can travel $(20)_{16} = 32$ more routers.

f) Identification number is $(3)_{16} = 3$ (Ans.)

g) Sevice type is 0.

It means normal service type.

Ans. to Q. no. 2

The mutable part of header is the [option + padding] field. This part is mutable and can be from 0 to 40 bytes. It is used for testing and debugging.

Fragmentation and Reassembly: When ~~a packet~~ a packet is too long i.e. data is too big then the packet is broken to smaller fragments by taking the same header, changing fragment offset and more fragment bit and then dividing the data. When the packet is received all fragments are reassembled.

Let's say, the maximum transferable unit, MTU is 1400 bytes
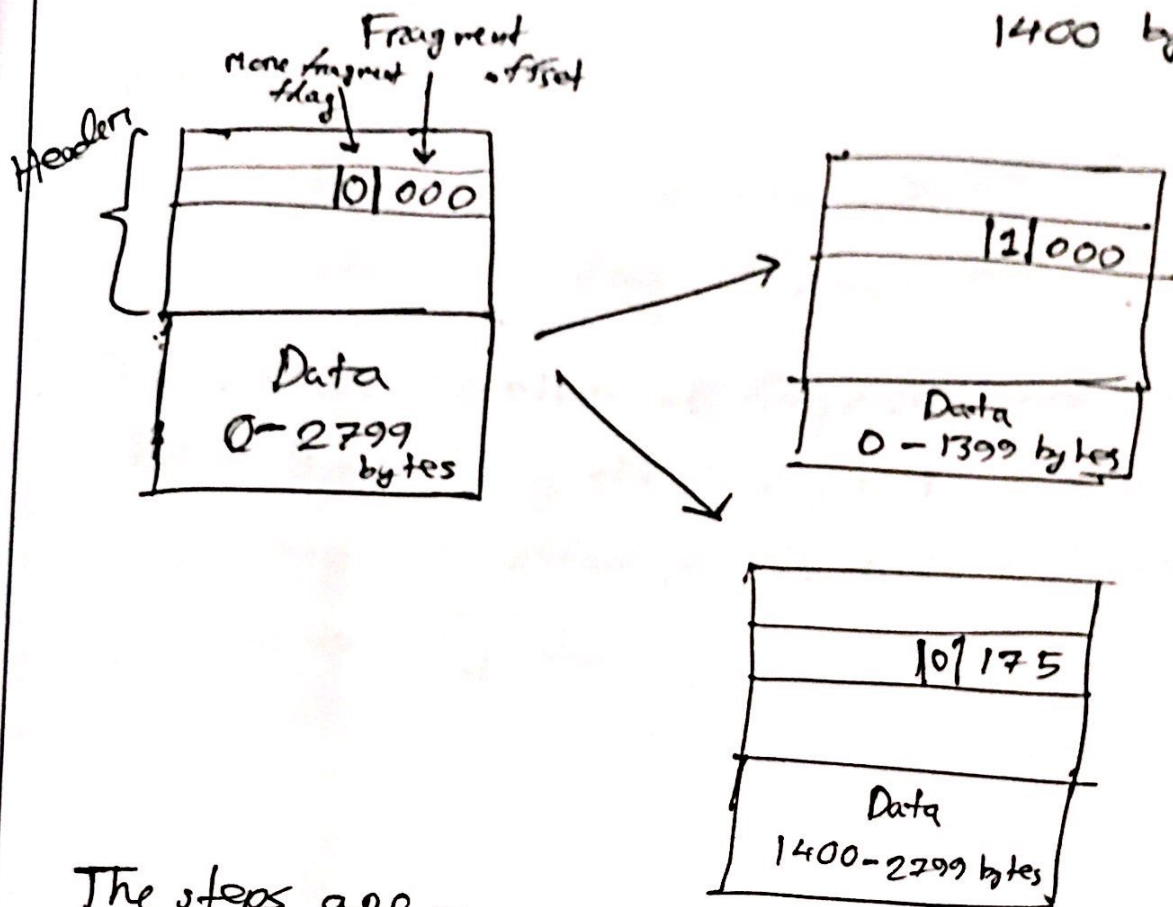
The steps are —

i) First the header is copied for fragmented framers and for all fragments except last one, more flag is set to 1.

ii) The length of the fragment is divided by 8 and is set in fragment offset of next fragment(s). Here 1400/8 = 175 which is value in the second packet's fragment offset.

iii) Data is divided between the two fragments.

Fragment one has 1400 bytes which is equal to the one MTU.

iv) Continueing this process for all fragments, we set more flag of last fragment to 0.

The receiver receives all fragments and combine them by adding the data portions using the fragment offset till it finds last fragment with more flag 1.