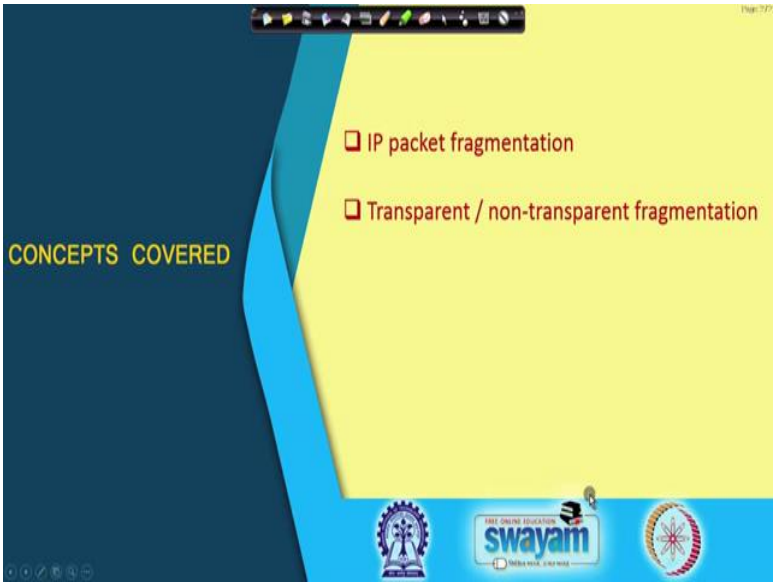
Ethical Hacking
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 06
IP Addressing and Routing (Part I)

So, let us continue with our discussion. In this lecture we shall be talking about IP Addressing and Routing, because as I mentioned that in the Internet TCP/IP plays a very important role and it is the IP layer which is at the network layer, which is responsible for all the routing of the packets and some addressing issues. So, the title of the lecture is IP Addressing and Routing part I.

(Refer Slide Time: 00:49)

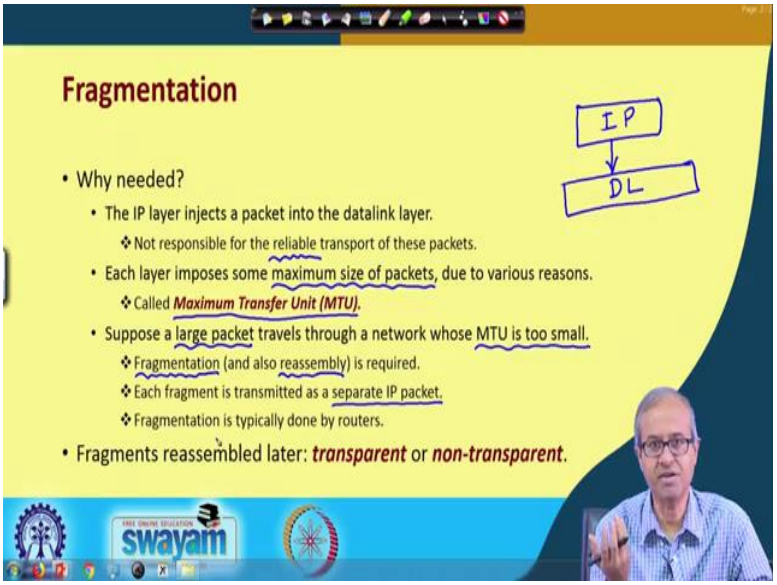


Now, in this lecture I shall be broadly talking about IP packet fragmentation; IP addressing I will be talking about a little later. Firstly, we will be talking about IP packet fragmentation and there are broadly two types of fragmentation which is there, transparent and non transparent; we shall be looking at those.

This fragmentation comes from the point that I had mentioned earlier, that when a packet is being given to the IP layer for routing, the packet may be too large, there may be some networks where the IP software or the layer is so configured that the maximum size of a packet is limited. So, if the incoming packet is larger than that, then the packet may have

to be broken up or fragmented into smaller packets ok. This is the basic idea behind fragmentation.

(Refer Slide Time: 01:49)



So, let us come to this fragmentation as I said, we require because we need to break up a packet at times. Now, the way it works if you look at the layering, you have the IP layer sitting at the network level and below the IP layer you have the data link or the physical layer or the hardware layer whatever you call, let us call it the data link layer. So, whenever the IP layer wants to send some packet for delivery, it will first be giving that packet to the data link layer ok.

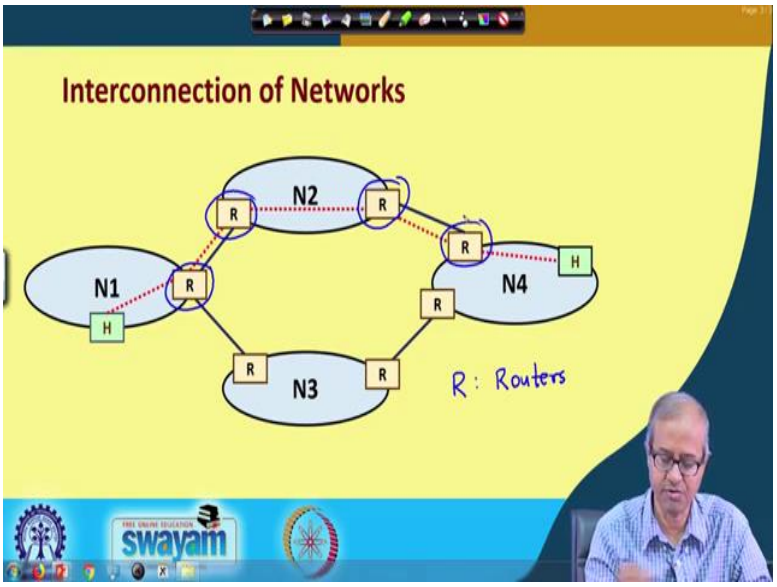
So, the IP layer simply injects a packet into the data link layer and IP does not consider at all regarding reliability. It is a basic datagram service where each of the packets are being sent out as independent entities and as I mentioned earlier, the packet might get lost. Duplicate packets may get generated and the ordering of the packets is also not maintained. There is no guarantee right.

Now, the point that I was mentioning, that at the layer of IP; in fact, at every layer there is a maximum size of the data unit that can be handled, that is primarily because of the size of the buffers, the number of bits that is reserved in the packet format and so on ok. And this limit or limitation is referred to as Maximum Transfer Unit or MTU; the MTU can vary from one network to another. So, when a packet traverses through multiple networks, it may encounter various different MTU values alright.

So, as I had said, suppose I have a large packet and it is trying to flow through a network, but MTU is rather small. So, what will happen? There will be a process called fragmentation which will take place, but the packet will be broken up into smaller packets and somewhere later the smaller packets will have to be put together again to get back the original packet, that part is called reassembly; so, fragmentation and reassembly.

And each of these fragments that are created they are treated as separate IP packets and are transmitted separately and this fragmentation is typically done by the routers when it receives a packet from some other network and wants to route it inside the network or maybe to the next network right. This fragmentation broadly when you are doing fragmentation and reassembly, the process can be transparent or non-transparent, this we shall be seeing in some detail.

(Refer Slide Time: 04:43)



Now, this will be our network model; we will be assuming that we have a collection of networks, these N1, N2, N3, N4, these are different networks and you can see this small R's, this R's are nothing but routers. This R's refer to routers; routers are essentially, they are networking devices which operate at the IP layer level ok. So, whatever we are talking about regarding IP fragmentation, it will be taken care off inside that router ok.

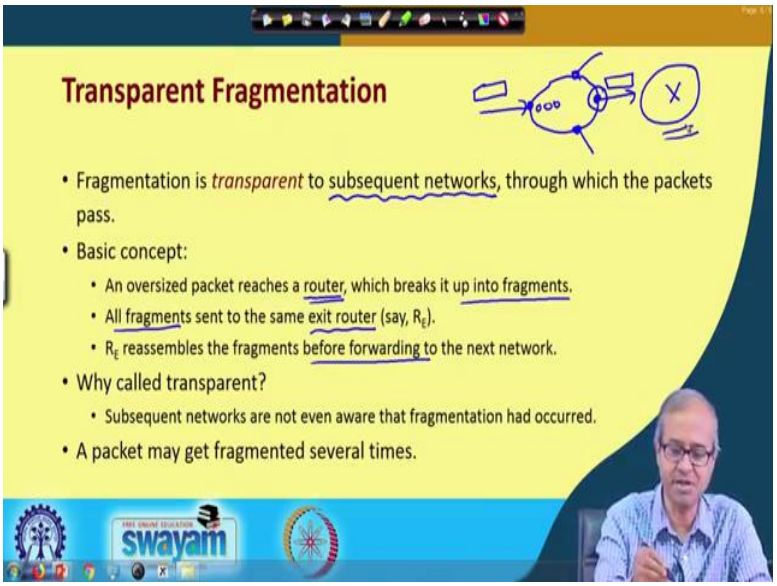
So, you see there are routers at the boundary of the different networks and the different networks are typically connected through these routers. For example, N1 and N2 are

connected via these two routers, N2 and N2 and N4 are connected by these two routers and so on right.

Now, suppose there are also some host, some computers connected to the network, let us say I have one computer, a host H here and there is another host here, suppose I want to transmit some data packet from this host to this host in N4. So, there can be multiple paths that the packet can take. So, I am showing one of the path shown by this red dotted line where it will first be coming to this R, then this router to the N2, through N2 it will cross, then it will entered this and finally, it will reach the final network.

So, the idea that I was talking about may be N1 and N4 is able to handle large packets, large MTU, but N2 is an intermediate network through which the packet is flowing. So, if N2 does not allow large enough packets, then there may be fragmentation and that fragmentation will typically be carried out by this router. Now, this packet whenever it enters N2, this router will be fragmenting the packet into multiple smaller packets and then it will be forwarding ok, this is the basic idea.

(Refer Slide Time: 07:01)



So, let us first look at transparent fragmentation; transparent means networks does not understand that fragmentation has taken place, it is transparent let us say. The idea here, whenever there is fragmentation, this subsequent network does not realize that fragmentation has taken place. So, each of the fragments when they come, they are treated

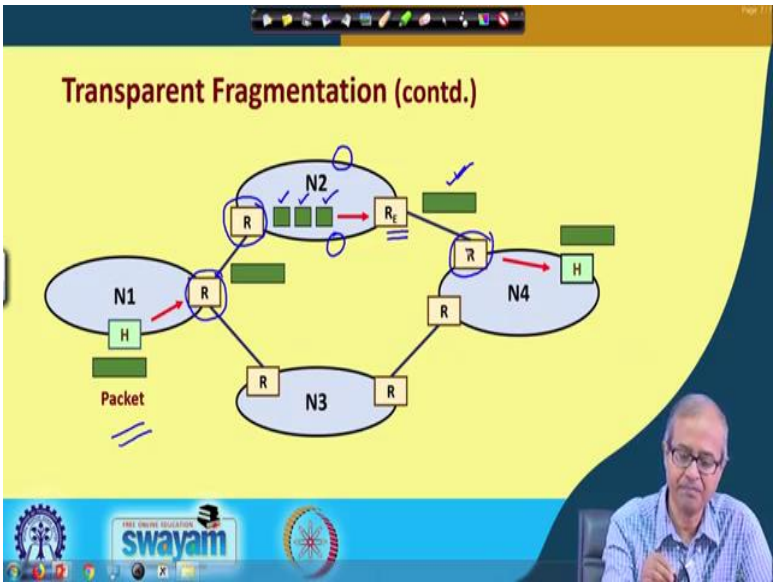
as independent IP packets and they are routed separately. So, these intermediate networks are not at all worried about the reassembly part, only it is doing the fragmentation ok.

But, you see the idea is something like this, suppose I have a network here, a packet is coming, it is getting fragmented, it is coming out. So, the subsequent networks does not need to know that a fragmentation has taken place. The idea is that if there is fragmentation, then their exit router in the same network will be responsible for reassembly, it will again put together the fragments back. So, that the original packet which was there, here it will again come out of this network. So, fragmentation and reassembly happens inside this network and it is totally transparent to the other networks right.

So, the idea it follows is that say, suppose a large packet reaches a router which breaks it up into smaller packets or fragments and each of these fragments are routed and there is a constraint here, they must be sent to the same exit router; let us say R_E, because in a particular network, there can be a multiple routers connected, there can be a multiple routers and connection to the outside world.

So, all the fragments must be sent to the same exit router, so that the exit router should be able to do the reassembly, put the fragments back together again ok. This R_E will reassemble the fragments into the original packet before it can forward to the next network, this is the basic idea, this is called transparent, because these subsequent networks are not aware of the fact that fragmentation is taking place. Now, in general because a packet may be traversed in multiple networks on its way to the destination, this kind of fragmentation and reassembly may happen multiple times. So, this is something you also need to remember.

(Refer Slide Time: 09:51)

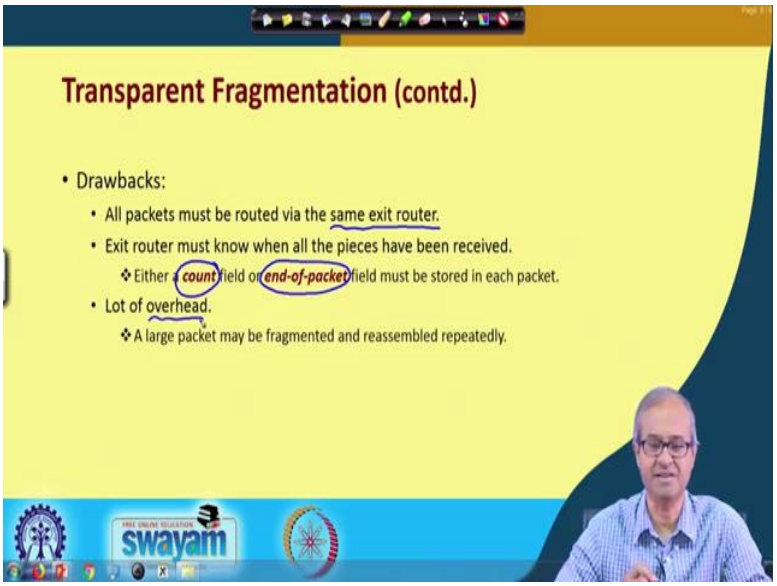


So, talking about transparent fragmentation, I am illustrating this here with the help of this simple example. Suppose I have a packet here, a larger packet which is being generated by this host in network N1, sent to this particular router, this router forwards it to this router of network N2. So, this original packet reaches R.

Now, R sees that this packet is too large and the MTU of N2 is smaller. So, what it does, it breaks up the packet into three smaller packets, these are the fragments and they are sent to the same exit router, there can be multiple other routers also, here I have not shown, I am showing only one router.

They are sent to the same exit router which will be again reassembling them back to the original packet and that packet will now reach network N4. Now this particular router and network N4 will not be aware of the fact that there was a fragmentation earlier, because what it receives is the original packet, this is transparent fragmentation.

(Refer Slide Time: 11:05)



Now here one of the drawbacks for transparent fragmentation is, here we are saying that all packets must go through the same exit router, because the exit router will have to reassemble the fragments, but the problem is even if there are multiple exit routers all the fragments will have to go to the same router for handling. So, that router might get overloaded. So, if there was a scope for parallel processing using other routers, maybe the forwarding of the packets would have been faster, but here because we are using the same exit router, the burden on that router increases, it may become slow ok.

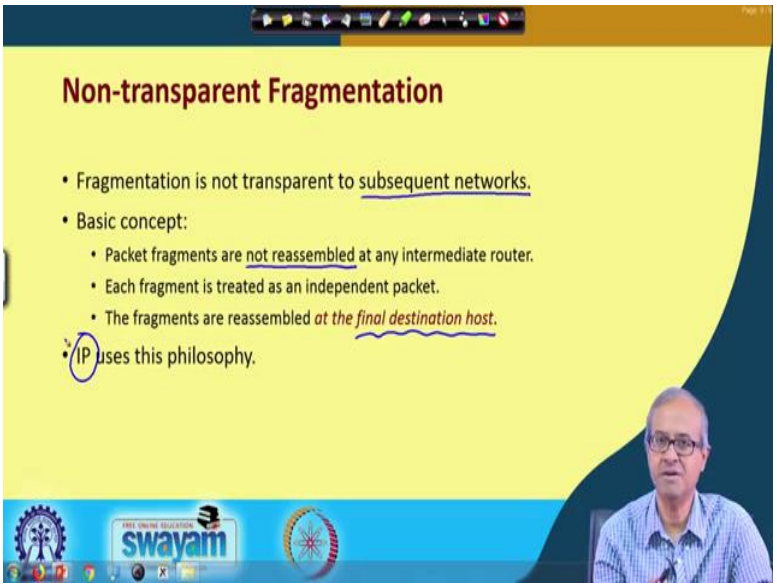
Now, another thing is that, talking about the exit router; exit router must be able to know that whether it had received all the fragments of a packet, so that it can put together all the fragments again. How it can know? There are two alternatives you can think off?

Now, each packet can come with a count field; count field will indicate how many remaining fragments are there. So, when it reaches the exit router, exit router looks at the count field in the packet. So, when the count field reaches 0 let us say, so, it will know that all fragments have come, now I can reassembled them or there can be a special delimiter like end of packet. The last fragment of the packet will be marked specially so that the exit router will know that this is the last fragment, there are no more fragments after this. So, now, I can assemble fine.

Now, now here as I said here, lot of overheads are encountered, there are two things, one is we are putting more burden on one particular router. And secondly, after fragmentation

the exit router must be doing a reassembly. So, if this packet flows through multiple networks, there will be multiple possible fragmentations multiple reassembly again fragmentation again reassembly. So, the amount of overhead is much higher here.

(Refer Slide Time: 13:25)

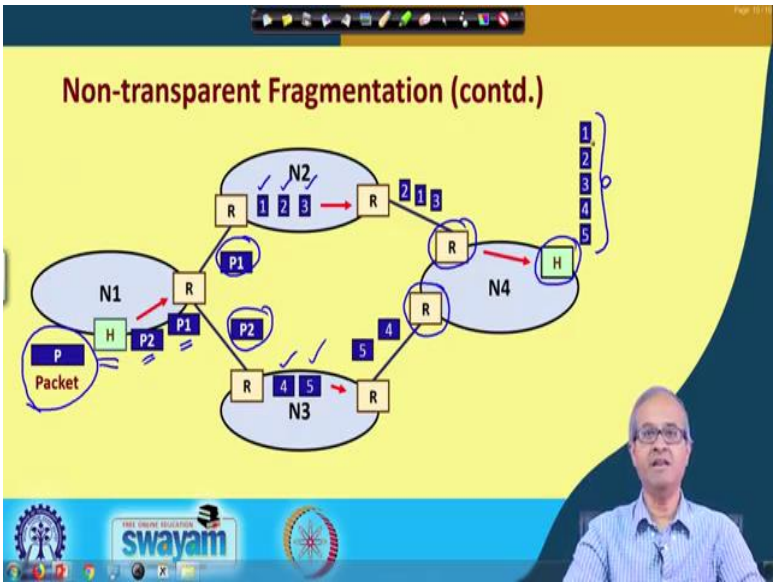


Now, coming to non-transparent fragmentation, this is what the IP protocol follows. Here the fragmentation is not transparent to the subsequent networks, the subsequent network will know that fragmentation has taken place. Suppose, a router divides a packet into four fragments; so all those four fragments will go to the next network as separate packets. So, the subsequent networks will be receiving all these four smaller packets as independent separate packets ok. So, there is no reassembly that is done in the intermediate stages.

So, the basic concept is that the fragments are not reassembled at any of the intermediate routers, fragments are created and let them flow to the destination through the intermediate networks or routers wherever they are. Each fragment is treated as an independent packet as it said and the responsibility of the reassembly in this fragment lies with the final destination host.

Suppose, I am sending the packet to your computer finally, all the fragments will come to the IP layer of your computer and that is the IP layer of your computer where these fragments will get assembled again ok, this is how non transparent fragmentation works and as it said, the IP protocol in TCP/IP uses or follows non transparent fragmentation.

(Refer Slide Time: 15:01)

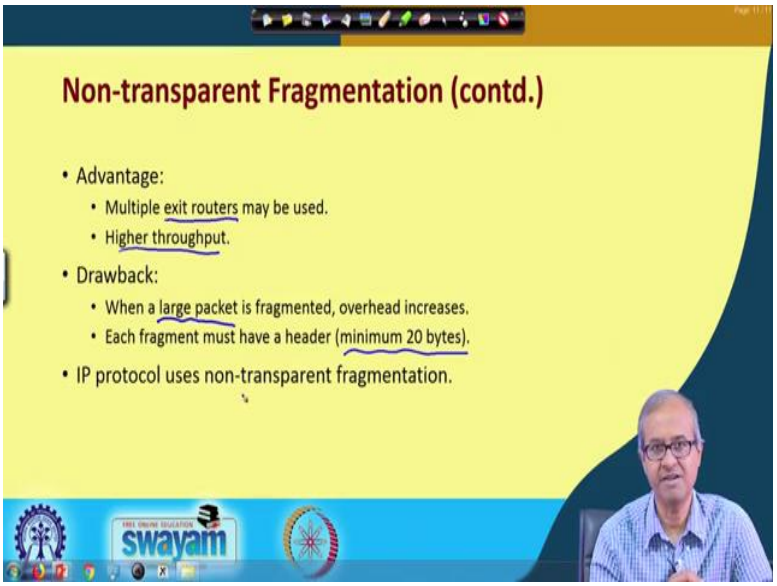


Now, non-transparent fragmentation, let me again also illustrate with an example. Suppose this particular host in network N1 is generating a packet here, which has to be transmitted. Let us say this packet is large enough. So, this is broken up into two fragments P1 and P2, it depends on the MTU of N1; N1 is small. So, it has to be broken into two fragments. Now, suppose datagram can flow through any path; let us say P1 goes through N2 and P2 goes through this network N3 it may so happen.

Now, when P1 reaches N2 maybe the MTU of N2 is even smaller. So, this P1 is broken up into three smaller fragments, let us say 1, 2, 3 and on the other hand P2 is broken up, let us say into two fragments 4 and 5 ok. So, this 4 and 5 will be forwarded to this network ,this N4 via some router and 1, 2, 3 will be forwarded to network N4 via some other router. Now, as it said IP does not guarantee order of delivery of the packets. So, maybe 1, 2, 3 will be received in some other order 3, 1, 2 or 2, 1, 3 in that order and may be here in the order of 5 4.

Finally, all these five smaller packets or fragments will reach the destination host and destination host will be looking at some of the fields in the header and will be putting together the fragments in proper order and it will generate the original packet. This is how non-transparent fragmentation works. Fragmentation can be taken, can be done by the different routers, assembly or reassembly will be done only at the final destination host, this is the basic idea.

(Refer Slide Time: 17:09)

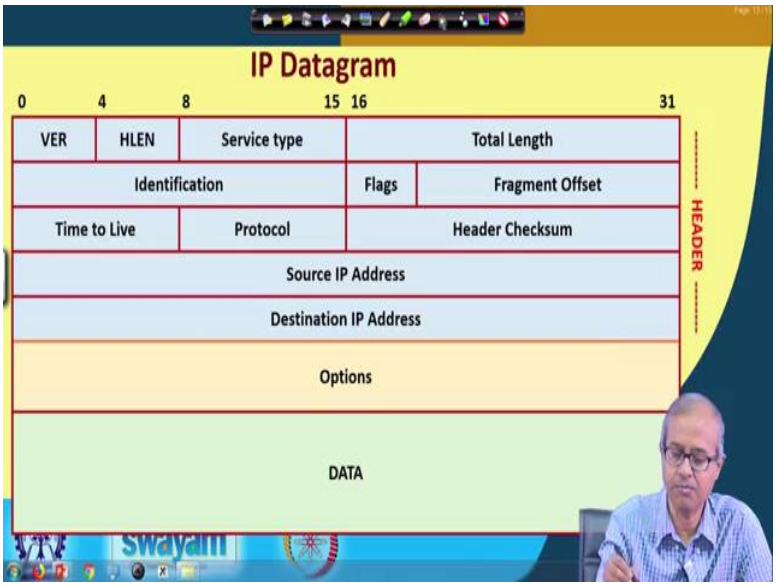


The advantages you can clearly see, here you can use multiple exit routers for transmitting the fragments and therefore, you can have better utilization of the networking resources the links, this will result in higher throughput quite naturally because you are using multiple paths together.

But drawback is that as the previous example shows, so, if the original packet is large, there will be multiple fragments. In the previous example, there are five fragments created and each of these five fragments are IP packets. So, there will be about 20 bytes of header ok. So, this header overhead will be increasing. So, the amount of data that will be flowing through the network in number of bits will increase, that is why some kind of you can say overhead in terms of the number of bits that is being transmitted that will be a little higher in this method right.

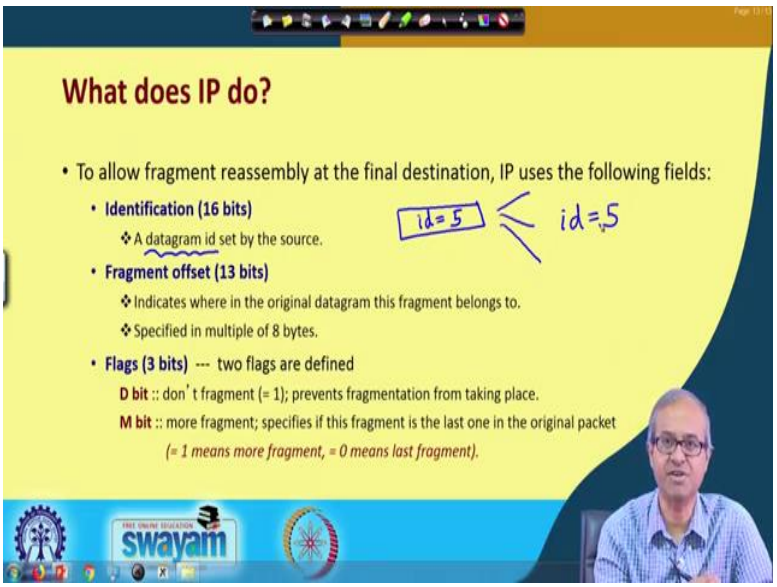
Because as I said each fragment will be having an IP header which is as you know is minimum 20 bytes and I mentioned IP protocol uses this philosophy non transparent fragmentation.

(Refer Slide Time: 18:31)



Now, let us look at the header structure of an IP packet, this already we have seen earlier, these are the different fields. Now, with respect to fragmentation and reassembly, I did not discuss three of the fields earlier, they are this identification, some flags and this fragment offset, these three fields in particular are used for handling this fragmentation and reassembly; let us see how.

(Refer Slide Time: 19:09)

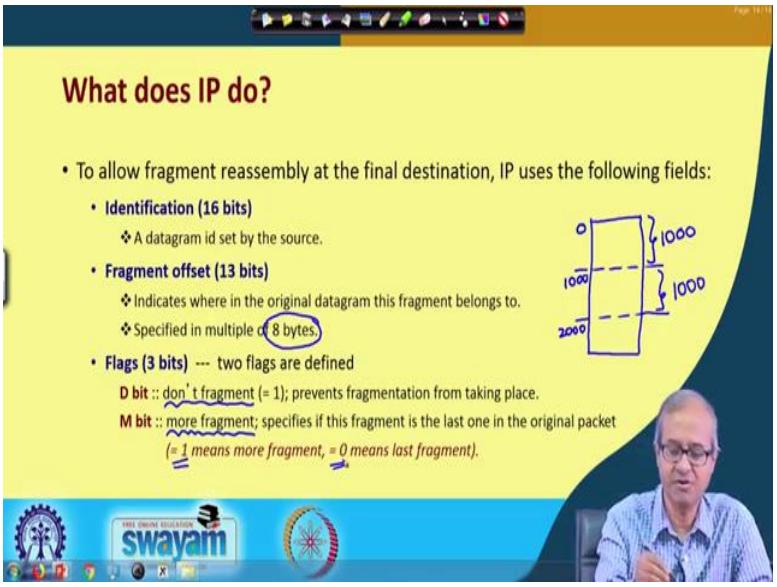


The minimum these fields are as follows, the identification is a 16 bit field, this identifies the datagram id; that means, the original packet id, the id is that if my original packet has

an id of let us say id equal to 5, then all the fragments that will be created from this fragment, they will all be having id of 5.

So, the id field indicates that although there are different fragments, different packets, but they actually belong to the same master packet, they have to be reassemble together, this is the idea behind id. It identifies that from which original packet this fragments were created. Next comes something called fragment offset, this is a 13-bit field.

(Refer Slide Time: 20:11)



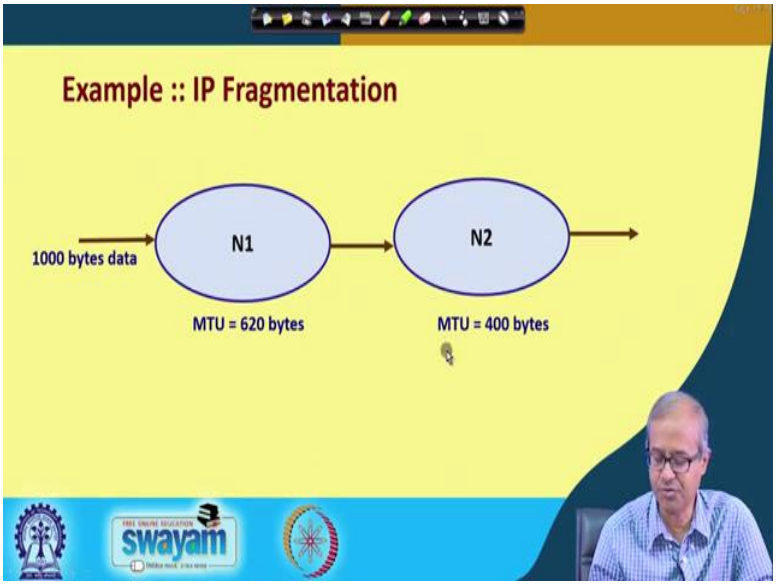
Well you see suppose your original packet was like this, let us say this packet was divided into three fragment; this is one fragment, this is one fragment, this is one fragment. So, the first fragment started from an offset 0; let us say the size of this, let us take an example, the size of this is 1000. So, the next packet will start from address 1000, this is the offset. Let us say this also has a size of 1000. So, the third packet will start from an offset of 2000.

So, when you create the fragments, you also mention the offset with respect to the original packet, so that when you are reassembling later, you will be knowing which order you will be putting this fragments back together ok. And this 13-bit field actually this specifies the offset in multiple of 8 bytes, you should remember this. So, instead of specifying this as a 16-bit field because the number of bits in the header is limited, it uses a 13-bit field and puts a restriction that offset has to be a multiple of 8 bits.

And there are flags, there are 3 bits reserved for the flags, but actually 2 flags are defined, one is a D flag, D stands for do not fragment. So, if this flag is 1, it tells the router that do not fragment this packet. So, you can enforce this that this fragment must, this packet must not be fragmented, it is carrying some other information which if you fragment there may be some problem.

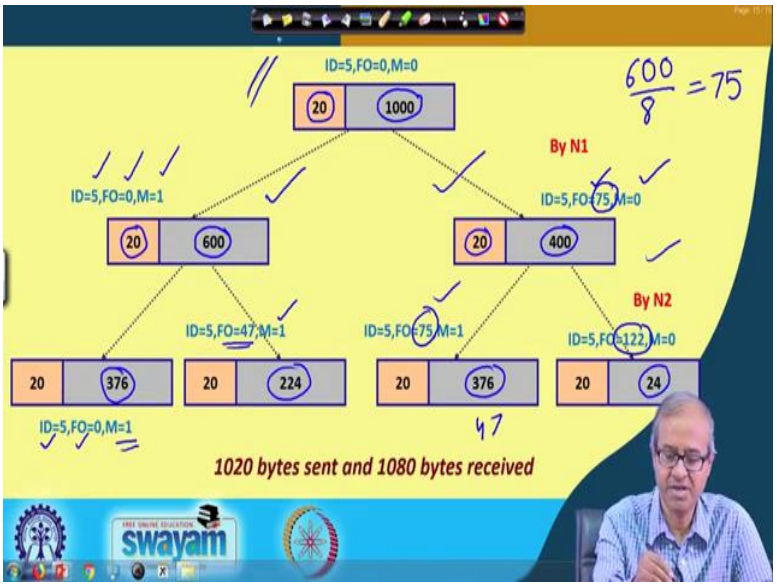
And the other field M, this stands for more; there are more fragments, this identifies whether the current fragment is the last fragment in the chain or not the last. If M equal to 1, it means there are more fragments coming after this, but if it is 0, it means that this is the last fragment of the original packet, there are no more after this ok. So, these fields are used for IP fragmentation and reassembly.

(Refer Slide Time: 22:31)



Let us take an example; let us assume that I have a scenario like this where I am trying to send, means 1000 bytes of data through IP, the first network has an MTU of 620 bytes, the second network has an MTU of 400 bytes; let us assume this.

(Refer Slide Time: 22:53)



So, in this diagram I am showing how the packets are fragmented into. This is my original packet; this is my original packet where you can see, there are 1000 bytes of data and of course, IP header will require 20 bytes, this will be the total packet. And in this header, let us say the ID is 5, let us say the ID of the packet is 5 and because there is no fragmentation, the fragment offset is 0, it starts from offset 0. M equal to 0 means there are no further fragment, this is only one single piece, single fragment packet.

Now, the first network has an MTU of 600. So, it has to be broken up, because it is 1020, this is the size of the packet. So, it will be broken up into two fragments as you can see, the first one will be having a data of 600 and there will be a header of 20. The remaining 400 will be carried by the second packet, with the header of 20. Now, you see the flags are set accordingly, ID will be the same, 5, this is the first fragment that is why the fragment offset will be 0 and $M = 1$ indicates that there are more fragments after, this is not the last.

Now, for the second one, you see there are 600 bytes before that and I mentioned that the offset is specified in multiples of 8. So, $\frac{600}{8}$, it becomes 75. So, this second packet carries a offset of 75 and it says $M = 0$ means there are no more fragments, this is the last one. Now, the third network has an even smaller MTU. So, here it is broken up into some smaller packets. Now, I leave it as an exercise for you, how the smaller packets have been broken into, you keep in mind that the number of bytes must have to be a multiple of 8,

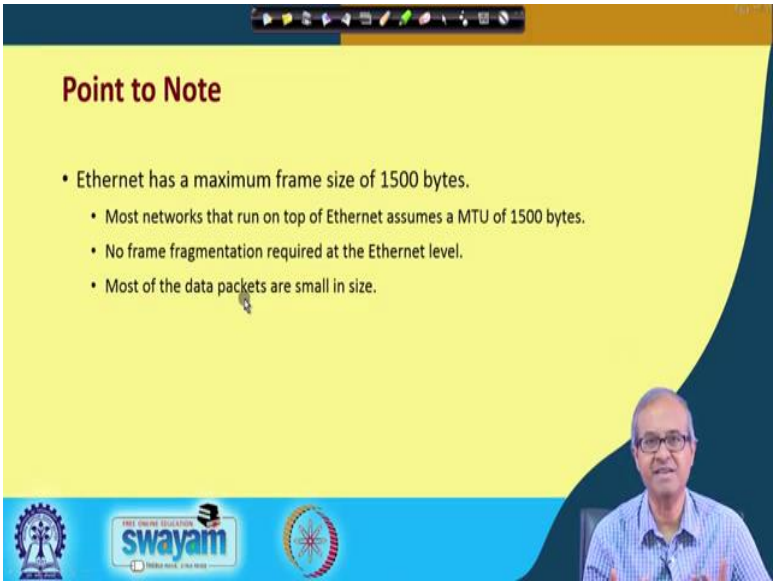
because the fragmentation offset has to be a multiple of 8. So, you cannot use something else, here it has to be a multiple of 8.

So, again for this case the ID is again 5, the first one has an offset of 0, there are more packets. The second fragment here that is generated from here, it will be have an offset of 47×8 is 376, 47 and again there are more fragments coming. And similarly for this second one, again there will be two fragments generated, the first one will be having an offset of this same 75.

More packets are then M equal to 1 and the last one will be having you see $\frac{376}{8}$ is 47 and $47 + 75$ is 122, it will be having an offset of 122 and this is the last fragment. So, this is how the packets are all generated, when a packet is fragmented ok.

So, here we have explained with the help of this example that how the packet is fragment and you see one thing that that, the original packet contained 1020 bytes because there was 20 bits of average of header, but here after fragmentation we are generating 4 fragments. So, the total amount of bits that are transmitted is 1080, this is the overhead I was talking about right.

(Refer Slide Time: 26:51)



So, the point to note is that this IP protocol that we are talking about, that here we are talking about this mostly runs on top of the Ethernet protocol in most of the networks we see. Now, in Ethernet which is at the data link layer, the maximum frame size is 1500, but

IP protocol can have a packet size of maximum 6500 approximate 64 k. So, when a packet goes down to the Ethernet level it has to be broken up into smaller frames.

So, there is some kind of fragmentation happening at the Ethernet level, just to avoid that some of the IP layers put a restriction that let us also limit the packet size to 1500, MTU of 1500, so that at the Ethernet level we do not have to carry out any additional fragmentation, this is one of the important points.

And the other important thing is that most of the data packets that are generated, the IP packets they are anyway small; they are less than 1500. So, only for very rare cases you encounter larger packets. So, there you force that they be fragmented at the IP layer itself to create fragments of size 1500 or less so that it can pass through the Ethernet layer without any further splitting or breaking ok.

So, with this we come to the end of this lecture. Now, as you recall in this lecture, we talked about basically how IP packets can be fragmented and reassembled. We mentioned that IP uses non transparent fragmentation.

Thank you.