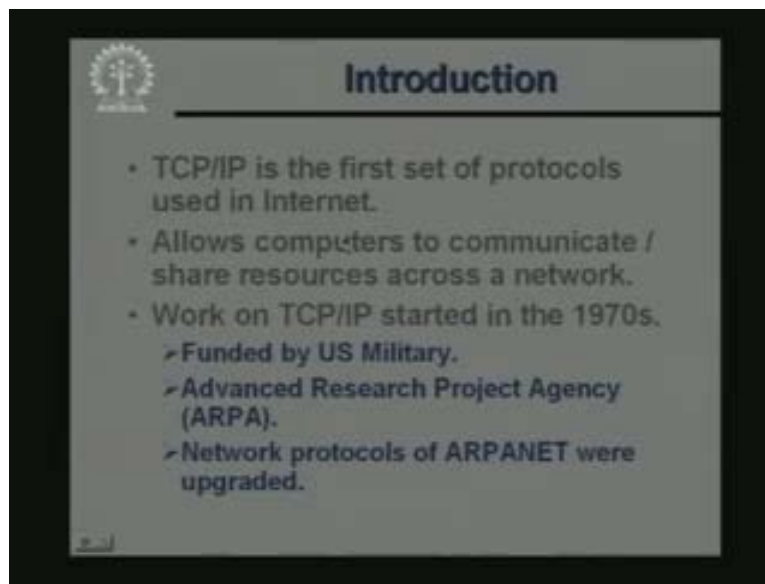


Internet Technology
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur
Lecture No # 03
TCP/IP- Part-I

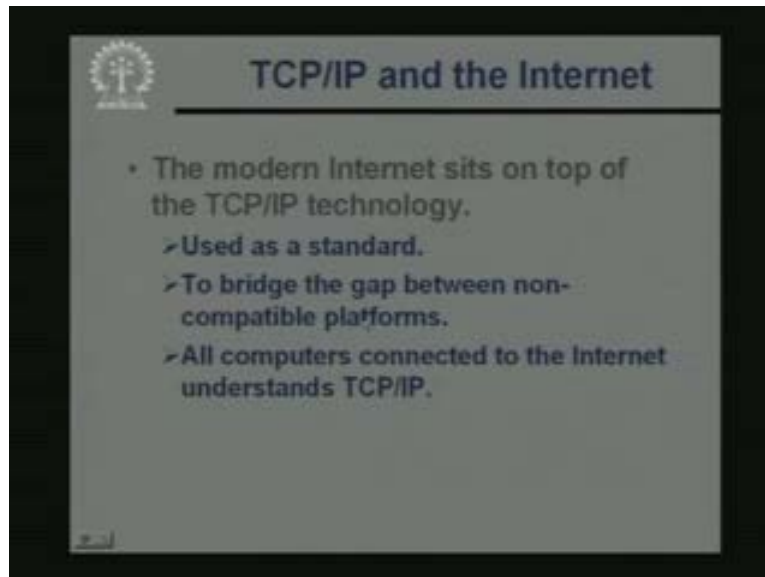
And this lecture is entitled TCP IP part one. So in this module two we would be covering the topics in 3 lectures. Here we would be mainly talking about the TCP IP protocols. In fact TCP IP is not a single protocol; it is a suite or a family of protocols. We would be looking at some of the important members of this family. But first let us try to understand what TCP IP is and how it has evolved over a period of time.

(Refer Slide Time: 01:33)



So talking about the internet, TCP IP is a protocol which well you can say it stated as early as in 1970s and it got very quickly accepted by a white community of users. In fact when the internet came into the being, TCP IP was the prime vehicle which was used to connect the computers in the internet and to allow them to communicate over the network. Using TCP IP the computers were able to communicate among each other. And also another very important thing, they were able to share some resources across the network. Some of the resources like dig space or some of the some of the expensive equipments were expensive in those days and over the network it was possible to share those resources. And work on TCP IP stated in the late 60s and in the early 70s it started to take shape. Now in US like most of the innovative developments, the initial research on TCP was funded by the US department of defense the military. So as part of their project it started with a very small network it was called ARPA. It was called advanced research project agency ARPA and the network which evolved in the process it slowly came to know as the ARPANET or the ARPA NETWORK. So this ARPANET was the first network you can say which started to use the first version of TCP.

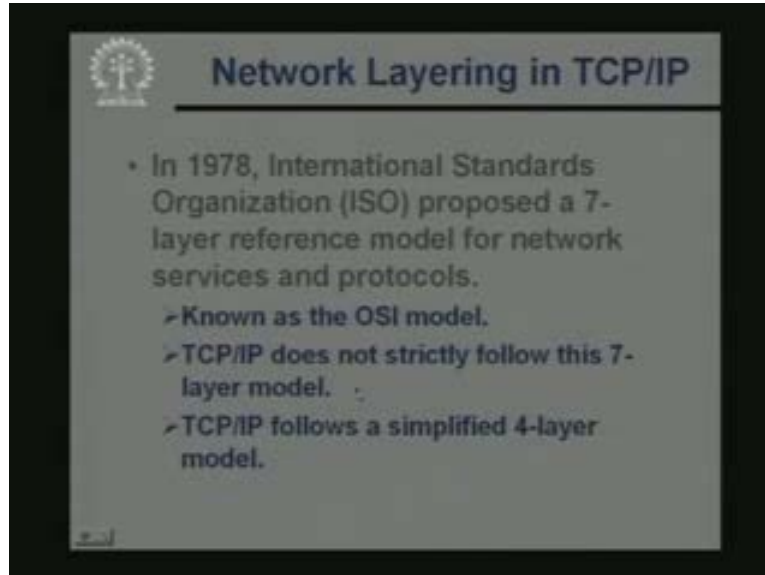
(Refer Slide Time: 03:35)



Now this TCP has become so widely accepted technology in modern internet that you can say that in today's world the internet as we see, this is entirely dependent or based on the TCP IP technology which lies under it. So in the internet TCP IP today is treated as a standard. Standard means say when you buy or purchase a new computer and if you want it to get connected to internet the first thing you must ensure that your computer learns or understands the language of TCP IP. Because this is important because all other computers which are connected to the internet. They know and understand TCP IP. So if your computer can talk in the same language then you can communicate with the others. But suppose you have a computer which has a proprietary system, it uses something other than TCP.

Then it will not be possible for your computer to send a message or communicate with some other computer which is connected to the internet somewhere else. So this TCP IP is a standard it has been used to bridge the gap between non compatible platforms. Well you say non compatible platforms, what I mean to say is that some of the machines you are trying to connect may be running the windows operating systems. Some may be running some version of UNIX, Linux; some may be Macintosh machines, macs. So there are wide varieties of types of machines and operating systems which are in the use today. So if all of them have a common layer of software namely the TCP IP then using that common layer they can very easily talk among themselves and work in you can say synchronizing with each other. So the most important point you notice that all computers connected to internet today must understand TCP IP, this is important.

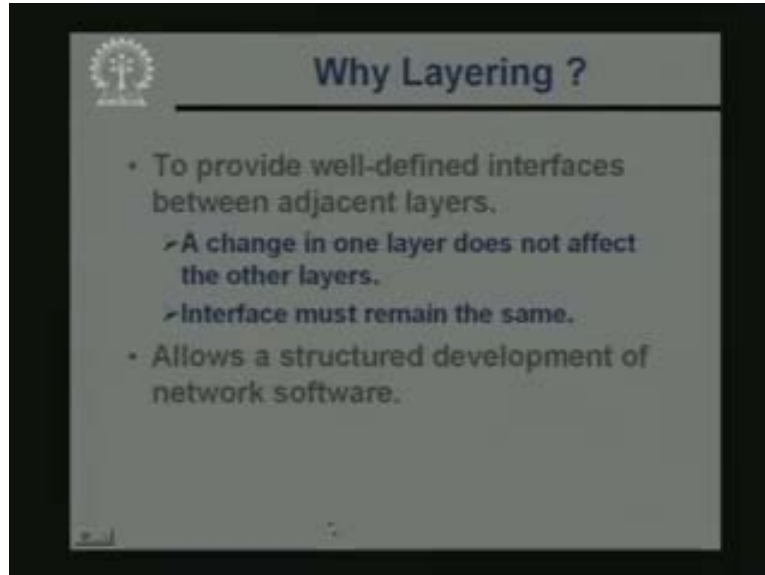
(Refer Slide Time: 06:03)



TCP IP, as I had said this is not a single protocol. This is a family of protocols and you recall earlier we had mentioned about the international standard organizations 7 layer reference model which is known as the OSI model. Now this OSI model has I had mentioned this had evolved through a consortium of a number of user groups. They had identified the different functionalities of the network software and they try to find out what are the basic functionalities and the relationships. If you can separate them out into layers then the development of the networking software may become much easier. So analyzing the requirement and the interaction and the functionalities of the different levels of network usage they identified 7 different layers in the hierarchy. Now as the standard what you were developing this seven layer architecture internet sitting on top of TCP IP as already started to grow or evolved.

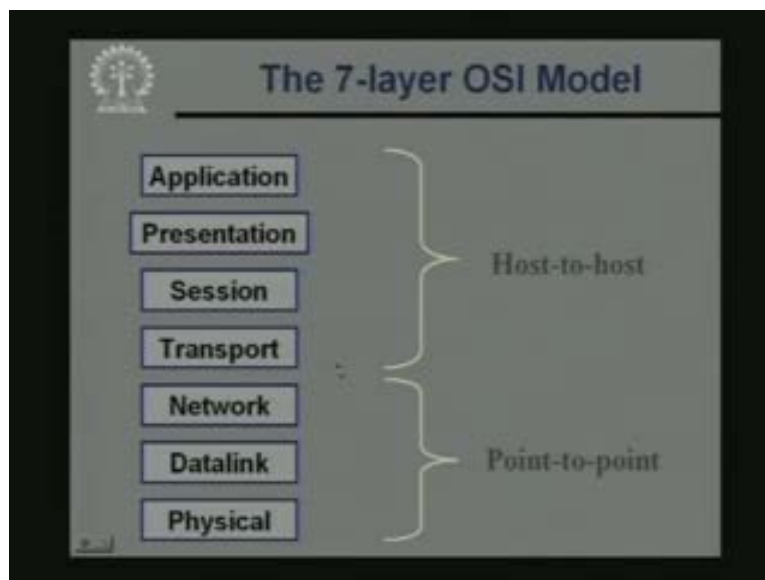
So larger and larger number of users they started to use internet, they started to get familiar with TCP IP and TCP IP. You can say was a parts and parcel of their daily activity or work. So even though the 7 layer OSI model was a very well thought of model it had a very sound logical basis of this layering. But, TCP/IP which is not based on such a strong logical base, but it attracted a much larger user community. Because there was an application name with the internet which was in place and it had some very interesting properties and it stated to grow very rapidly. So TCP IP became much more acceptable to the user community. So the truth today is that all though the 7 layer model may be a better way of carrying out the layering, but in reality most of the protocols which are in use today, they are based on TCP IP which is not a 7 layer model. In fact TCP IP follows a simplified form of the 7 layer model; you can identify 4 layers in this model. Now let us see how this modeling is carried out.

(Refer Slide Time: 09:00)



But first, let us try to have a look again at the purpose of layering. Well we develop software in terms of layers. Because number one is that if we make some change in one of the layers as long as the interfaces remain the same, the other layers need not be modified. So if the interface remains the same we can make changes to a layer pretty easily. This is of course one reason; the second reason is that if we have well defined layers your total software can be defined into well-defined modules. So the software development process also becomes much easier.

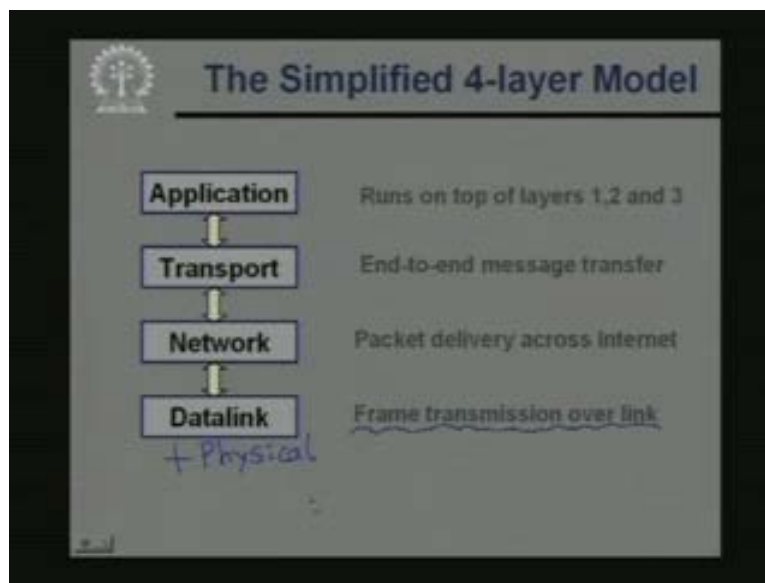
(Refer Slide Time: 09:57)



Now just to recall the OSI 7 layer model, had these 7 layers starting from the lowest level physical, data link, network, transport, session, presentation and application. Now you recall one thing. Suppose you have a network like this, there are several nodes connected. Right these are several networks nodes connected and there are some links which are joining them. Suppose this is my source and this is my destination. I want to send some message from the source to this destination from the source to this destination. Now there are some of the layers, in fact the lower 3 layers physical, data link and network. They work on a point-to-point link means they work or they cooperate in sending the data over this link, over this link and so on.

Whereas the higher 4 layers for them the intermediate thing is more like a black box. Only the 2 end systems are important and whatever this source is sending as if the destination is receiving directly. This is the advantage you have in layering the lower 3 layers of the software. These are working on the point-to-point to links. But they ensure that the packets which are starting from S will finally reach D. In contrast the higher 4 layers, they are not worried about how the packets are delivered to D. They are more worried about what is reaching D. Finally these are therefore called host-to-host layers because for them the internal network is a black box. Only the two communicating entities are the 2 ends the source and the destination. They are important as if the source is directly talking to the destination.

(Refer Slide Time: 12:17)

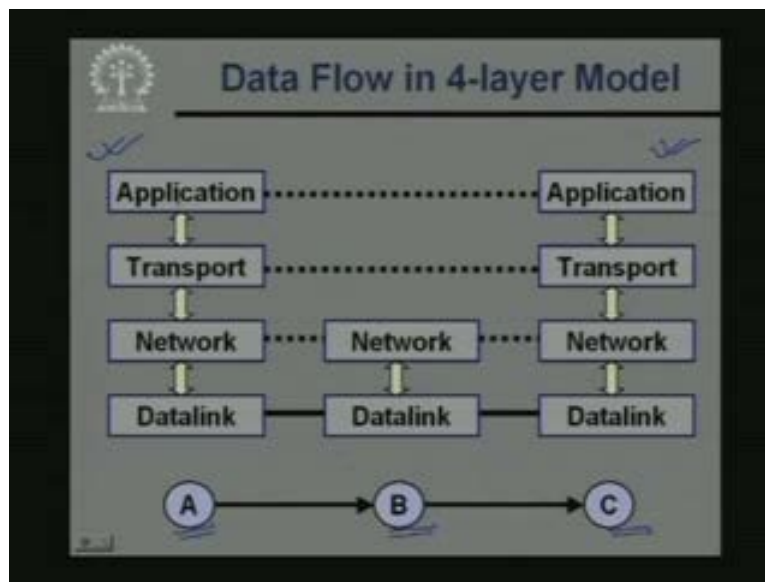


Now in TCP IP the simplified 4 layered model which is used, well if you draw an analogy from the OSI model approximately the layering would look like this. At the lowest level this is sometimes called the data link layer. This layer will be responsible for the transmission of frames over link. But one thing you understand all thought the lower level link is called the data link layer. This must also include the physical layer. Now the reason they are put together in a single block or layer is that in most of the modern systems what happens is that we have also called network interface card or a nic which

we put inside your computer and through a networking cable we connect that nic. Now that nic that piece of hardware it works both as a physical layer and as the data link layer. So both the two layers get embedded into the hardware interface card. That is why we really cannot distinguish the two layers in modern day systems.

That is why they are often embedded into a single layer. Now the next higher layer the network layer. This is responsible for packet delivery across the internet or source sends a packet. It is the network layer in all the intermediate nodes as well as the source and destination. It will be responsible for routing or forwarding the packet systematically from the source to destination along the correct path transport layer is the end-to-end link where the source and destination can talk directly and application is any program which can run on top of these. Now examples of applications may be electronic mail. Email may be any other program you are running which is communicating with any other client server programming, in fact which is communicating with another program on the other machine. They can be treated as application.

(Refer Slide Time: 14:52)



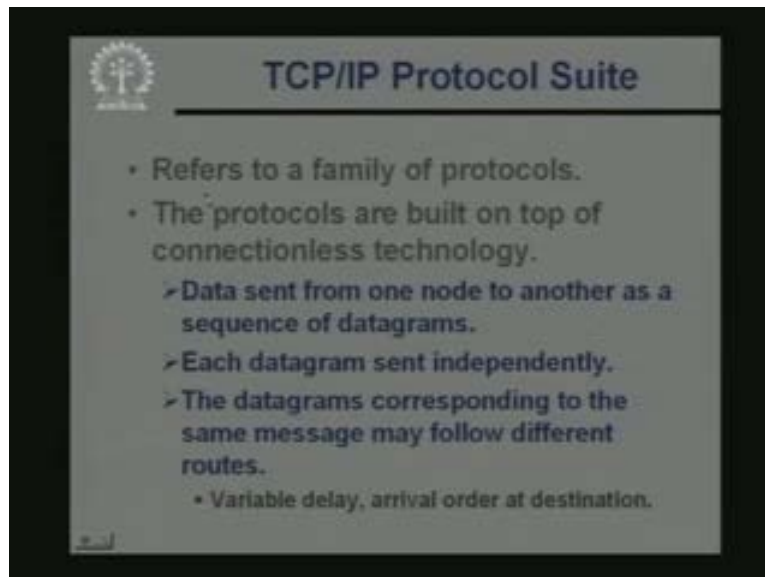
Now in a 4 layer model, let us look into this diagram in a slightly more detail. Now what we are showing here is that there are 3 nodes A, B and C. Suppose A is the source, C is the destination and the packets are the data from A, go to C via an intermediate node B. So when I say that A is the source and C is the destination I mean is that on machine A, there is some application program which is running. Similarly on machine C there is some other application which is running which will be receiving the data send by the application running on A. So what the application program running on machine A will do is that, it will send some information down to the transport layer. Transport layer will be sending it down to the network layer.

Network layer will take some decision that will it will be looking at the destination address. The address where the packet has to be delivered and in case this node A has

more than one outgoing links. It will take the decision through which outgoing link the packet has to be forwarded. So the network layer will be sending back the information to the data link layer corresponding to this selected link. For example I have selected this link, so over this physical link the data packets or the frames will go to machine B. Machine B is not the final destination. Now here again there can be several outgoing links. So what this machine B will do, it will again send back the packet to its network layer. And the network layer will again take a decision depending upon the destination address that over which link we have to forward it.

Suppose it selects this link, so now from the network layer it again comes down to the datalink of that corresponding selective link. So finally the packet reaches C. So from C again it traces the reverse route datalink to network network to transport and transport back to the application. So the point to note is that in all the intermediate nodes only the networks layers up to the network layer, datalink and network these two layers are coming into the picture. One is at the two ends where the end to end layers like transport and application are being used. So this diagram shows that how typically data flows in a four layer model network model. So now let us come to TCP IP specifically what the TCP IP suite of protocols really mean?

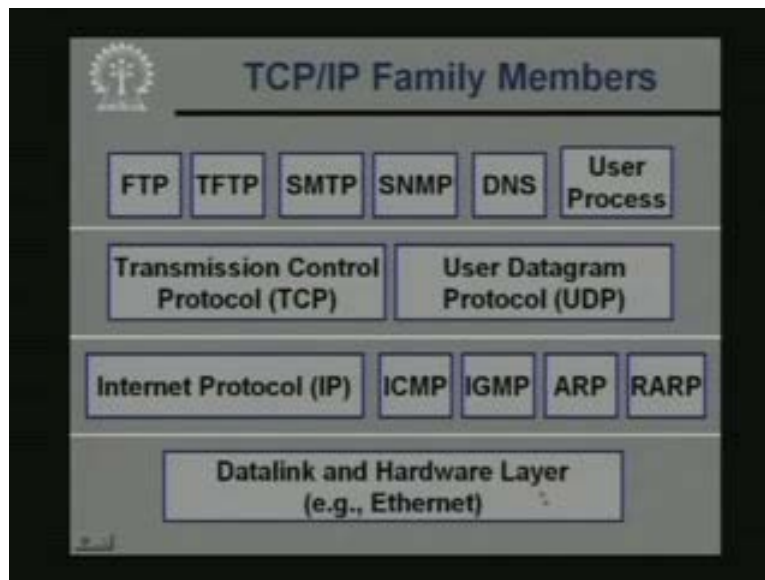
(Refer Slide Time: 18:01)



Now as I had mentioned TCP IP does not refer to a single protocol or a few two or three protocols. In fact this refers to a family of protocols. All these protocols are built on top of a so called connection less technology. Now connectionless technology as you had seen earlier, these means datagrams. There is a basic mechanism of transporting datagrams over the network. So in TCP IP the basic idea is that data will be sent from one node to the next as a sequence of datagrams. Now exactly the way datagrams work each datagram is independent of the others. So the datagrams will be sent independently. Now again it is a characteristic of datagrams. Since they are send independently there is no guarantee that the datagrams that comprise of the same message will be following the

same path. Say suppose I have a message I break up the message into 5 datagrams and I send them independently, each of these 5 datagrams may be following different paths. So there may be reaching the destination following different paths. And accordingly the order of arrival may be different. Some of the datagram may get lost in transit and if there is some error checking time out retransmission then some duplicate datagrams may also get generated. So variable delay arrival order at destination are possible.

(Refer Slide Time: 19:57)



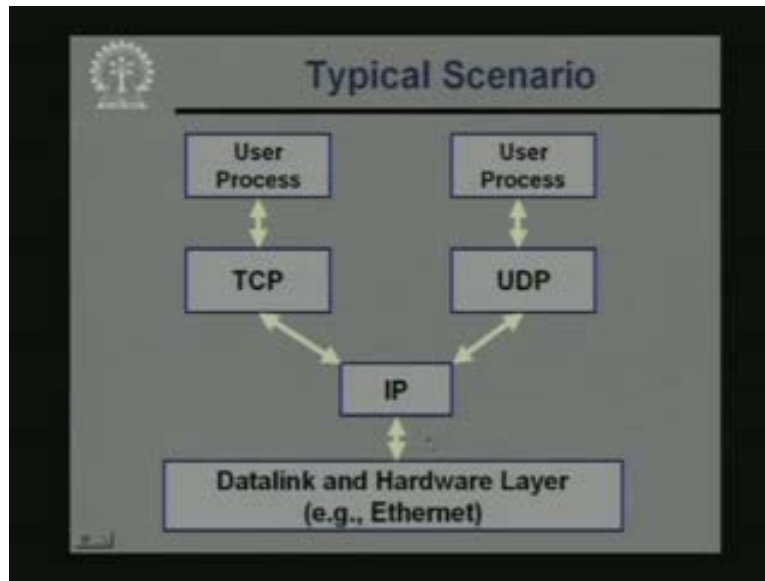
Now let us have a very quick look at the important members of the TCP IP family. This diagram summarizes the different important family members in TCP IP. Of course at the lower level we have the data link and the hardware layers this refers to the network interface card as I told you. Typically we use Ethernet at this level at the next higher level, this is the network layer level in TCP IP the protocols that works at the network layer level is called internet protocol or in short IP. So IP is the main protocol which is working at the network layer level. But in addition there are several auxiliary protocols like internet control message protocol, group message protocol, address resolution protocol and reverse address resolution protocol which also works at the network layer level.

But they have specific functions like ICMP is used to generate and send some error messages, like for example address resolution protocol is used to translate from an IP address into an Ethernet address. For example, so IGMP and RARP also have some specific purposes. But in a typical application you will be using internet protocols. These protocols are typically hidden from the applications. These are invoked and used in a transparent way. As a user you will not be able to see that they are been used or as a programmer also. You will not be using them directly. Now at the transport layer level you have two options. You can either use transmission control protocol or TCP or you can use user datagram protocol or UDP. So you see although we call it TCP IP actually

you can also use UDP and IP. TCP IP is just a name for this family moving up to the application layer level.

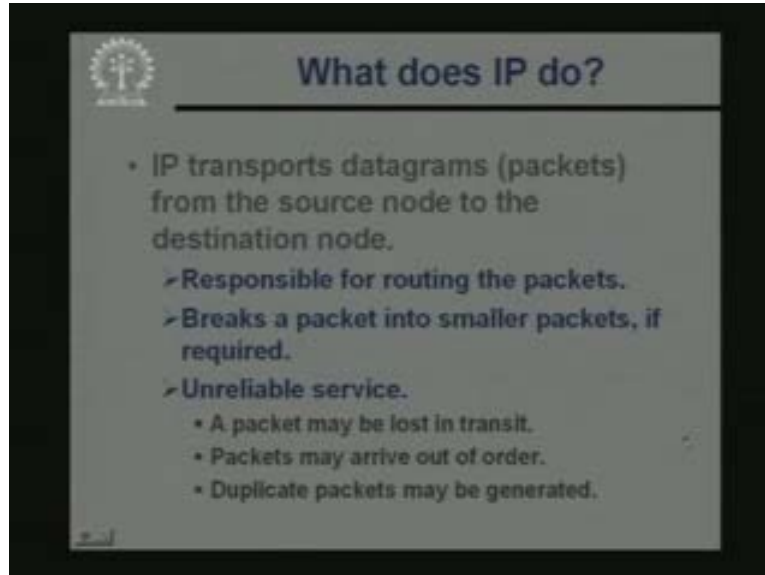
So at the application layer level there are a number of well-defined applications. Only a few of them I have listed here we have file transfer protocol, we have trivial file transfer protocol, we have simple mail transfer protocol, we have simple network management protocol, domain name server or you can have any general user application which has been developed. This is only a short listing, general you can have any number of applications at this level. So typically an application running on this level. For example this user process. This can interact with either TCP or UDP depending on what you want. TCP or UDP in turn will interact with IP then IP will interact with the lowest layer. This diagram shows that simplified typical data flow model.

(Refer Slide Time: 22:30)



This is the typical scenario for at the application layer level the user process they will be using either TCP or UDP. There is a choice, there is a characteristic, they will be in turn using IP and IP will be interacting with the lowest layer. So the most important members of TCP IP family are TCP, UDP and IP; these three. So first let us try to say briefly what the basic functionalities of three models are? First let us look at IP the protocols that works at the network layer level has. I had mentioned that a network layer protocol is responsible for the correct routing of a packet from the source to the destination. So it will take some decision that where to forward or send the packet next and the packet will find its way from the source to the destination taking help of the IP layer. IP layer software that is running on each of the intermediate nodes.

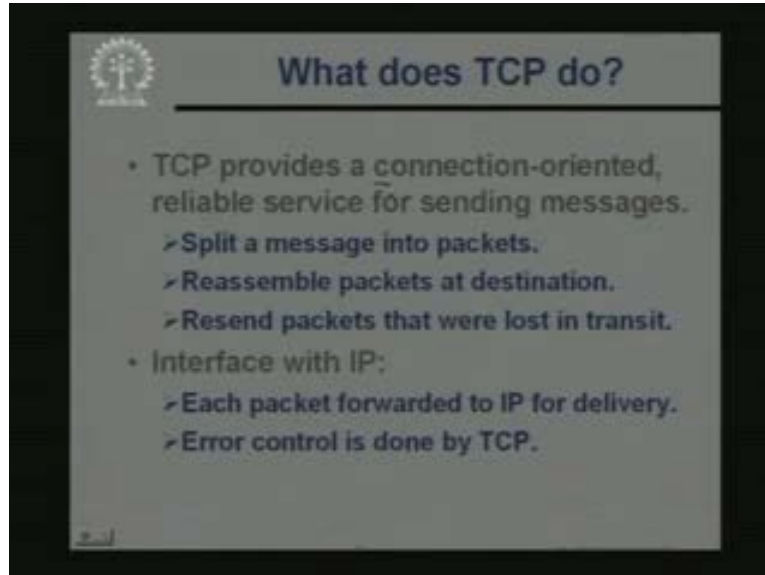
(Refer Slide Time: 24:58)



Basically IP is a transport system for datagrams. So at the level of IP it is only datagrams which flow on the network. So IP is basically a datagram packet delivery system. It is responsible for routing the packets. Of course moreover if it sees that a packet is too large it cannot be handled by a network, then it can break a packet into smaller pieces which are called fragments which will be talking about later. That how it is done? So a packet may be broken into smaller fragments of packets which later will need to be reassembled again to get back the original packet. Now this reassembling is done at the final destination. But the point to notice that IP does not do any kind of error control. It is an unreliable service, it is basically a datagram service and all the unreliability that comes with datagram it stays with IP also like a packet may get lost.

Packet may get arriving, may arrive out of order because the individual packets may follow different paths to the network. And in case of time out and retransmission duplicate copies of the same packet may be flowing through the network. So the final destination the IP layer at the final destination has to manage all these things packets arriving out of order. Some packets not coming at all and multiple copies of the same packet coming. So these issues have to be handled there. Now looking at TCP which works at the transport layer level and you recall the transport layer level is an end to end protocol where the two hosts which are running on the two machines they talk to each other.

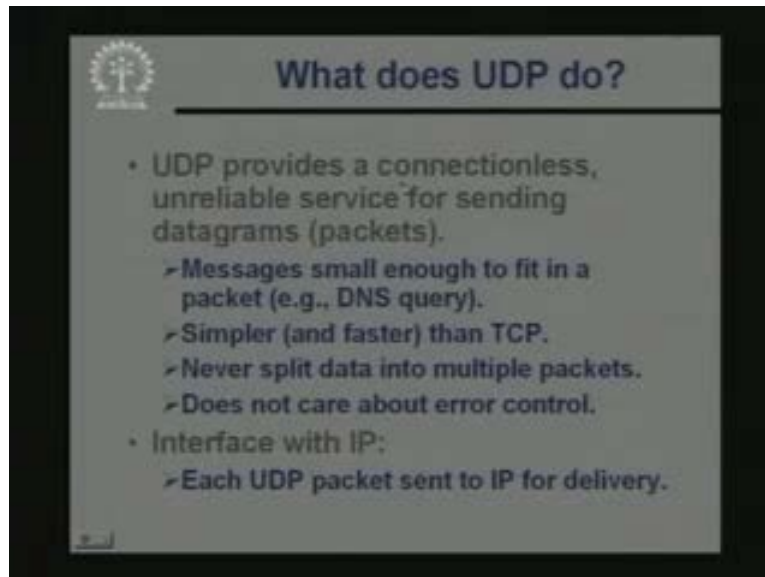
(Refer Slide Time: 27:09)



TCP is a reliable transport layer protocol, well when you say reliable. What do you mean is that it is connection oriented reliable service. When you say connection oriented, well the user or the user application gets an illusion that there is a connection which has been established between the two end systems. So the application program running on one computer believes that a connection as been established with the application running on other end system. And I can send some message directly to the other machine. But what happens in practice is that ultimately the data will be transported by the IP layer below. When IP is an unreliable layer. So TCP has to do some additional error checking and in case of error TCP will have to recover from the error. So TCP will be given illusion to the layers above it that well everything below you is a very reliable network. So essentially the tasks which are performed by TCP is that splitting a message into packets, reassembling the packets at the destination again and it checks if some of the packets are missing.

So if there are missing and explicit request is sent back to resend that packets and in this way reliability is ensured. And TCP sits on top of IP so the way it interfaces is that the packets which TCP generates will be forwarded to IP individually for delivery. But IP does not do any error checking or error control. So all error control is the responsibility of TCP. So this is a nutshell is what the TCP layer is supposed to do and the user datagram protocol called UDP. That is also a transport level protocol, but that is different. Well as compared to TCP, TCP tries to provide reliability it tries to establish provide a connection oriented service. But UDP does not try to anything UDP says that well I am trying to transport a packet as fast I can, I am not concerned about reliability. I am not concerned about in which order the packets are going. I will try to impose a minimum amount overhead. I will try to deliver the packet fast this is the philosophy behind UDP.

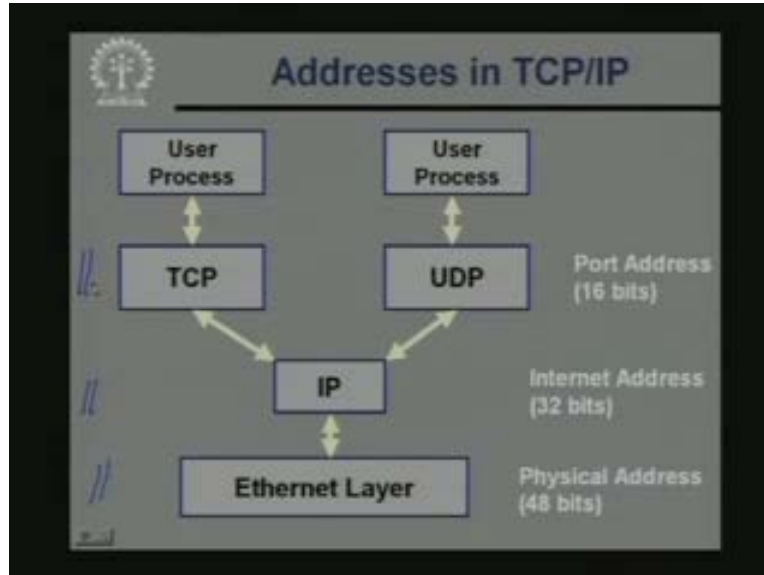
(Refer Slide Time: 29:52)



So UDP provides a connectionless service like there is no apparent connection that is set up between the source and the destination. This is an unreliable service because it does not do any explicit error checking or error control. The applications which prefer to use UDP are those for which messages are small enough which can be fit into a single packet. A typical example is the query that is sent out by the domain name system. DNS query is an example of such a message. In fact DNS uses UDP for that purpose because the protocol is much simpler. You do not have to establish a connection, do not have to maintain a history of the packet so that if a packet gets lost you cannot send back a request for retransmission.

So since you are not doing all these book keepings, this protocol is much simpler in comparison. This is simpler as well as faster as compared to TCP. So since it is simpler it will never split a data into multiple packets. It is expecting that the packet will be of a size which can fit into a single datagram and it does not care at all about error control interface with IP is very simple each UDP packet is a datagram it is simply forwards to IP for delivery. So interface with IP is also very simple here. Now there is another issue.

(Refer Slide Time: 31:36)



Well if you have another look at this TCP IP stack. Just showing the important protocols TCP, UDP and IP broadly there are three layers. If you recall the physical and the data link layer, the network layer and the transport layer. Talking about the addresses in each of these three layers, there are three different addresses which are used at the lowest layer. Just assuming the lowest layer is an Ethernet. Here you use something called the physical address of the interface link. In this case, this will be a so called Ethernet address and this is a 48 bit address. This 48 bit Ethernet address is embedded inside the network interface card which you plug into your computer. So it is that card which a hardware address has built into it that is the Ethernet address. So when a frame finally comes to your computer, it must come with the Ethernet address as part of the packet or frame.

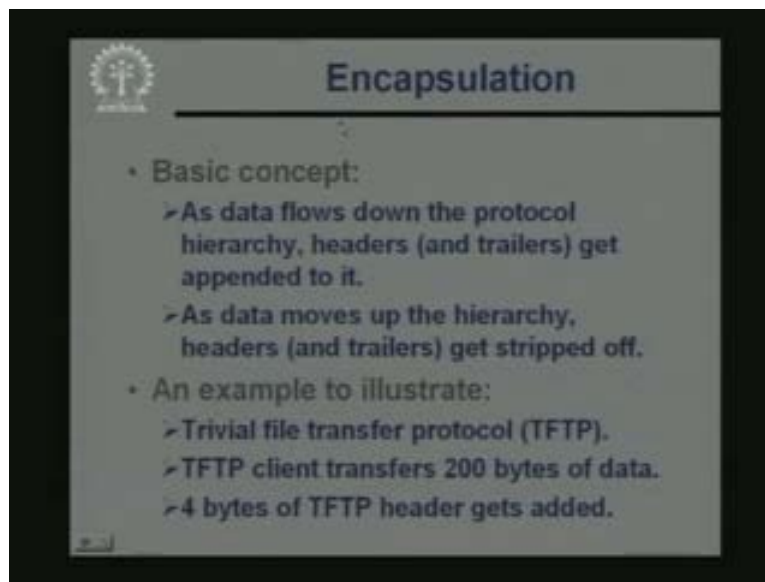
But when you are looking at the network layer it is a 32 bit IP address which is used to identify the computers in the internet. Now this 32 bit IP address is allocated and assigned in a much more systematic way. This is a logical address, no computer comes with that address built in you can program a computer you can assign any IP address to it. Now the purpose of this assigning of IP address is that you are trying to make it easier for the intermediate nodes in the network to forward and route packets just by looking at the IP addresses. But once the packet enters or comes into the LAN where the final destination node is situated, so there you have to finally find out the Ethernet address of the destination. That is done through that ARP protocol I talk I talked about a little ago. So using ARP you get the Ethernet address and finally the packet will reach the final destination addressed by the 48 bit Ethernet address.

Now at the level of the transport layer, now you recall the transport layer is the end to end layer. So here two applications on two end hosts are communicating. Well here also there is an address called port address. It is a 16 bit wide address, well let us see why this is required or needed. Now at the transport layer level when there are two end hosts and two applications running there are trying to communicate, it may be very much possible that

would each of these machines. There is not only one application but several applications running. That is the typical scenario in most computer systems today. There is time sharing, there is multi-tasking. So there are many programs which are running at the same time. So you must mention that when you are sending a message to exactly to which application program at the other machine you are trying to send the message.

So this port number in essence identifies an application which is running on the particular machine. This is why we have hierarchy of addresses. Physical address to ultimately identify the machine through its hardware interface card. This IP address in order to identify the network where the machine belongs and finally to send route the packet to that network and the port address to identify an application running on the end machine end host. Now any layering of software just see this networking software all of them have some sort of layering. We talked about the seven layer OSI model. Now we are talking about the 4 layer TCP IP model. Now when there is layering the packets flow in the systematic way from top to bottom and again from bottom to top. Now as this information flows up and down some additional information gets appended or removed during these movements. Now let us see what this means and why this is done?

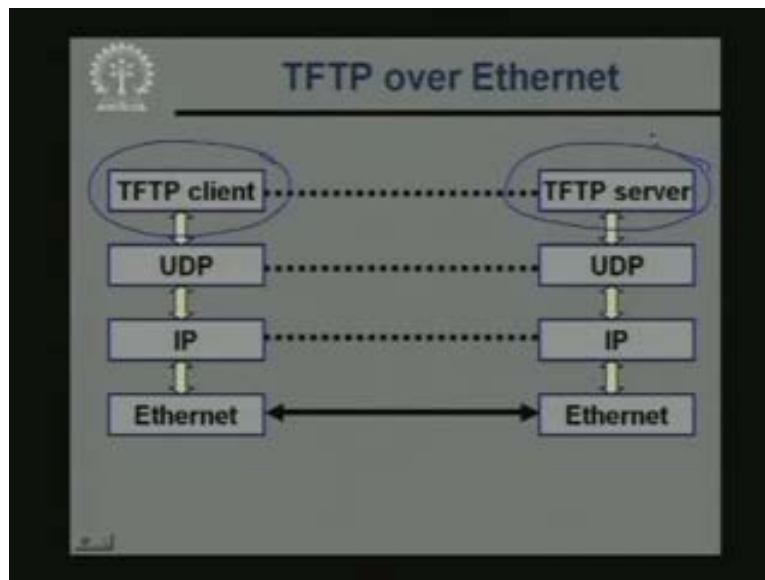
(Refer Slide Time: 36:47)



Now this addition of this information as a packet moves along the hierarchy is called encapsulation. So let us look at this issue of encapsulation now. When some information is data flows down the protocol hierarchy from the top to the bottom, some information gets added to it. These are some additional information, this additional information are in the form of headers and possibly trailers. So you have the original message or data you add some more information to it and send it to the lower layer. Similarly the lower layer will add some more information to it and put it to the next lower layer. So in this way it goes on. But when it reaches the other side the flow of data is reversed data moves up the hierarchy now.

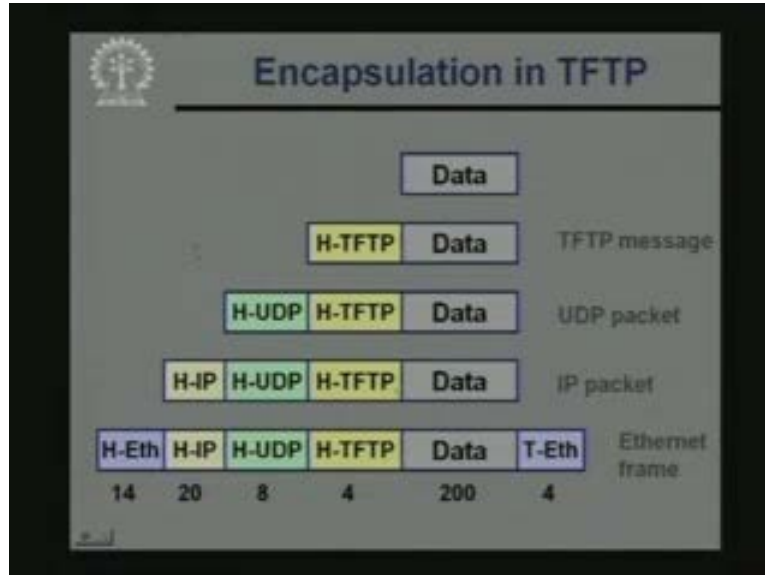
Now as it moves up these headers and trailers which were appended, they get removed systematically one after the other. Now here let us try to illustrate with the help of an example. Let us take an example of an application layer protocol. A simple application layer protocol called trivial file transfer protocol. Well for those of you who are already familiar with the file transfer protocol. Just let me tell you that this TFTP is a simpler version of the FTP protocol which uses UDP for delivery of the data. So TFTP is a simpler version of FTP. So we assume that the TFTP client is trying to transfer 200 bytes of data and this TFTP protocol, this appends a header which is of size 4 bytes. So with this information let us see how this encapsulation takes place.

(Refer Slide Time: 38:57)



First we look at the four layer protocol diagram Ethernet, IP, UDP and at the application layer level. On one side you have the TFTP client and the other side you have the TFTP server. So first when the data is being sent the data will slowly move down the hierarchy from TCP, TFTP to UDP to IP then to Ethernet. Then it will actually flow through the physical link and on the other side it will be flowing up the hierarchy. So as it moves down the headers and trailers will get added and as it moves up the headers and trailers will get removed. So here in this example let us see, as I mentioned TFTP runs on top of UDP.

(Refer Slide Time: 39:55)



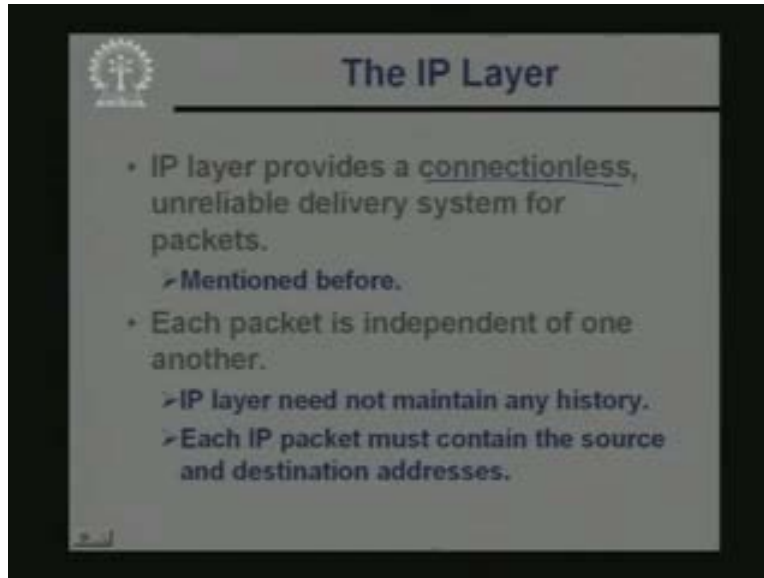
Let us see how the headers and trailers are appended. So in this diagram we are just showing how the headers and trailers are getting appended. So well when you say appended? If you look back at the previous diagram, so were actually talking that when the messages moving down the hierarchy. So when it moves up the hierarchy, exactly the reverse process will take place the headers and trailers will get removed. So this is the data which we are trying to send. Now this data was 200 bytes long. First the data gets encapsulated as a TFTP message. The TFTP protocol appends 4 bytes of header to the data to produce a message packet. This header will be required by the TFTP protocol of the server running on the receiving end. Because the receiving end will be looking at this data and will be deciding what to do with this.

Now this message packet, then gets forwarded to UDP for delivery. So the UDP protocol also appends a header out here. This is an 8 byte of header. So UDP header is of size 8 bytes. Similarly UDP will be sending this packet to IP for delivery. So IP will again be adding a header. Now for an IP the minimum header size is 20, I am assuming minimum it is 20. Finally for the physical delivery of the packet it has to be send down to the Ethernet layer and Ethernet layer appends both a header of size 14 bytes and a trailer of size 4 bytes. So here you see that you were trying to send a data of 200 bytes. But ultimately if you add this numbers up it becomes 250. So in order to send 200 bytes actually we are sending 250 bytes over the network.

These additional 50 bytes constitute the overhead. Now this is a characteristic of any transmission or communication that we attempt over the net. So whatever we sent there will be some additional overheads that you have to take care of. So this encapsulation example shows you that how this encapsulation adds some headers and trailers and these leads to additional overheads in the packets. Now let us look at a slightly more detailed look at the IP datagrams or the IP protocol. As I said that IP protocol takes a packet from the TCP or UDP layer and it appends a header as the example we just saw showed and it

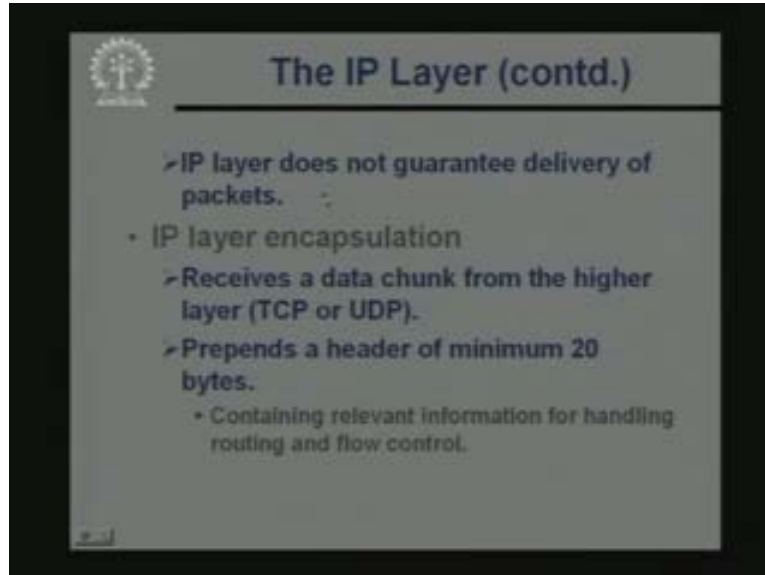
sends it down to the Ethernet layer. So first let us try to understand what kind of header the IP layer puts in? What are the contents of the header and their purpose?

(Refer Slide Time: 43:28)



Let us try to understand that first. So first just to recall we had mentioned briefly earlier IP layer provides the connectionless and unreliable delivery system for packets. This we have already mentioned before. Since this is a datagram delivery service each packet is independent of the other and since they are independent IP layer need not maintain any history of the packets; each packet is sent individually. Now since no history is maintained each packet must contain the source and the destination addresses. Because this is very much similar to the why you post a letter. So a letter is posted as an individual entity. So unless you post the complete destination address, there your letter will not reach. And if you want an acknowledgment back you also have to put your own address. So source and destination addresses are required in this kind of scenario.

(Refer Slide Time: 44:40)

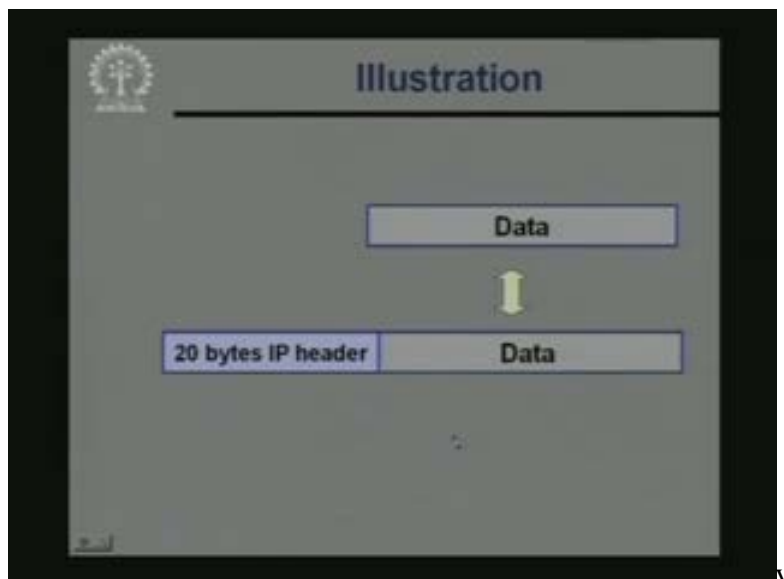


The IP Layer (contd.)

- IP layer does not guarantee delivery of packets.
- IP layer encapsulation
 - Receives a data chunk from the higher layer (TCP or UDP).
 - Prepends a header of minimum 20 bytes.
 - Containing relevant information for handling routing and flow control.

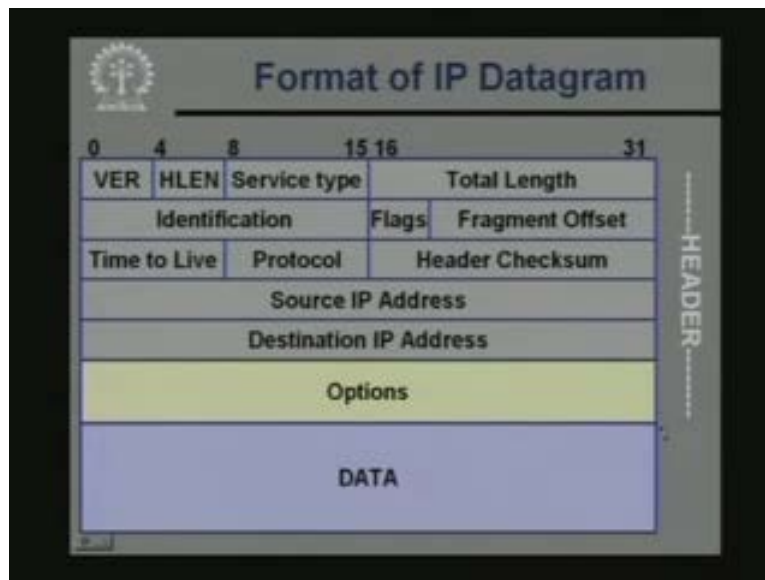
And since IP layer is a datagram based service error checking is not done. It does not guarantee delivery of packets. So if you need to do any error checking it must be done at the higher layer. IP does not do any error checking. It simply attempts to forward the packets through the correct route. It can fail a packet may be lost, this IP does not report any error. If such a thing happens or it does not try to recover from that error. And the way IP layer does the encapsulation is that it receives the data chunk. This you had seen in that example from the higher layer it can be TCP or UDP. It adds the header to it as we had said that the minimum header size is 20 bytes. And these 20 bytes will contain all the relevant information which will be needed for routing and flow control.

(Refer Slide Time: 45:44)



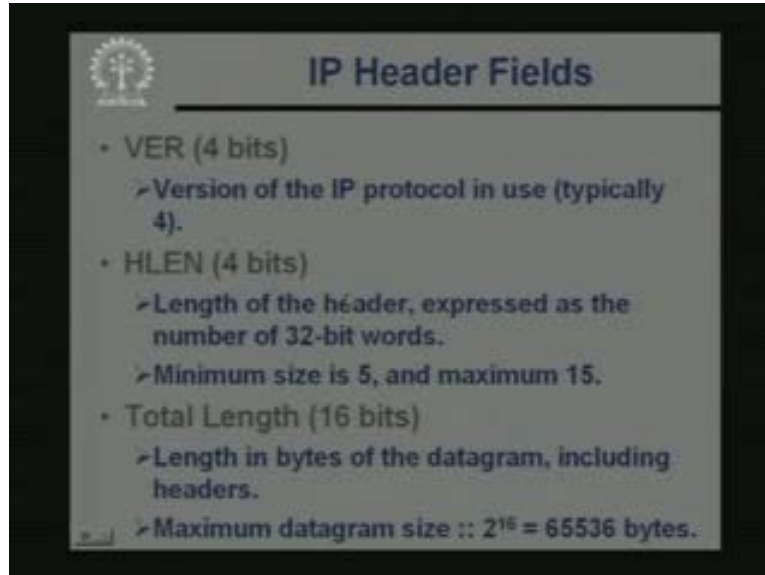
So now let us see what this encapsulation means diagrammatic. This means this we have the data and the IP layer will add this 20 bytes header to the data to form a so called IP packet. So this is the so called datagram which will get delivered. It takes the data from the higher layer TCP or UDP. This data will be coming from either TCP or UDP and these 20 bytes minimum header will be append to it and this datagram will be created.

(Refer Slide Time: 46:32)



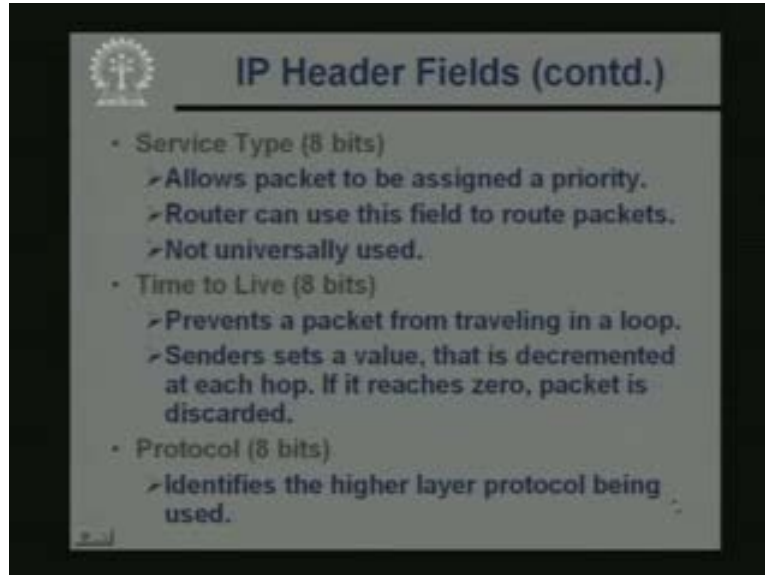
This diagram shows you this syntax of the IP header. In the previous diagram we had showed a data and the header part. Here the data part is out here. This section is the data remaining portion constitutes the header. Now in this diagram well every row corresponds to 32 bits or 4 bytes. If you look at the first 5 rows of this diagram, these are essential information and what we have mentioned here as options. These are optional items. So if there are some optional items the header size may become more, but minimum there are 5 rows which mean 20 bytes. Now let us look what are the fields we will be explaining. The purpose of these fields one by one VER means the version. This is the header, length, service type, total length, identification, flags, fragment, and off-set time to live, protocol, some kind of header checksum and source and destination IP addresses. These are the information which is present in the IP header.

(Refer Slide Time: 47:55)



Now let us see what these means. Version is a 4 bit field. This indicates that which version of the IP protocol you are using. Most of us today use version 4. Of course IP version 6 has already been standardized and it will be becoming more popular very soon. But still today the most popular version of IP protocol is the version 4. So the first field will contain the number 4 in binary. The second field the header length, this is also 4 bits, and this field indicates the length of the header expressed as the number of rows in the diagram or 30 bit words. Minimum size is 5 which mean 5 into 420 bytes in 4 bits. Maximum number can be 15. So including header including the options the total IP header size can be 15 into 4 or 60 bytes. The third field is the total length which is a 16 bit field. This indicates the total length of the datagram in bytes including the headers. Since it is 16 bits, the maximum datagrams size can be 2 to the power 16 or 65536 so many bytes. So this field will give you an upper limit to the size of the datagram.

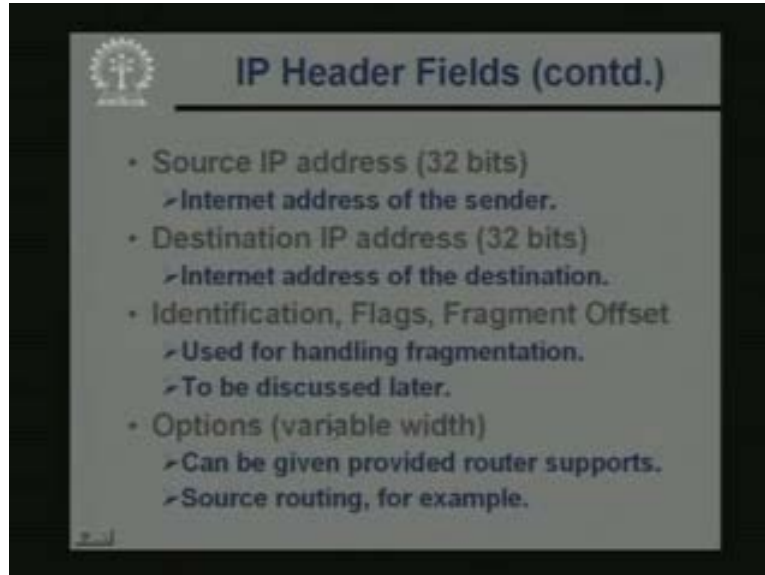
(Refer Slide Time: 49:37)



Well service type is a field which is not really used universally. So using this you can allow packets to be assigned a priority. But you can have this feature provided the intermediate nodes or the routers can interpret this field and can give higher priority. It can give higher priority to some of the packets as compared to the others. But normally very few installations we have this kind of a scenario. So this field is typically unused. There is another field called time to live, this is an 8 bits field, this is an interesting field see IP routing, see in IP routing as I said the IP layer will forward a packet from one node to the other. Now the IP layer at each node will take a decision where to forward it next. But suppose due to some reason the information that is available to some IP layers somewhere is wrong.

So instead of routing in the correct path it is routing through the wrong path. So it is possible that a packet may go on circulating in a loop and this may go on indefinitely. Now in order to control this you really do not want this to happen. So the time to live prevents a packet from travelling in the loop. The way it works is that the senders of the packet will set a value and store it in this field, this value will get decremented at each hop. Hop means from one node to the other as it moves on the value gets decremented. As soon as the value reaches zero the packet is discarded. So depending on the value the sender has set the maximum time the packet can stay gets limited. The protocol field indicates that which higher level protocol was being used TCP or UDP they have some specific values.

(Refer Slide Time: 51:42)

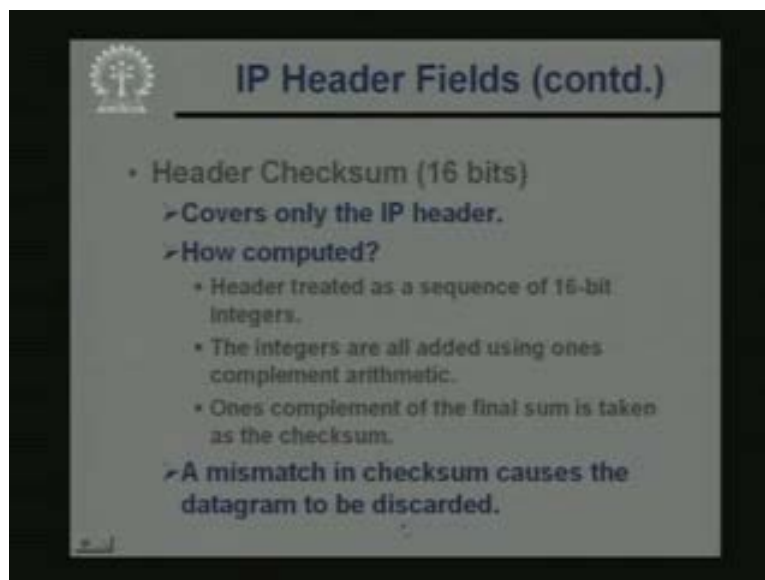


IP Header Fields (contd.)

- Source IP address (32 bits)
 - Internet address of the sender.
- Destination IP address (32 bits)
 - Internet address of the destination.
- Identification, Flags, Fragment Offset
 - Used for handling fragmentation.
 - To be discussed later.
- Options (variable width)
 - Can be given provided router supports.
 - Source routing, for example.

Similarly the IP header field, this source IP address, the destination IP address, this we would be looking at more detail later. These are 32 bit fields. There is identification flags and fragment offset these are used for handling. So called fragmentation this also would be discussed in later. Options are used if you need or if it is supported by the routers again well one feature that you can use. Using options field is source routing where the source of the packet can tell that which path the packets need to follow. So instead of leaving the decision to the intermediate routers the source can take that decision.

(Refer Slide Time: 52:32)

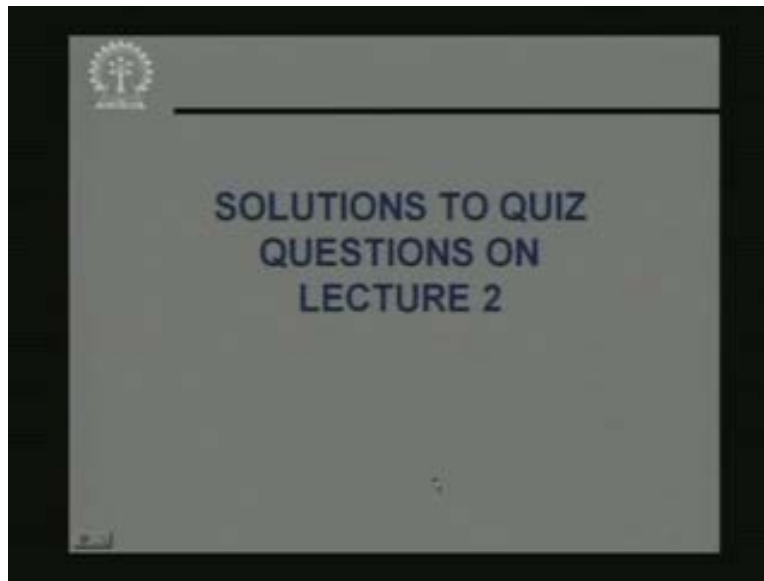


IP Header Fields (contd.)

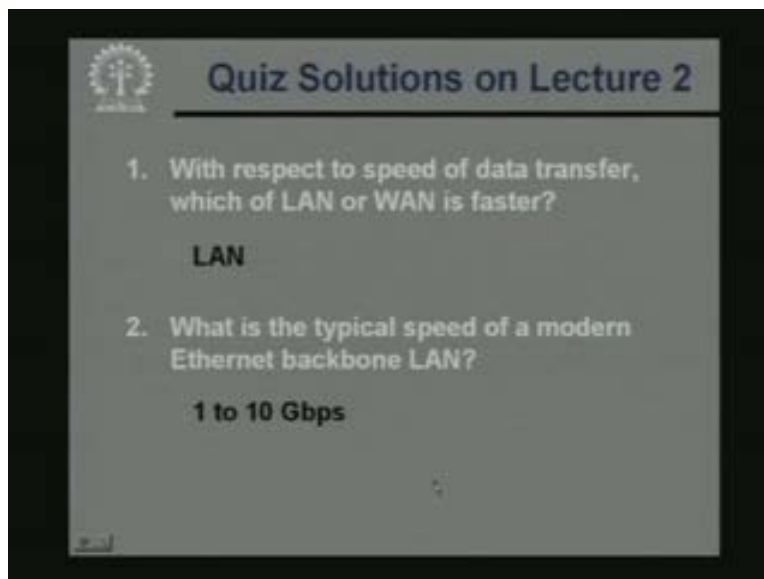
- Header Checksum (16 bits)
 - Covers only the IP header.
 - How computed?
 - Header treated as a sequence of 16-bit integers.
 - The integers are all added using ones complement arithmetic.
 - Ones complement of the final sum is taken as the checksum.
 - A mismatch in checksum causes the datagram to be discarded.

And finally the header checksum, there is a checksum which is computed only on the IP header and the way it is computed is that the header is treated as a sequence of 16 bit integers. These integers are added using ones complement arithmetic and ones complement of the final sum is taken as the checksum value. So if the checksum mismatch is found the datagram is simply discarded. No attempt to correct the error is made whether the datagram is correct or not. That is made so, with this we come to the end of this particular lecture, lecture number 3. So now let us have a look at the solutions to the problems which I had given in our last lecture, lecture number 2.

(Refer Slide Time: 53:32)

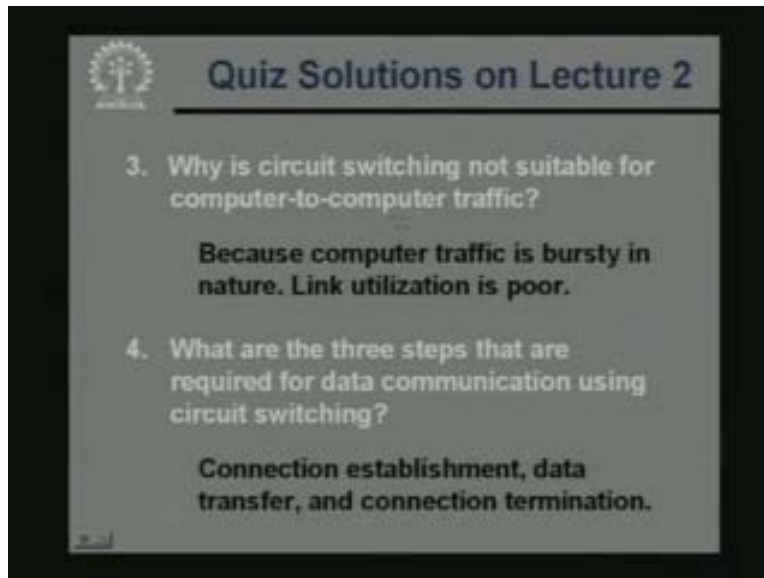


(Refer Slide Time: 53:40)



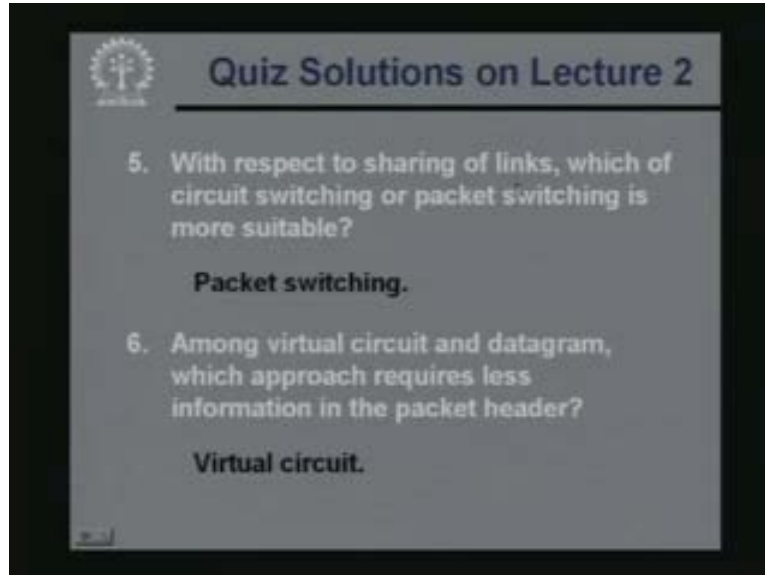
Let us quickly browse through this. The first question was with respect to speed of data transfer which of LAN or WAN is faster. Undoubtedly is LAN because the LAN you can have speeds which can range up to 1 giga bites per second or more. But in WAN it is much less. What is a typical speed of a modern Ethernet backbone LAN? Backbones LAN are faster typically we have 1 Gbps or in some modern installations it has gone up to 10 Gbps even.

(Refer Slide Time: 54:08)



Why is circuit switching not suitable for computer to computer traffic? Because the computer traffic is bursty in nature and in circuit switching you are establishing a dedicated connection. And for busy traffic you will not be utilizing the link at all times so link utilization will be very poor. What are the 3 steps that are required for data communication using circuit switching? Well first you have to establish the connection then transfer the data and finally the connection has to be closed.

(Refer Slide Time: 54:45)



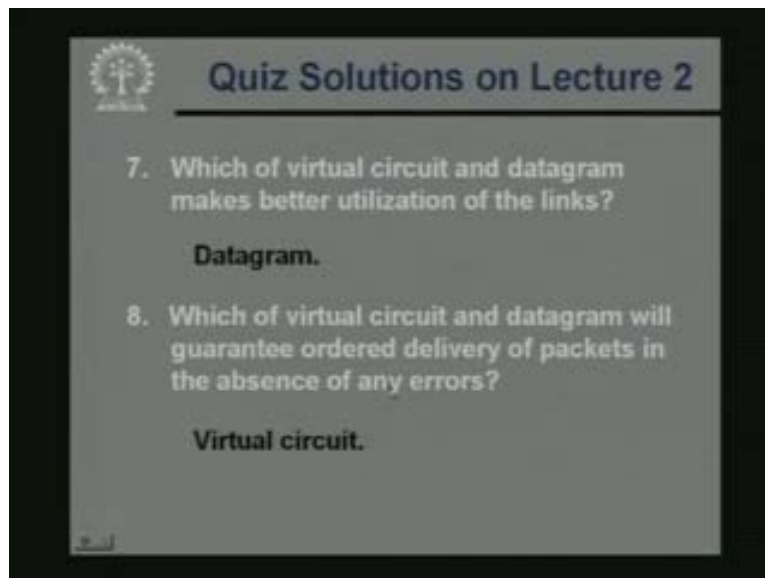
The slide is titled "Quiz Solutions on Lecture 2" and features a logo in the top left corner. It contains two quiz questions and their solutions:

5. With respect to sharing of links, which of circuit switching or packet switching is more suitable?
Packet switching.

6. Among virtual circuit and datagram, which approach requires less information in the packet header?
Virtual circuit.

With respect to sharing of links which of circuit switching or packet switching is more suitable. Because circuit switching some of the links may be dedicated. But in packet switching none of them are. So packet switching is more suitable in this respect among virtual circuit and datagram which approach requires less information in the packet header. Obviously virtual circuit this will say later also because in a virtual circuit the connection is already established. But in a datagram each packet must contain the source and destination address.

(Refer Slide Time: 55:20)



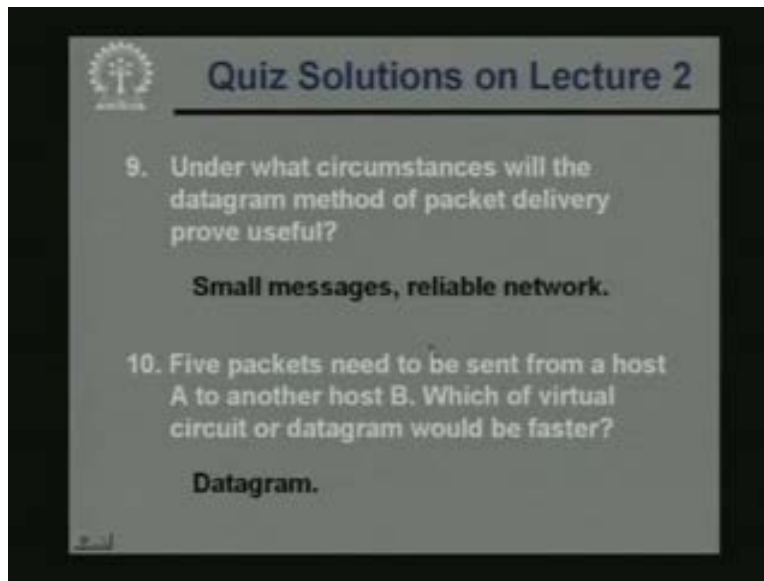
The slide is titled "Quiz Solutions on Lecture 2" and features a logo in the top left corner. It contains two quiz questions and their solutions:

7. Which of virtual circuit and datagram makes better utilization of the links?
Datagram.

8. Which of virtual circuit and datagram will guarantee ordered delivery of packets in the absence of any errors?
Virtual circuit.

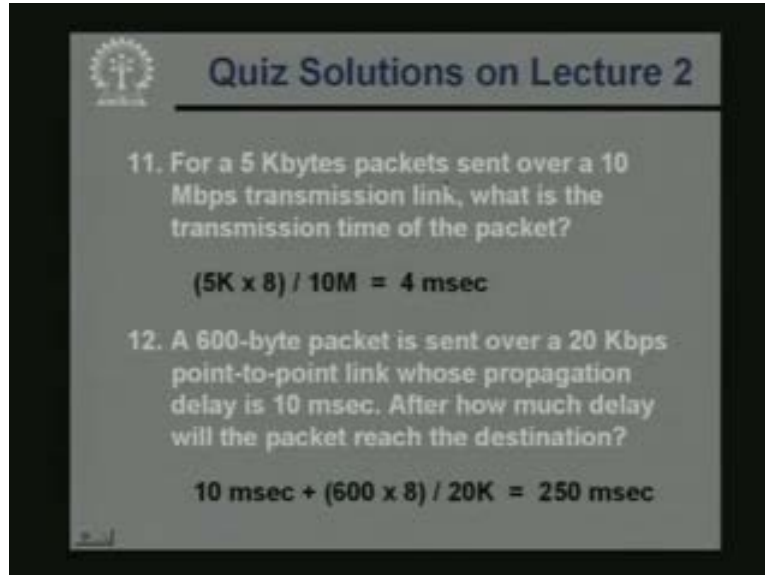
Which of virtual circuit and datagram makes better utilization of the links? Datagram. Because datagram may follow different paths and the links may be better utilized, 8 which of these 2 guarantees ordered delivery of packets. In the absence of any errors datagram does not guarantee ordered delivery because packets may follow different paths. But in virtual circuit the TCP layer will allow ordered delivery of packets.

(Refer Slide Time: 55:53)



Under what circumstances the datagram method of packet delivery prove useful small messages less errors reliable network. 5 packets, well 5 means few needs to be sent from host A to another host B which of virtual circuit or datagram would be faster. Datagram because you need not have to do any initial circuit or connection establishment that overhead is removed.

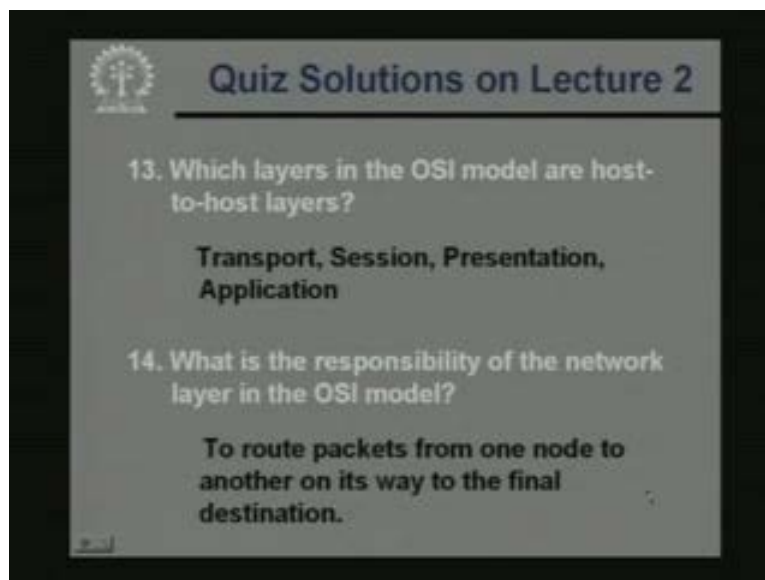
(Refer Slide Time: 56:24)



The slide is titled "Quiz Solutions on Lecture 2" and features a university logo in the top left corner. It contains two quiz questions and their solutions. Question 11 asks for the transmission time of a 5 Kbytes packet over a 10 Mbps link, with the solution $(5K \times 8) / 10M = 4 \text{ msec}$. Question 12 asks for the total delay of a 600-byte packet over a 20 Kbps link with a 10 msec propagation delay, with the solution $10 \text{ msec} + (600 \times 8) / 20K = 250 \text{ msec}$.

For 4 kilo bytes packets sent over a 10 Mbps link, what is the transmission time? Well this is easy to calculate 5 kilo bytes means 5K into 8 bits divide by 10 megabits. It is 4 milli seconds, a 600 byte packet is sent over a 20 Kbps link whose propagation delay is 10 milli seconds. After how much delay the packet will reach the destination while in this calculation you will first take the propagation time plus the packet transmission time, the total comes to 250 milli seconds.

(Refer Slide Time: 57:00)

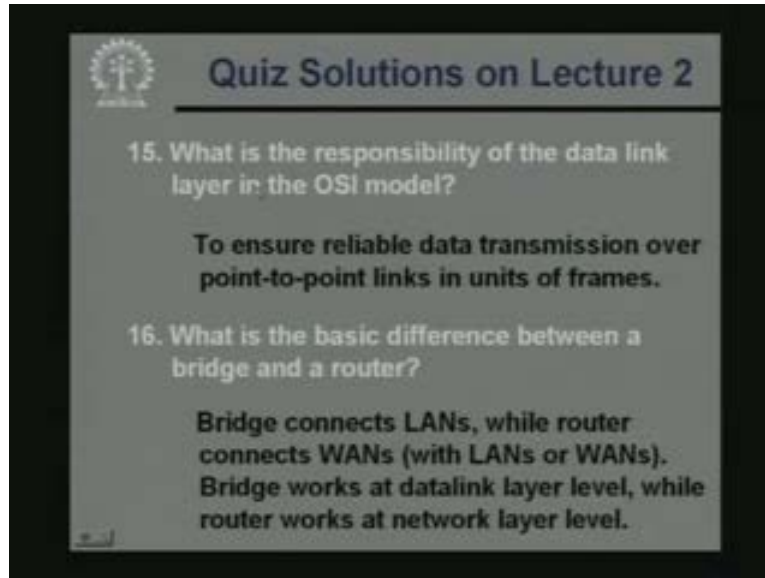


The slide is titled "Quiz Solutions on Lecture 2" and features a university logo in the top left corner. It contains two quiz questions and their solutions. Question 13 asks for the host-to-host layers in the OSI model, with the solution "Transport, Session, Presentation, Application". Question 14 asks for the responsibility of the network layer, with the solution "To route packets from one node to another on its way to the final destination."

Which layer in the OSI model are host to host layers? This you know transport, session, presentation, application, these are host to host or end to end layers. What is the

responsible of the network layer in the OSI model to route packets from one node to another. This is the main responsibility on its way to the final destination.

(Refer Slide Time: 57:23)



The slide is titled "Quiz Solutions on Lecture 2" and features a logo in the top left corner. It contains two quiz questions and their solutions. Question 15 asks about the responsibility of the data link layer, and question 16 asks about the difference between a bridge and a router. The solutions are provided in bold text.

Quiz Solutions on Lecture 2

15. What is the responsibility of the data link layer in the OSI model?

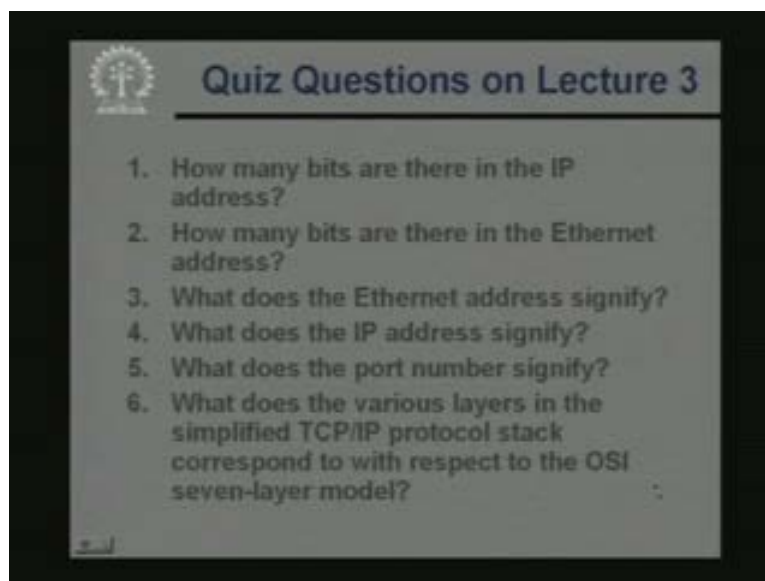
To ensure reliable data transmission over point-to-point links in units of frames.

16. What is the basic difference between a bridge and a router?

Bridge connects LANs, while router connects WANs (with LANs or WANs). Bridge works at datalink layer level, while router works at network layer level.

15. What is the responsibility of the data link layer? To ensure reliable data transmission over point to point links typically the unit of data transmission is frames. What is the basic difference between bridge and a router bridge will connect LANs. While router can connect WANs to LANs or other WANs, bridge works at the datalink layer level while router work at the network layer level.

(Refer Slide Time: 57:54)



The slide is titled "Quiz Questions on Lecture 3" and features a logo in the top left corner. It contains six quiz questions related to IP addresses, Ethernet addresses, and the TCP/IP protocol stack.

Quiz Questions on Lecture 3

1. How many bits are there in the IP address?
2. How many bits are there in the Ethernet address?
3. What does the Ethernet address signify?
4. What does the IP address signify?
5. What does the port number signify?
6. What does the various layers in the simplified TCP/IP protocol stack correspond to with respect to the OSI seven-layer model?

Now some questions from today's class is, in today's class how many bits are there in the IP address?

How many bits are there in the Ethernet address?

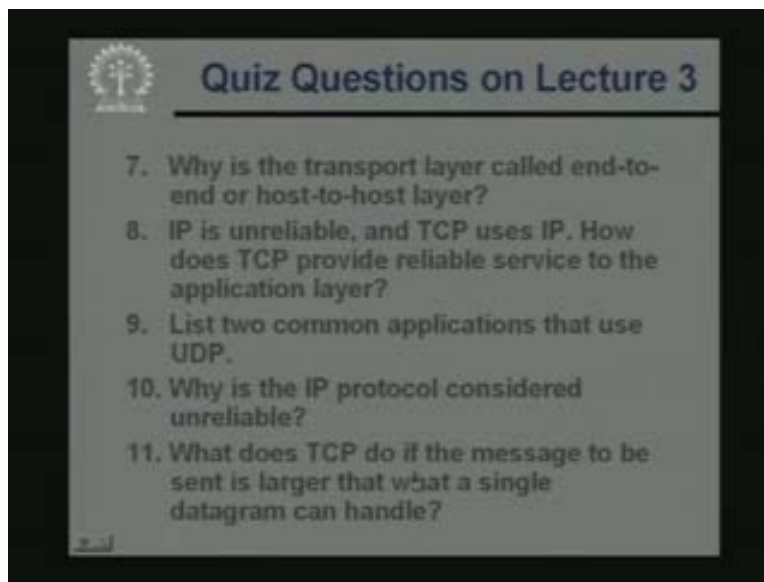
What does the Ethernet address signify?

What does the IP address signify?

What does the port number signify?

What does the various layers in the simplified TCP IP protocol stack correspond to with respect to the OSI seven layer model?

(Refer Slide Time: 58:23)



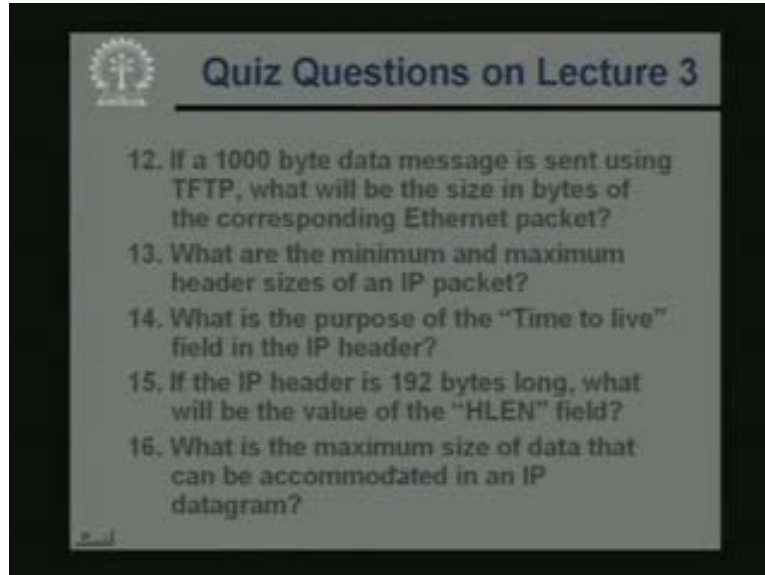
Why is the transport layer called end to end or host to host layer?

IP is unreliable and TCP uses IP how does TCP provides reliable service to the application layer.

List two common applications that use UDP why is the IP protocol considered unreliable.

What does TCP do if the message to be sent is larger than what a single datagram can handle?

(Refer Slide Time: 58.54)



If a 1000 byte data message is sent using TFTP. What will be the size in bytes of the corresponding Ethernet packet?

I mean after encapsulation what are the minimum and maximum header sizes of an IP packet?

What is the purpose of the time to live field in the IP header?

If the IP header is 192 bytes long, what will be the value of the HLEN field?

What is the maximum size of data that can be accommodated in an IP datagram. So with this we come to the end of today's lecture. Thank you.