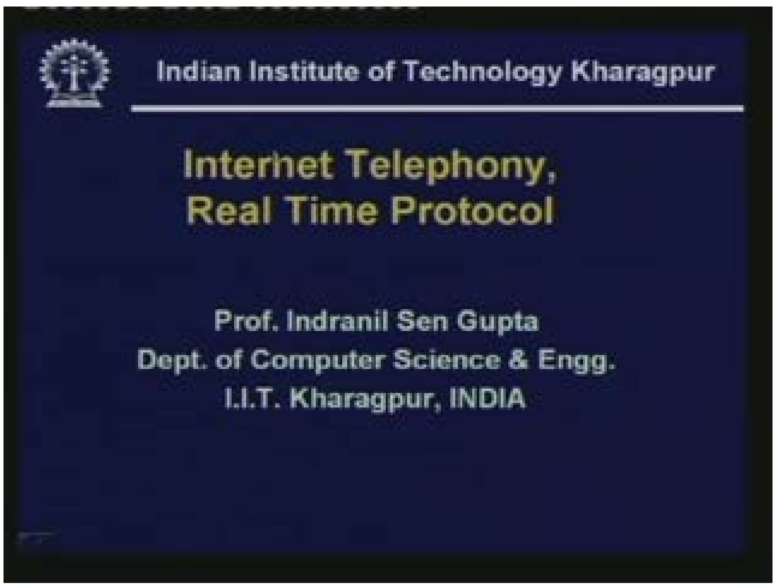
Internet Technology
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur
Lecture No #37
Internet Telephony

You know last lecture we were talking about multimedia networking, its basic requirements and problems the issues involved. We talked about the stream applications that we see in the internet scenario so widely nowadays. So in this lecture we shall be continuing from where we left in the last lecture. We shall be talking about some other important multimedia applications that we have in the internet scenario namely IP telephony or voice over IP. And then we shall we talking about some of the related protocols and standards which are used for the purpose.

(Refer Slide Time: 01:36)



So the topic of our discussion today is internet telephony some real time protocols.

(Refer Slide Time: 01:42)

So we start with internet telephony.

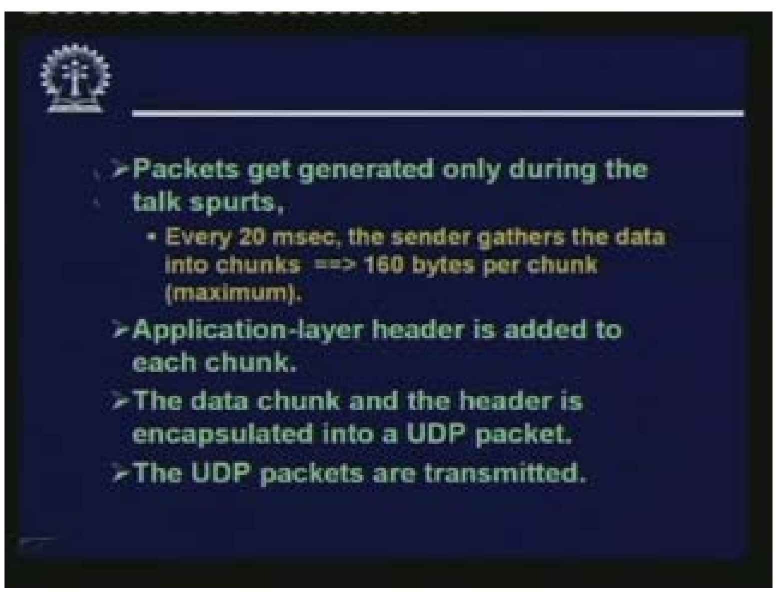
(Refer Slide Time: 01:47)

So internet telephony which is also called voice over IP is one of the most important applications of interactive multimedia over internet. This is called interactive because two parties sitting on two different computers over the internet. They talk or chat among themselves using the microphone and the speakers which are connected to their PCs. This is interactive because of the reason I mentioned in the last class. If I am one of the parties in the communication involved I will talk for some time then I will wait for the person at the other end to give a reply following which I will again talk. So this is some kind of interactive scenario and since it is interactive delay is or the response time becomes very important. After I finish my part of the talk if for instance there is a one second gap before I can hear the party at the other end, it will not be very good for me typically.

These kinds of large delays are not tolerable in these kinds of interactive applications. So basically this is voice chat over the internet. This works as I said the speaker at either end. Both the ends speak into the microphone which is connected to the PC and in turn when the other person talks the voice is heard on the speaker which is again connected to the PC. Now if you look at the speech pattern of a typical person when he or she talks on the telephone there is an alternate talk spurts and silent periods. This I have repeatedly mentioned. When I talk on the telephone I do not talk continuously I talk for certain period of time, then I give a pause and I listen to what the other person at the other end is saying. So if you think of the data or the data packets that are generated out of this voice signals, data packets may be not be generated and transmitted continuously.

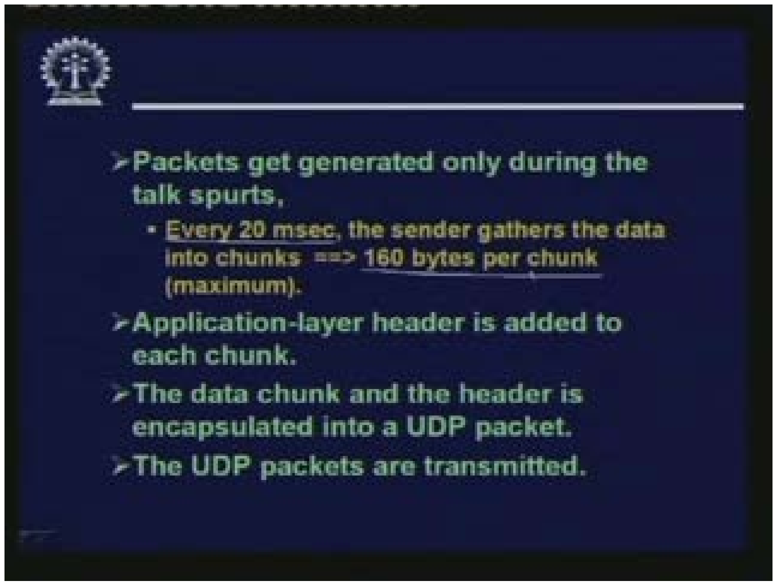
There will be certain period of time where I am speaking and the voice packets are getting generated. Similarly there are times when I am not speaking I am in the pause mode. There is no necessity for sending any packets in that time. No voice packets are generated. So there is an alternate between you can say spurts of packets and pause durations. So here we call it alternating talk spurts and silent periods. Typically for decent quality of audio we transmit 8 kilobytes per second. That data rate during the talk spurts which means when we are talking the data that are generated they need to be transmitted at the rate of 8000 bytes per second which comes to a bandwidth of 64 kilobits per second.

(Refer Slide Time: 05:12)



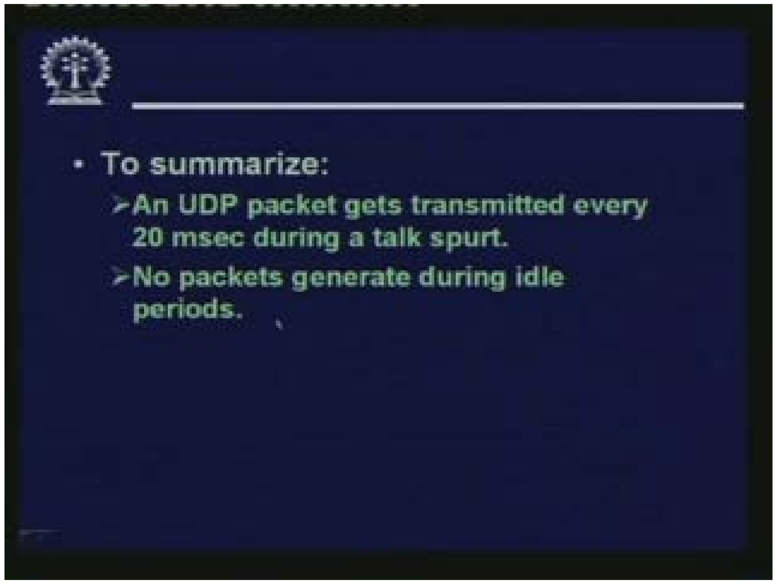
As I mentioned, the data packets that corresponds to the voice needs to be generated only during the talk spurts. The way typical voice over IP systems work is that every 20 milliseconds, the sender will gather all the data packets that are being generated in the 20 millisecond time into junks. Now if you calculate 8000 kilo bytes per second, then in 20 milliseconds you will be generating 160 bytes of data. So every 20 milliseconds if you frame a packet it will be of size 160 bytes. This is how the source works. The source continuously collects or digitalizes the data that corresponds to voice signal after suitable encoding depending on the data rate. Here I have assumed 8 kilo bytes. So in 20 milliseconds you will accumulate 160 bytes. You make it into a packet and then send it.

(Refer Slide Time: 06:17)



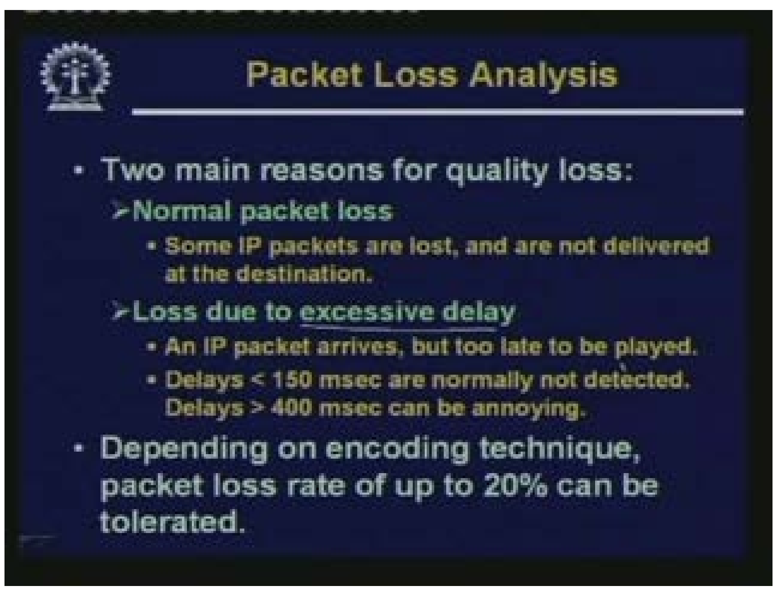
So maximum 160 bytes per junk depending on the period you are talking. This needs to be sent every 20 millisecond. This is the bandwidth requirement and after these 160 bytes of junk is extracted application layer headers are added to it. This application layer header can contain a number of things like the sequence number, for instance and some other information also. The whole thing, the data junk along with the header they are encapsulated into an UDP packet which are then transmitted. So the point to notice is that here typically we use UDP, because we do not want the unpredictable packet delays that we have encountered in TCP to take care of the packet losses and the other errors in transmission. We use UDP here.

(Refer Slide Time: 07:15)



So basically when you are transmitting voice over a network and UDP packet gets transmitted every 20 milliseconds during a talk spurt. But when you are not talking, when you are in the idle mode, no packets are generated. So the 8 kilo bytes per second bandwidth that I have talked about, that is the bandwidth required to be available. But it is not that you are continuously using that bandwidth. This is that.

(Refer Slide Time: 07:54)

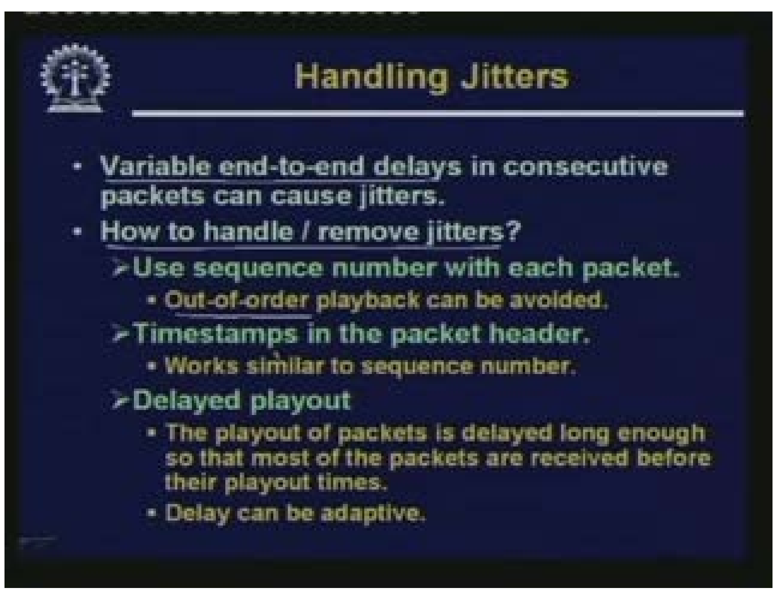


Now let us do a quick loss quick analysis of the packet losses that are encountered when we are transmitting this kind of voice packets. Now typically you may find that when you run a voice over IP application or over the internet. Sometimes the quality of voice is not good. There is break in the voices some parts in the voice is not legible. There are two basic reasons for this one is due to the normal packet loss as happens in the internet some IP packets are lost and are not delivered at the destination. Now as I said since we are using UDP even if a packet is lost the protocol does not try to recover from this error. If a packet is lost, it is lost. It will not be delivered at the destination.

The second reason in this kind of application is there can be some packet loss due to excessive delay. See here the issue is that the packet arrives but arrives too late while with each packet you can assign a maximum time at which it needs to be played. Now, say a packet arrives which must be played with in a time limit of say time t1. Now you find that the packet has arrived at your node, but the time t1 has already elapsed. So it is better to drop that packet rather than try playing it in a delayed mode. This is the basic idea if a packet comes which is excessively delayed simply drop it or if a packet comes out of order there is no point in playing it again just drop it.

And in a typical voice application, if such delays are less than 150 milliseconds. They are normally dropped, not detected easily. If you here the played back voice in your ear but larger delays may be creating problem. For instance greater than 400 milliseconds may be unacceptable. This can be very much annoying. There can be long gaps or breaks in the voice sometimes you may not be able to understand what the other person is saying. But in general depending on of course the type of encoding technique you are using packet loss rates of up to 20 percent can be tolerated even if loss up to 20 percent occurs. The voice you receive is of a sufficiently acceptable quality. You will be able to understand the content of the message.

(Refer Slide Time: 10:50)

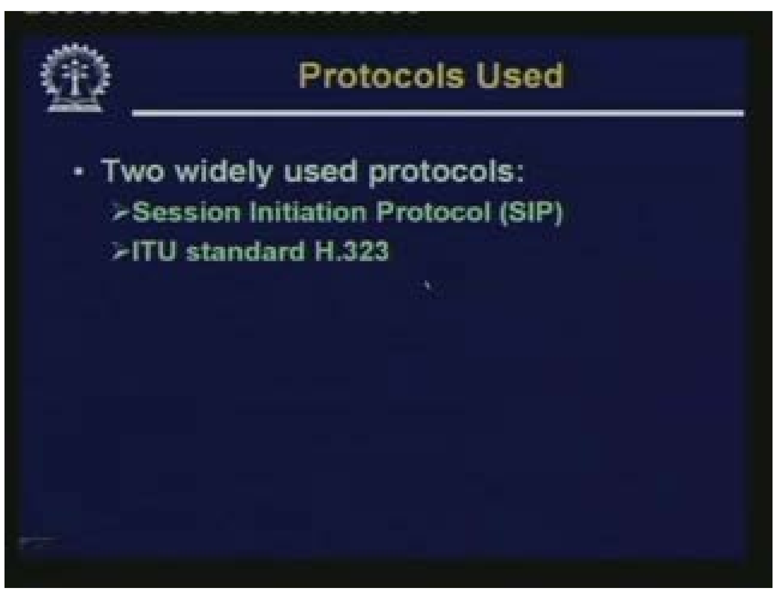


Jitters is another issue which we had mentioned in our last lecture, also jitters if you recall are caused by variable end to end packet delays in consecutive packets. So if a set of packets are coming which corresponds to a particular stream of voice and if there is variable delay between the consecutive packets they will not come to my note and when I play back it will not be smooth to my ears, there will be gaps, gaps will be variable, so these are called jitters. Now in voice over IP applications there are some mechanisms to handle or wherever possible to remove jitters. First approach is to use either sequence number or time stamps along with all the OS packets.

Now if you use sequence number with each packet you will get an idea that in which order the packets are expected if you find packet is arriving out of order. You do not play it back just drop it. Similarly instead of sequence number you can also put some kind of time stamp in packet header which is in some way a measure of the time. And the clock somehow you generate a time you put that stamp on the packet. And the receiver will check the relative orderings of the times, if a packet comes later than a packet which is already being played; you simply drop it. So sequence number or the time stamp both work equally well so you can use either of them. Time stamps also help in one thing it can give you some idea of the exact delay that the packets are encountered in transit.

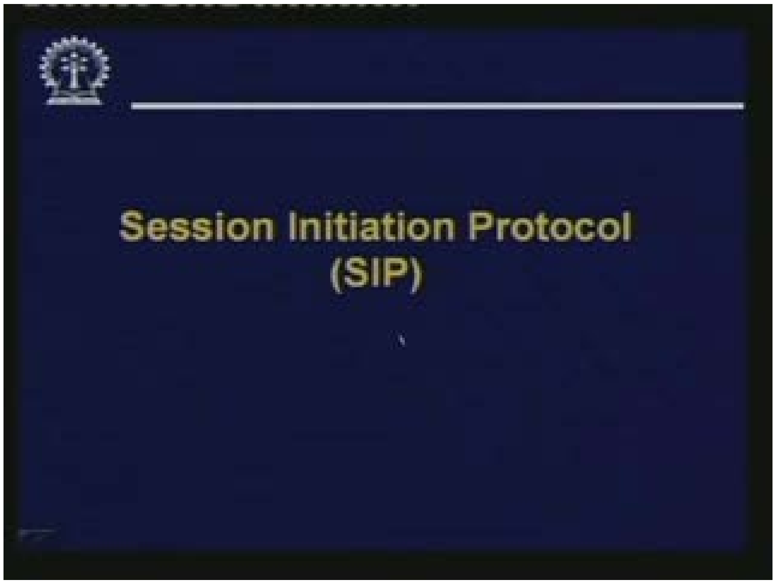
But normally a sequence number will not give you that information it will only give you some information about the ordering of the packets. And the there is another method called delayed play out. Delayed play out means you do not play the packets as soon as they receive; you try to delay them long enough. So that most of the packets are received before that play out times and then you start playing. So here delay can be adaptive depending on the play out times the tolerance you have. So you can try to minimize jitter by pushing the packets as much as possible back in time and try playing it at the latest possible time. So that if there is any jitter hopefully the packet has already arrived by that time.

(Refer Slide Time: 13:43)



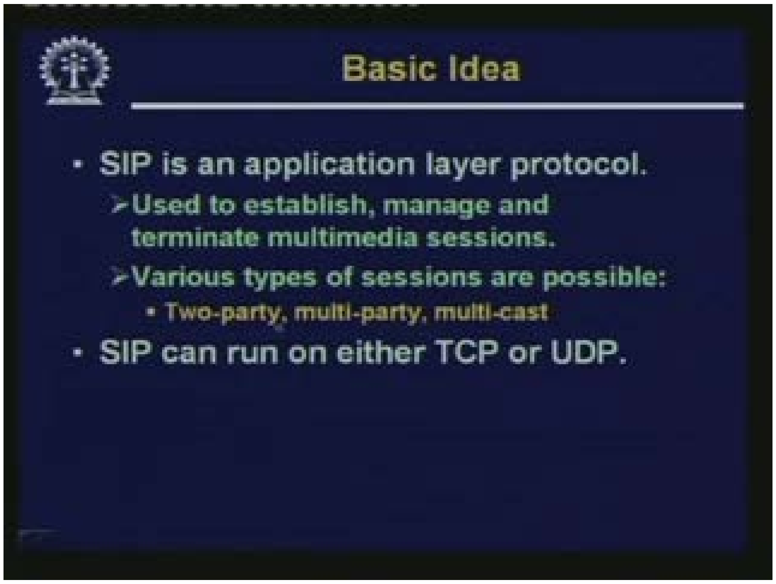
Now in these kinds of applications there are two protocols which are used quite widely. One is called the session initiation protocol or SIP and the other is a standard which is proposed by ITU this is called H.323. Let us look at the SIP protocol first.

(Refer Slide Time: 14:05)



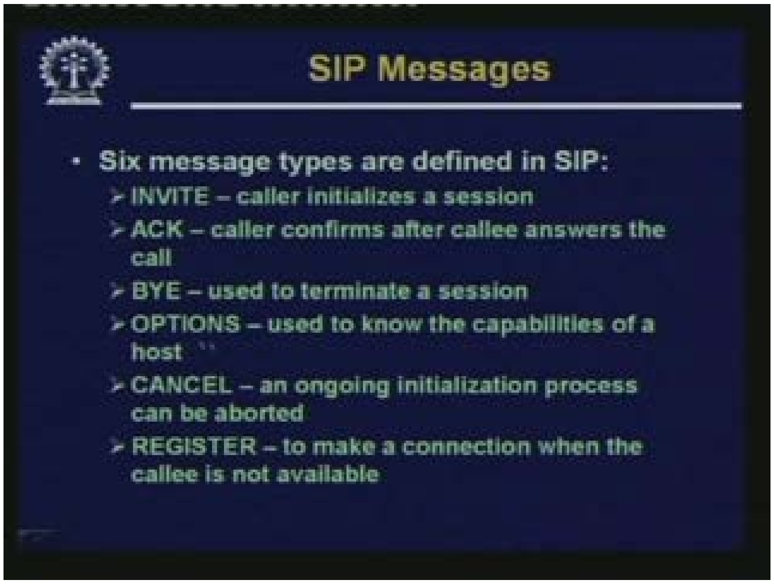
Session initiation protocol.

(Refer Slide Time: 14:11)



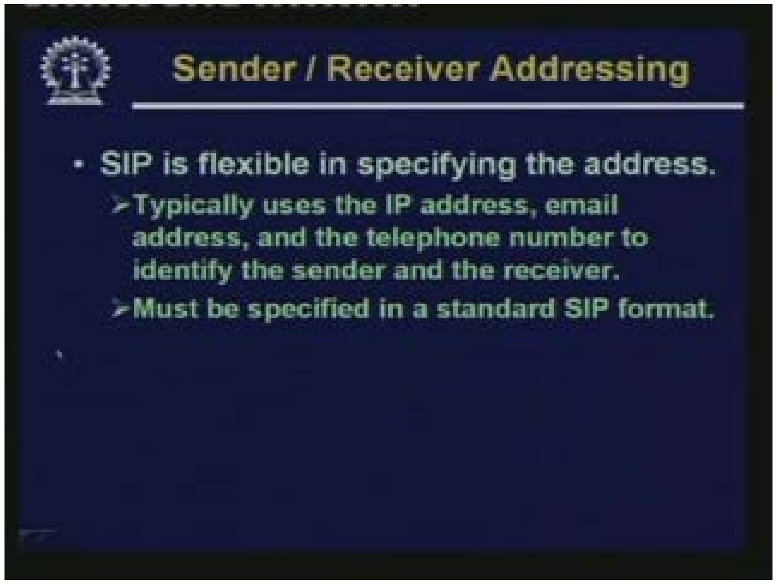
Now as is true for almost all multimedia applications, SIP is an application layer protocol. This is used to establish manage and terminate multimedia sessions. This SIP supports various types of sessions to cater to the wide variety of applications that run on that internet. You can have a two party chat, multi-party conferencing. Multi cast means some kind of news broadcast one person speaking many persons listening. So two party, multi-party, multi cast, depending on the requirement of the application you can have any of these supported by SIP. And the second thing is to notice that SIP is adaptive in terms of its underlying transport layer it can run on top of TCP or UDP depending on the quality of the network on top of which you are trying to run, you can choose to use TCP or UDP.

(Refer Slide Time: 15:13)



Talking of the SIP messages, there are 6 types of messages which are defined. There is an INVITE message where a caller tries to connect to the callee to initialize a session. There is an acknowledgment message where the caller after getting back an okay signal from the callee; the caller confirms that the session is on. So this ACK will be sent after caller gets a confirmation from the callee by a message used to terminate a session. OPTIONS this message type is used to know the capabilities of a host during the initial connection establishment this auction can be negotiated. CANCEL, an ongoing initialization process can be aborted. Suppose you are negotiating the OPTIONS, you can just give a CANCEL go back to the data transmission mode and of course there is a message called REGISTER where you can make a connection even if the callee is not available right at that point in time. So these are the 6 message types that are supported in SIP.

(Refer Slide Time: 16:39)



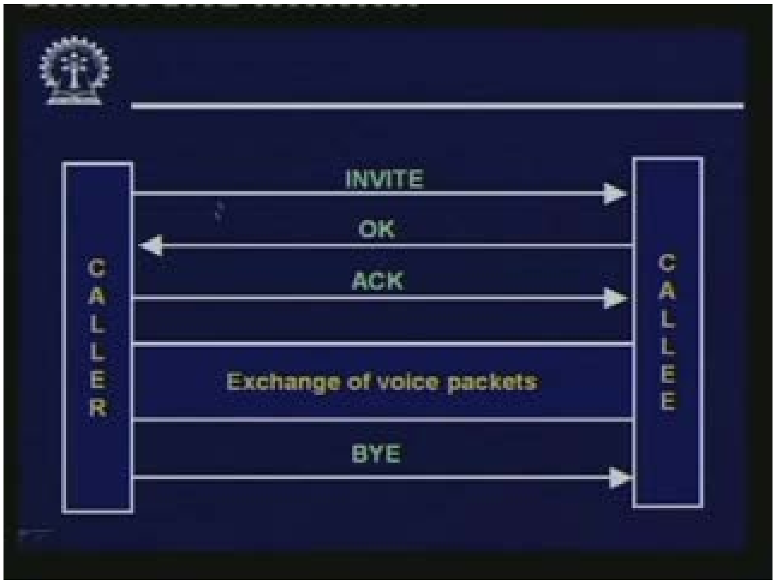
Just like when you are making a telephone callee you need to type in the telephone number. Similarly for SIP when you are trying to contact the person at the other side there must be some mechanism to specify the address. Now SIP specifies a well-defined addressing format. But does not specify exactly what are the components you need to put in which would form the address. That is up to the up to the end users to decide. Typically the address will consist of the IP address of the receiving machine, the email address of the person you want to contact and also the telephone number.

So these kinds of information are typically used just for the identification of the sender and the receiver. It is not that this email address and the telephone number is actually used for the communication. These are used only for the sake or the purpose of identification. And as I said there is a standard SIP format to specify the address. Now if you want you can put some additional components also in the address part. Some secret password can also be there. So it is up to you the kind of application you are trying to run over SIP. So, but whatever you put must be in the standard SIP address format.

(Refer Slide Time: 18:05)

Now a simple SIP session where a caller is trying to call up a callee and initiate a conversation will consist of three parts. First is to establish a session. Now this we shall see, for this we require or need a 3 way handshake protocol. Secondly we have the communication phase where the actual voice packets are sent to and fro. For this purpose 2 temporary ports are used and thirdly the session is terminated where either of the parties can initiate this process. These are the 3 parts to an SIP session.

(Refer Slide Time: 18:56)

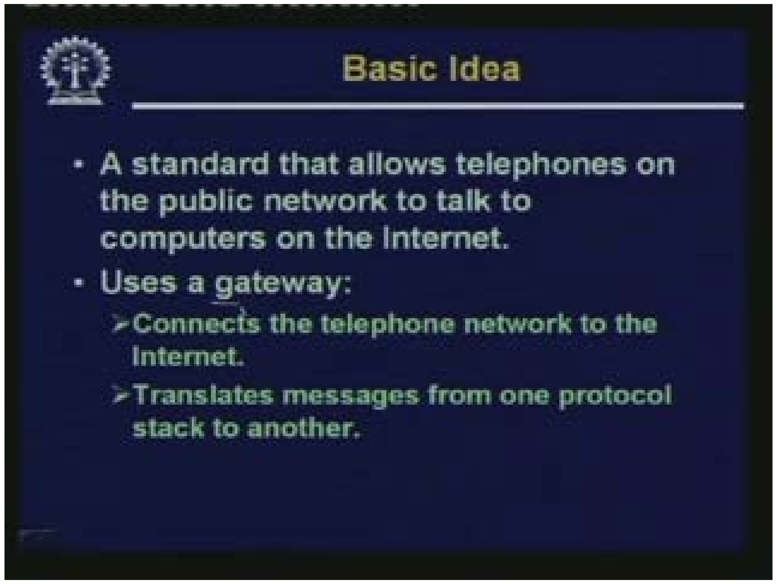


Now this diagram actually shows the different steps. First out here the 3 way handshake is there where the caller sends an INVITE message the callee sends back an okay acknowledgment. Caller sends an ACK at this point the connection is established. Then out here the voice packets are exchanged in both the directions. Finally when the call is over any of the parties can send a BYE message to break or terminate the connection. So you can see this SIP protocol is designed specifically for IP telephony or voice over IP application. So the kind of messages the kind of services they are precisely suited for the communication of voices interactive voice.

(Refer Slide Time: 19:58)

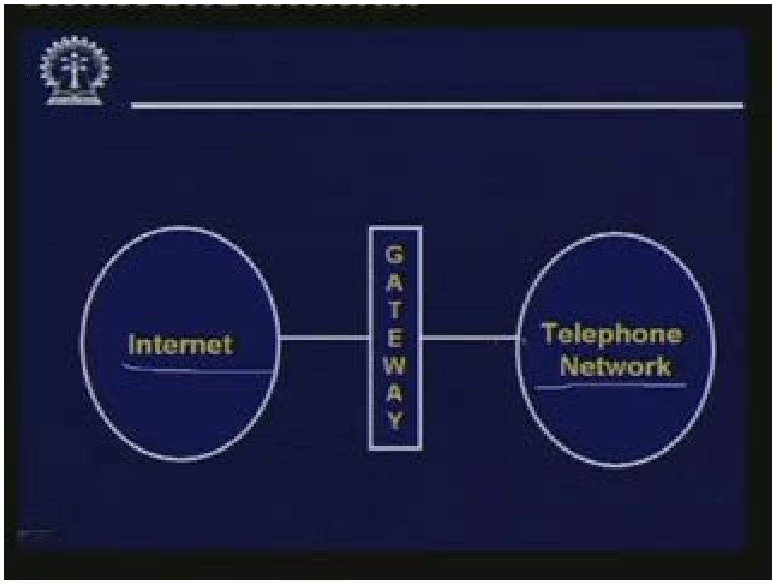
The next standard we talked about was the H.323 standard. This is slightly different in terms of standardization. This is a standard which allows or telephone on the public network to talk to computers on the internet. Now in contrast in SIP two computers connected to the internet was able to communicate among themselves. But now I have a standard landline set. From the landline I want to connect to a person who is sitting on a computer for that I need to use this protocol.

(Refer Slide Time: 20:42)



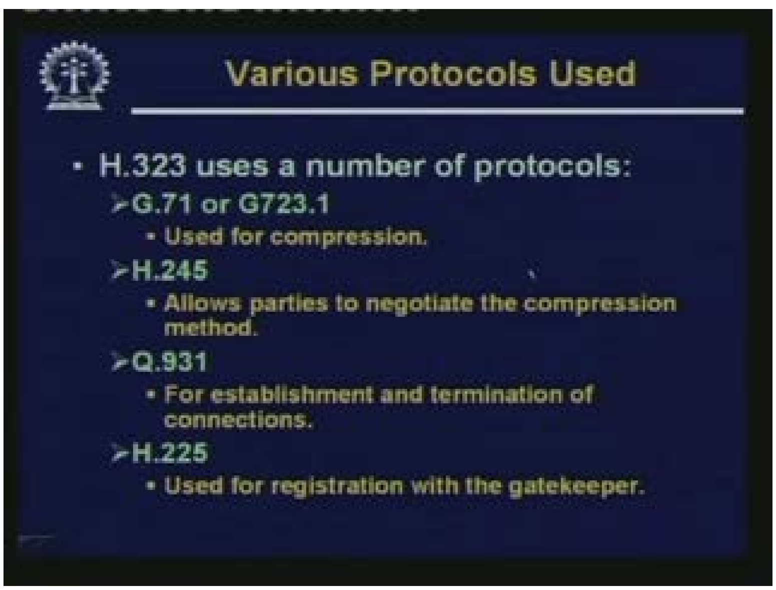
So here the concept is that you require a gateway in your network. Suppose in your network there are many users who are sitting on the computers want to make voice calls want to receive voice calls there will be a gateway on your network. Also there will be something like a gatekeeper which will be providing some kind of synchronization connecting the person who is making to trying to make a connection. Connecting to the gateway helping them in completing the connection and so on. We will see the steps. So a gateway is used in conjunction with a gatekeeper and this is used to connect the telephone network to the internet and the gateway carries out some message translation. Because inside when the voice packets are generated they may be generated using some particular kind of protocol using some minimalistic kind of headers. But when you are trying to send those packets to the outside world then you need to encapsulate them into proper IP packets say using UDP or TCP whatever. So that translation of the packets needs to be done at the gateway level. So the H.323 protocol specifies or supports the use of this kind of gateways.

(Refer Slide Time: 22:13)



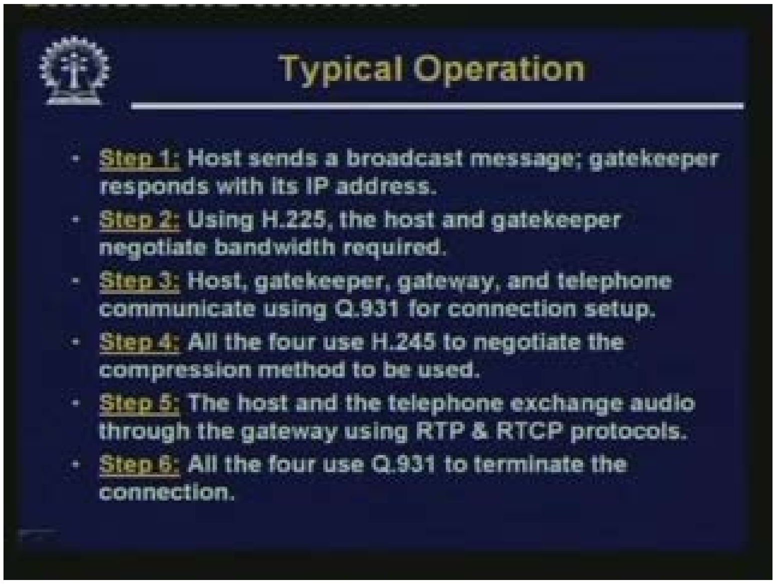
Pictorially it looks like this. You have the internet on one side you have the public telephone network on the other side. If you have a gateway in between you can make calls between these two communities of users. The gateway typically is located at the sight of the user premise where the organization network is present.

(Refer Slide Time: 22:41)



Now H.323 is not a single protocol in fact it consist of a set of protocols as this slide shows. First voice is compressed before they are packetized and for compression. There are two alternate protocols you can use either G.71 or G723.1. There is a protocol using which the parties can negotiate exactly which compression method is used. That protocol is H.245. This is the negotiation protocol for connection establishment and termination you use Q.931. And for you need another protocol H.225.

(Refer Slide Time: 23:37)



Now talking about a typical session out here so a typical session will look like this. First step the host wants to make a connection who wants to make a connection will send a broadcast message. The gatekeeper sitting on the same LAN same network will respond back with its own IP address. Then using the H.225 protocol the host and the gate keeper will negotiate the bandwidth required. So as a user I may want to have a high quality voice chat or I may agree to have a low quality chat. So the bandwidth requirement is negotiated to myself as the caller and gatekeeper who is sitting on my network. Then host gatekeeper gateway and the telephone.

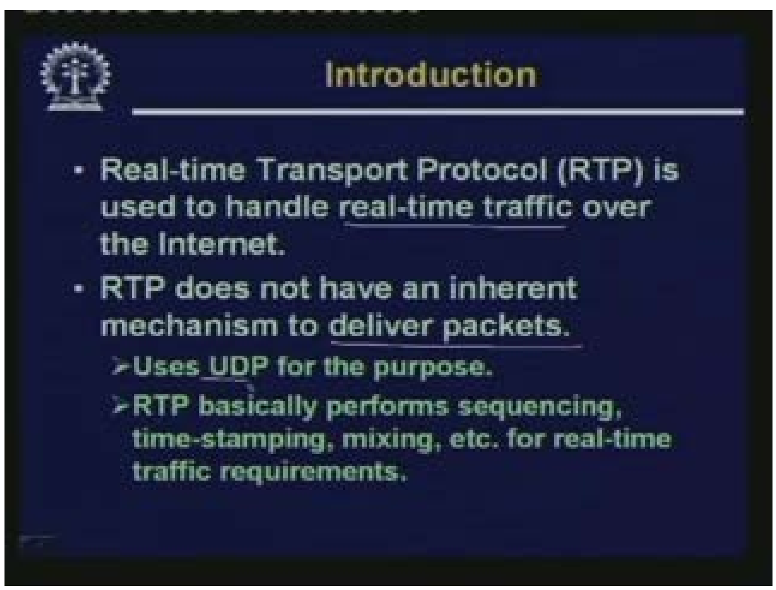
So now some kind of communication is established they are or is being tried to establish. There are four parties involved. The host is there gatekeeper is there gateway is there and the destination telephone is there. Now they communicate using the Q.931 protocol for the actual connection setup. Moreover all the 4, they use H.245. To negotiate the compression method to be used. Because most of the telephone sets today they support compression methods. So you need to specify what compression method is supported and to be used. Then in step 5 actual audio messages are sent back and forth. For this we use RTP and RTCP protocols.

We will talk about this a little later. Real time protocol and real time control protocol, these are used for sending audio packets back and forth between the host and the telephone through the gateway. And finally again all the four parties involved they use Q.931 to finally terminate the connection. So this is how typically this H.323 protocol works in the internet scenario when you want to make a seamless connection between a public telephone network and a private internet. So the presence of the gateway and the gatekeeper helps in the connection and maintaining the link between the two ends until or unless someone wants to terminate the connection. Now as I said for the actual transmission of the data you need some protocol called RTP or RTCP.

(Refer Slide Time: 26:27)

Let us now look at the basic purpose or the basic services such protocols offer. Let us look at the real time protocol or RTP first.

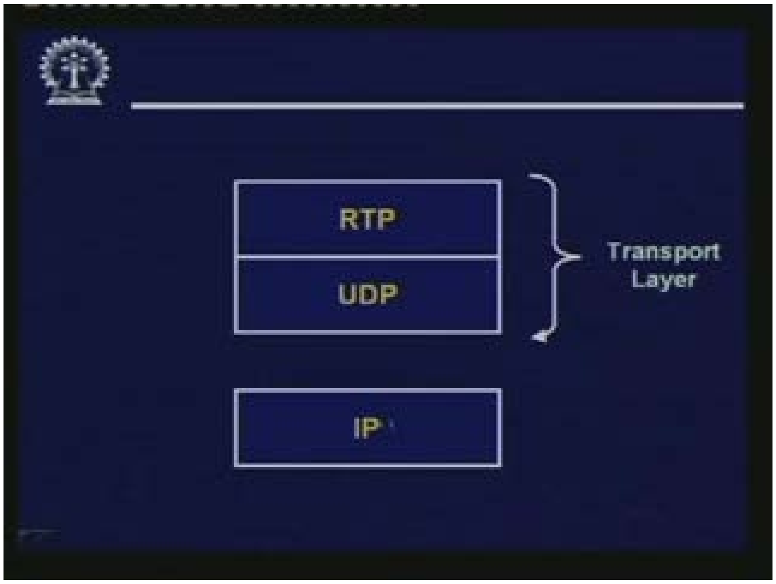
(Refer Slide Time: 26:39)



Now real time protocol or real time transport protocol. This is an application level protocol; it is running on top of UDP. As the name implies this is used to handle real time traffic over the internet. This RTP is an application level protocol. This does not have an inherent mechanism to deliver packets. It is simply just adds some headers sends some control messages at the application layer level. But at the lower layer it relies on UDP for the actual delivery of the packets. So it is the UDP layer which will actually be delivering the voice packets and RTP something which will be sitting on top of UDP.

But the question arises if UDP finally is sending the packets, what is RTP protocol is doing? So RTP protocol is responsible for a few things it can perform the sequencing like the receiving RTP can check the sequence number in the packets or the time stamps mixing. If there are several sources to be mixing them in a proper way and there were some other things also like if you want to move backward forward due to some reason. So whatever real time traffic requirements are there. So RTP tries to provide some service at the application layer level for that.

(Refer Slide Time: 28:19)

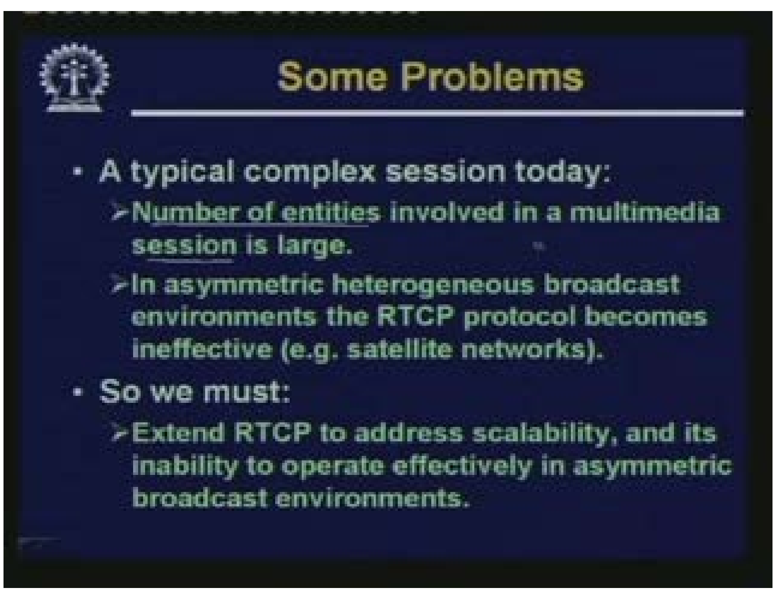


As I said pictorially it looks like this. At the lower level you have IP, on top of it you have the UDP which actually delivers the packets and RTP sitting on top of it. So RTP sometimes instead of treating it as an application layer protocol. This RTP and UDP in a combined way you can treat it as a transport layer protocol for real time audio traffic. So applications will be invoking the RTP layer from top. But actually technically speaking RTP is an application layer protocol which is running on top of UDP. So a typical multimedia session will look like this.

(Refer Slide Time: 29:06)

You relay on real time transport protocol when you are actually transmitting or receiving the voice data packets. You relay on real time transport control protocol. There is an auxiliary protocol RTCP. These are used for transmitting control information. RTP is used for transmitting data RTCP is used for transmitting control information. The situation is quite similar to the case where you have the FTP kind of protocols. Where the data and the control are being transmitted over two separate channels. But of course in FTP we have a single protocol, but here the protocols are also separate. The protocol for sending the control packets. The protocols for sending the data packets, they are separate.

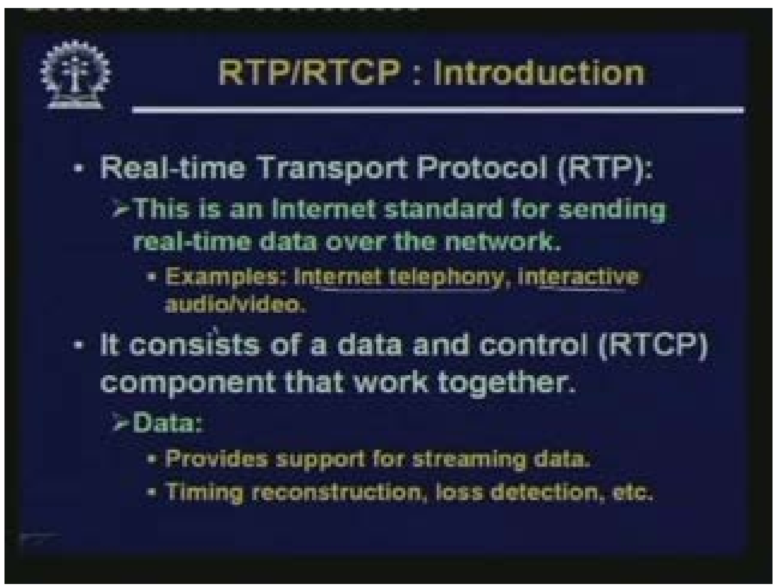
(Refer Slide Time: 30:02)



Now some problems that you may face you look at a typical complex session that we have today where there are a number of entities that are involved in a multimedia session. We have audio, we have video, we have other kinds; we have a number of different kinds of media. So if you are using RTP for transmission of voice, so we need to make some kind of synchronization. If you look at asymmetric heterogeneous environments like the satellite networks here the RTCP protocols becomes ineffective because the real time protocol is an interactive protocol. The satellite round trip propagation delay is pretty large about a quarter of a second round to propagation delay you need to encounter.

So for sending and receiving the control messages itself will take you about half a second. So the real time requirements will not be met. So these are some issues which need to think of when you talk about the environment under which you are trying to run RTCP. So actually what we need is that we need to extend the RTCP protocol to address this scalability issue and to address its inability to operate effectively in asymmetric broadcast environment. Just the example I have shown. So you need to make some modifications to the basic RTCP protocol so that you can handle these kinds of environments and scenarios.

(Refer Slide Time: 31:45)

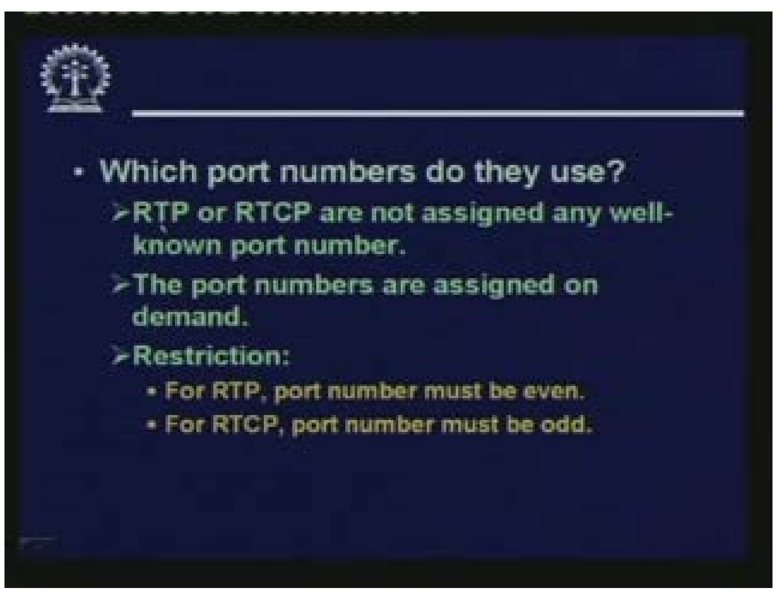


So now let us look at RTP and RTCP in some detail. So RTP as I said this is an internet standard for sending real time data packets over the internet typically. Examples are internet telephony but in general you can also transmit interactive audio or video. So RTP is general in that sense. So you may think that it is primarily suited for internet telephone and voice. But you can also use it for transmission of video. As I said it uses a data component and a control component which is guided by the RTCP protocol. For the data component you have a support for sending and receiving streaming data packets. It supports some timing reconstruction loss detection some packet loss deliberate packet loss delayed playback. All these things whatever you can the RTP protocol will try to provide that. So that the jitter the other packet loss related drawbacks are minimized.

(Refer Slide Time: 33:07)

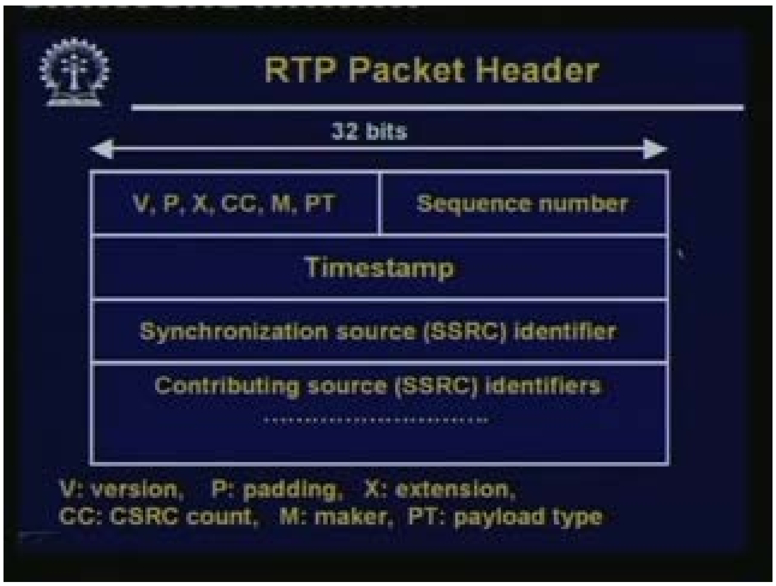
And the RTCP control protocol. This is the control part of the protocol it provides the following functions. It monitors whether data are delivered correctly or not. It identifies the source of the packets. It allows the members that are involved in a session to calculate the rate to send status messages. These are important in establishment and maintenance of a session once it is setup. So the control protocol silently monitors the progress of the session. And wherever it finds that some changes need to made it. It initiates the process so that some parameters are modified or adapted depending on the changing scenario in the network.

(Refer Slide Time: 34:01)



Regarding port numbers I have said that there are two channels; one for RTP and one for RTCP. So naturally they require two different port numbers. But these protocols do not use any well-defined port numbers. They are assigned on demand with only restriction that for RTP the port numbers that are assigned must be even, for RTCP the port numbers are assigned must be odd. This is just a convention which is followed. There is no very strict reason for this. If you look at the port number you will be knowing that whether it is a data port or a control port. This convention is followed in the RTP and RTCP protocols.

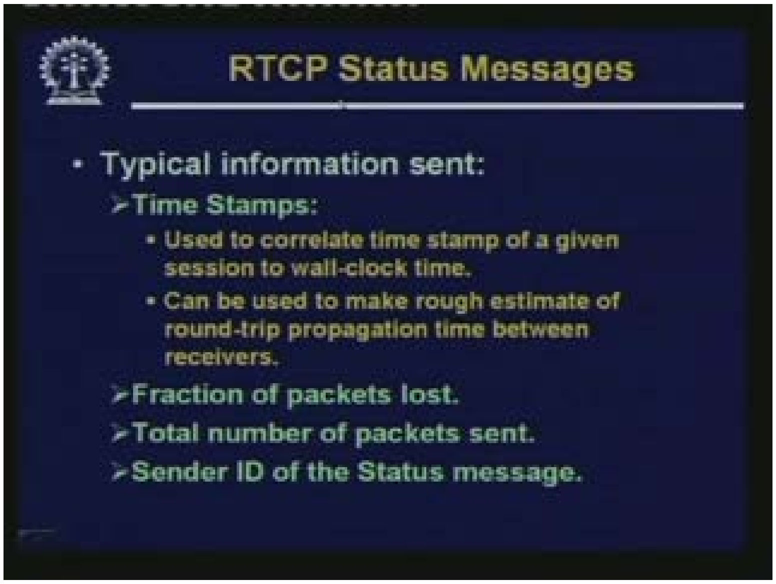
(Refer Slide Time: 34:49)



Here is a quick view of an RTP packet how an RTP packet looks like that. Means I have shown the header part only. See at the beginning you have some flags I am not trying to explain everything. There are some flags called versions some padding extension etcetera some payload type. There are a few kinds of information in the header which may be needed by the other parties when the connections are negotiated. There is a field for sequence number there is a field for time stamp. If you want you can fill in any one of them or if you want you can use both.

Now in addition to identify the sources there are two fields; synchronization source and contributing source identifiers. There are used basically for the purpose of identification of the source. Then you have the actual data. So this is how a typical RTP packet looks like. In addition to the data there is sufficient information in the header which will enable the receiver to identify the source and also will get some additional information to the receiver. Actually what is the payload type one? What kind of encoding has been done? What is the data rate this kind of things are present in the header? So all this information you can get.

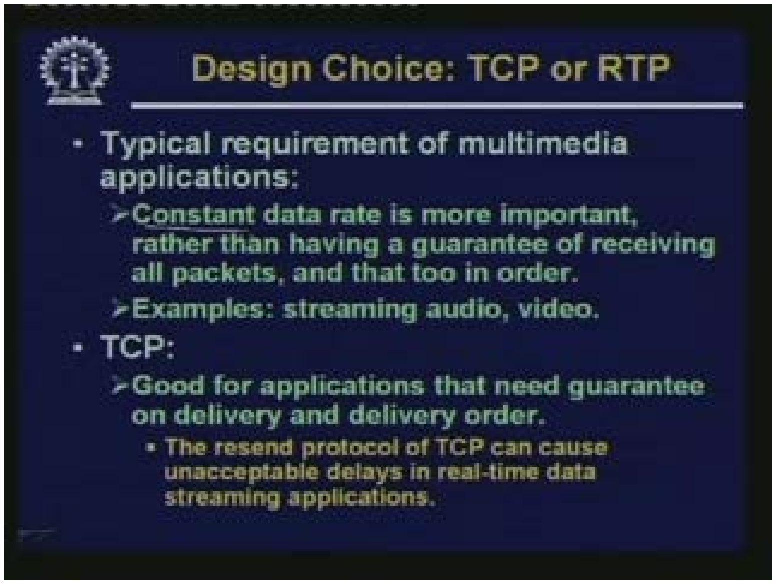
(Refer Slide Time: 36:19)



And RTCP status messages as I said RTCP this control protocol is used typically to send and receive some status messages between the parties which are involved in the communication. Now in the status messages you typically send the time stamps. Time stamps are used to correlate the time stamp of a given session to a wall clock time can be used to make rough estimation of the round trip propagation time between receivers. Here this control messages all carry time stamps. They are some measure of the wall clock time as I said. So these control messages will give some information to all the parties concerned that what are the present average delays on the links.

And how much time the packets are taking from reaching from one node to the other. So this gives some idea to the sender and the receiver so that they can adapt they are playback environment according to the changing scenario. So these can be used as I said to make rough estimates of the round trip propagation time. These status messages also carry information about the fraction of packets that are getting lost. Total numbers of packets that have been sent and of course the sender ID the person who is sending the status message. So any node can send the status message to the other parties and depending on this message other parties can update their information.

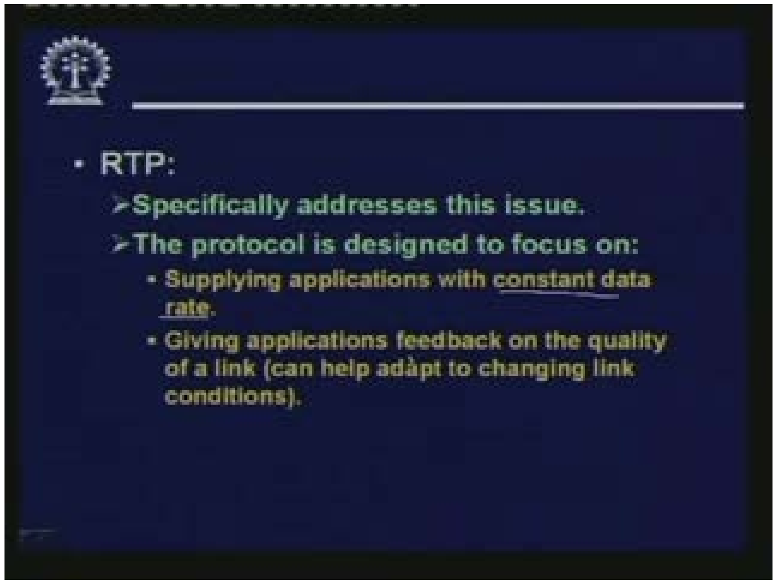
(Refer Slide Time: 38:14)



No, here there is an issue that when you are trying to use these kind of application voice over IP or any other interactive multimedia. You have a choice whether to use RTP or whether you use the TCP which is the standardly available to you. So there are some trades off here typical requirements of multimedia applications are constant data rate guarantee of receiving all packets correctly in order. That is not so important. What you need is that whatever is coming to me should come at a constant rate I do not care if some packets in between are corrupted or lost. But whatever comes to me should come at a regular rate. So that playback is smooth to my ears.

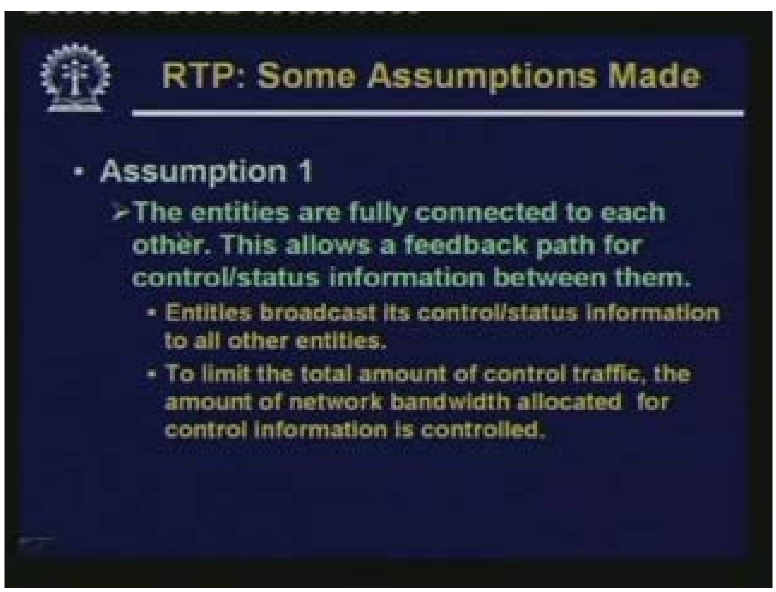
This is more important requirement for multimedia applications. So some examples we have seen streaming audio video. So if you think of using TCP, well TCP is good for applications that need guarantee of packets and the order of the delivery. But for real time applications I mentioned earlier also TCP may not be a good idea because TCP resends a packet if it is not received correctly at the destination. This resend protocol can cause unacceptable large delays in real time data so that you may experience very unwanted delays and jitters during the playback. So TCP due to this fault tolerant feature is not suitable in real time applications.

(Refer Slide Time: 40:00)



But RTP as I have said it specifically addresses this issue. The protocol is designed to focus on this important requirement constant data rate. And it also continuously gives the application feedback on the quality of links using the RTCP protocol which can help the nodes adapt to changing link conditions. For example if the link is very bad, then I can down group. I can basically downgrade my audio quality to a low quality audio whose bandwidth requirement will be less. So it is not advisable to use a very high quality audio encoding for a link which is of a poor quality. Because anyway the playback quality will be horribly bad in that case.

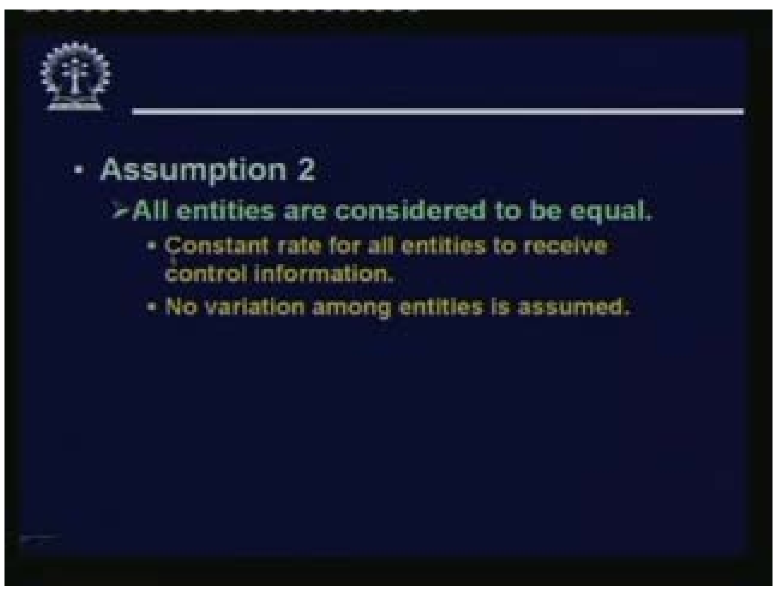
(Refer Slide Time: 41:55)



Now in RTP, now we make some assumptions. First assumption is that the entities are fully connected to each other because you sent this RTCP messages in a broadcast mode. This will allow a feedback path for the control messages using RTCP to flow between them. So actually this is the requirement of a broadcast. Entities will broadcast its controller status information to all other entities. So there is the entities are fully connected to each other because you sent this RTCP messages in a broadcast mode. This will allow a feedback path for the control messages using RTCP to flow between them. So actually this is the requirement of a broadcast entities will broadcast its controller status information to all other entities. So there is a significant amount of control traffic and in order to limit the total amount of bandwidth that is consumed by this kind of control traffic. You may want to limit the amount of network bandwidth that you want to allocate for this purpose.

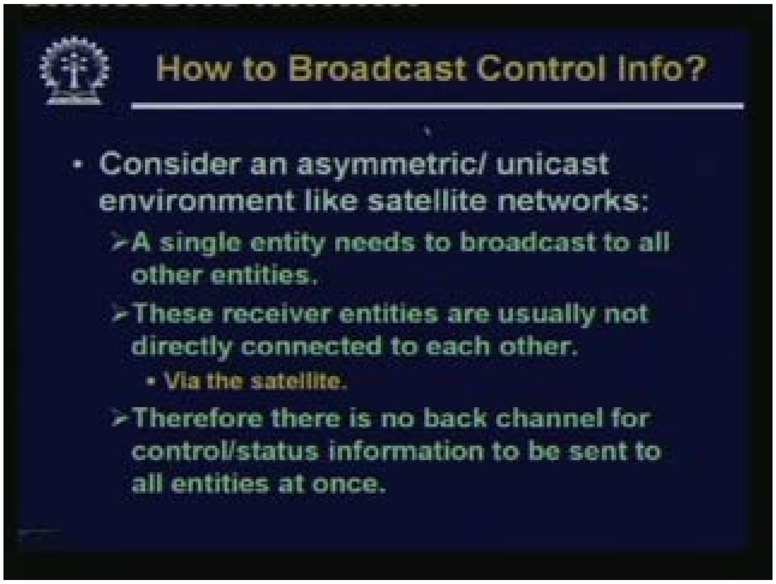
Here what I mean to say is that the RTCP protocol is used by the nodes to broadcast some status information to all other nodes. Now you can limit uprightly that this is the maximum amount of bandwidth I can provide to the different nodes for transmission and broadcast of these status messages. So in this way the transmission of these messages will not disturb the quality of the actual voice or the other kind of media that you are transmitting.

(Refer Slide Time: 42:23)



The second assumption is that all entities are considered to be equal. There is no master slave relationship between the entities. So it is constant rate for all entities not that some entity will be having privilege and getting a higher data rate as compared to other entities. So there is no variation among the entities.

(Refer Slide Time: 42:48)

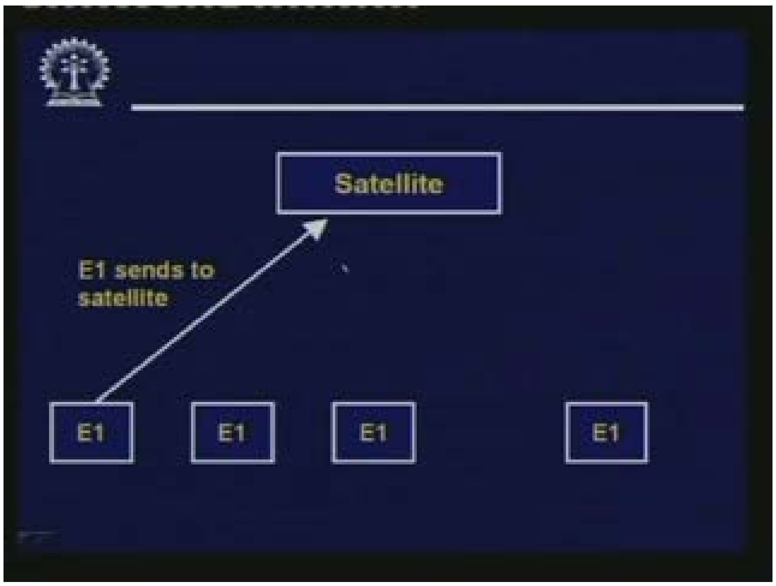


Now the issue is that how do we broadcast this control information. Now you consider an asymmetric unicast environment like the satellite networks. Now in a satellite networks suppose we consider this situation where a single entity needs to broadcast to all other entities. Now the receiver entities are usually not directed to each other. Rather they are connected via the satellite. So there is no dedicated channel or back channel though which the control status information can be sent directly to all the entities at once. They have to be sent via the satellite.

(Refer Slide Time: 43:36)

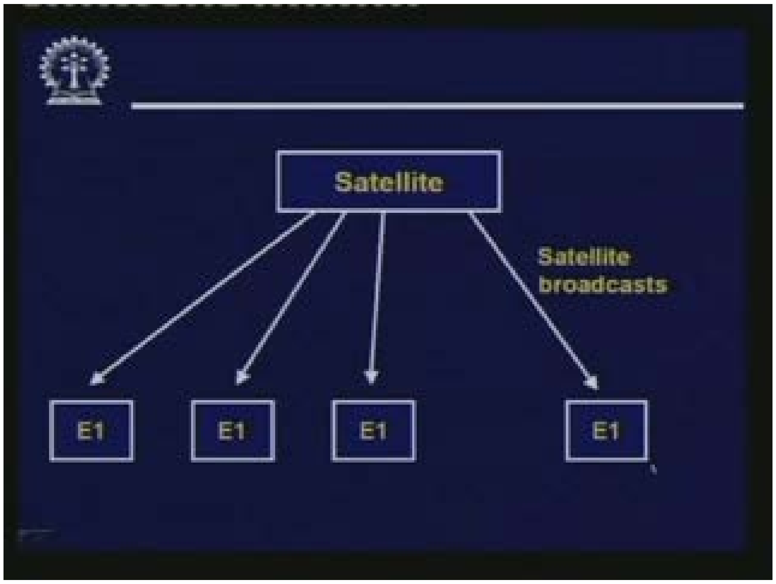
So we can use the satellite for the broadcast. The satellite actually works as a signal repeater sitting in the sky. So here basically what we are talking about is instead of having an entity broadcast to all other entities. We are doing it indirectly. The entity is sending the information to be broadcast to the source which is the satellite and it is the satellite which will be broadcasting to everyone else. It is a process called reflection. So the sender need not have the capability of broadcasting rather it is sending to an entity which is reflecting it to everyone else. So this is how it works in a typical asymmetric broadcast scenario like this satellite.

(Refer Slide Time: 44:26)



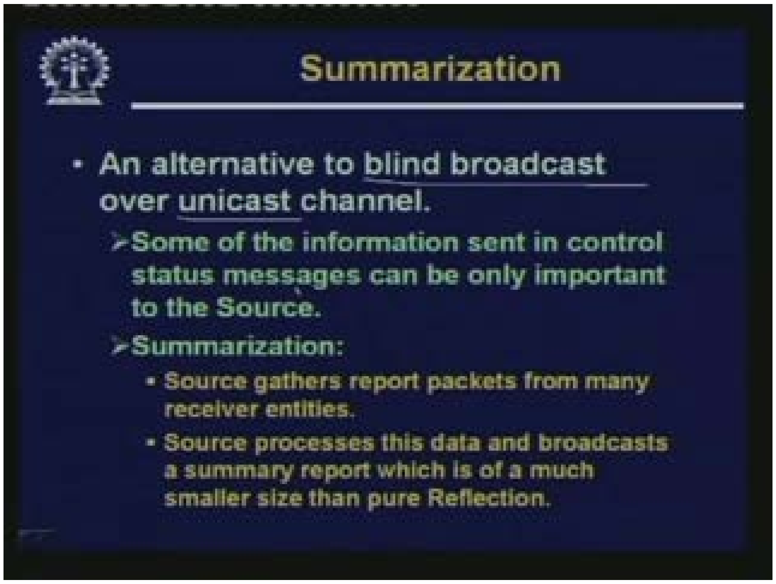
Diagrammatically let us try to illustrate it. Suppose we have a satellite out here and there are some entities out here. Some entity sends a control message to the satellite.

(Refer Slide Time: 44:41)



The satellite broadcast it back to everyone else. So this is how it works.

(Refer Slide Time: 44:48)

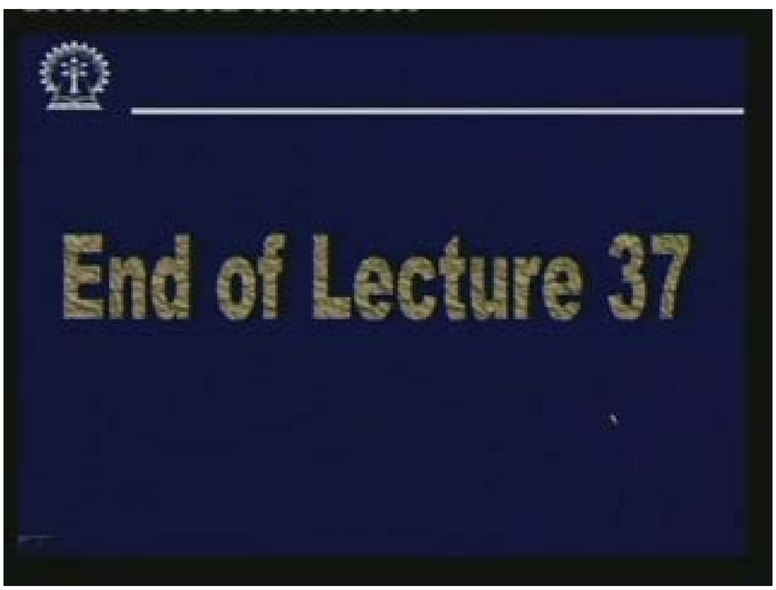


Now when you talk about broadcasting there is another issue. See suppose I am a node which is involved in communication using RTP or RTCP. So I have a lot of status information along with me though I broadcast everything to everyone else or it is good. If I prepare some kind of a summary information and only send this summary. Because summary typically will be of a much smaller size and it will consume much less bandwidth in the process. So the process of compacting the total amount of information available to me in the form of a summary and sending it to others is called summarization.

So summarization is an alternate technique to blind broadcast. So the earlier method we said the nodes whatever information they have they blindly broadcast to everyone. And it was a unicast channel to this satellite and from satellite they are broadcast to everyone else. But here what we are talking about is that some of the information that you are sending to control status messages may not be that important to the other nodes. They may be relevant only to the source who is trying to send the broadcast. Packet summarization says that the source will gather report packets from many other receiver entities.

And it will carry out some kind of data processing and create a summary report. The summary report will be broadcast to everyone else which will consist of more you can say relevant information which the other nodes require. And of course it will be of a much smaller size because you are not sending the whole data rather you are sending some kind of average mean average delay total number of packets percentage of packets dropped this kind of thing instead of sending detail information about all the packets.

(Refer Slide Time: 46:57)

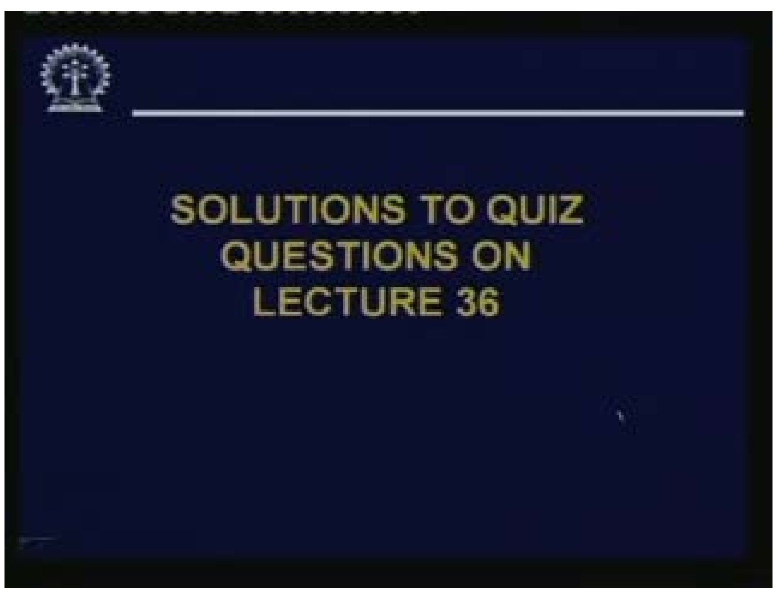


So with this we come to the end of today’s lecture. Now if you recall whatever we have talked about in today’s lecture and also the previous lecture we talked about some of the requirements of real time multimedia traffic in the internet scenario. We have said that more and more applications are emerging in the internet, which require this kind of applications. Well means you can also look at the futuristic applications which demand the same like video on demand. The day is not very far where most of us would like to watch video on demand sitting on our terminals. Now we can send a request for a particular video or any other kind of a media and that particular media will be played on my desktop in streaming mode typically. So the protocols that we use for the purpose has to gear up to have this kind of a support.

The other point to notice that I also mentioned it earlier that the present generation of the internet predominantly uses TCP and IP and the conventional routers which work on them for the transport of packets on the network. But if you look into the future this kind of applications will start emerging and the user will be requiring or demanding higher and higher quality. So there you should require or need an environment in which you can have some guaranteed quality or service parameter setup before the actual communication can start. For instance if I am downloading a video, well I may want that I need a guaranteed bandwidth of 2Mbps or 6Mbps or 8Mbps whatever I want setup between the two parties. But today's network will never give you such a guarantee. So this is one of the challenges.

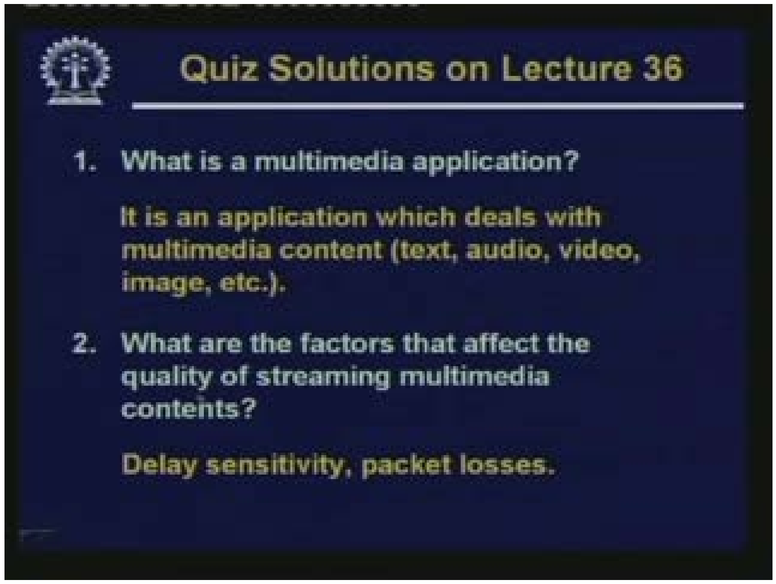
There are competing rivals to you can say TCP IP and other protocols like the ATM protocol the asynchronous transmission mode. The ATM protocol emerged with these kinds of applications in mind. So there is an inherent support of quality of service in the ATM protocol. Well ATM protocol is very good for these kinds of applications but unfortunately in the internet scenario you do not have an ATM back bone. The back bone that exists it is primarily an IP back bone. And IP back bone till today does not support quality of service or real time traffic requirements. So this is one challenge that is still facing us unless or until we move on to the next generation of the internet where these kind of support will be inherently present. We really cannot think of very high quality real time applications that we can you can say commercialize over the internet. Today we have services but there are limitations or constraints. So with this we come to the end of today's lecture.

(Refer Slide Time: 50:25)



Let us now look at some solutions to the questions we post in our last class.

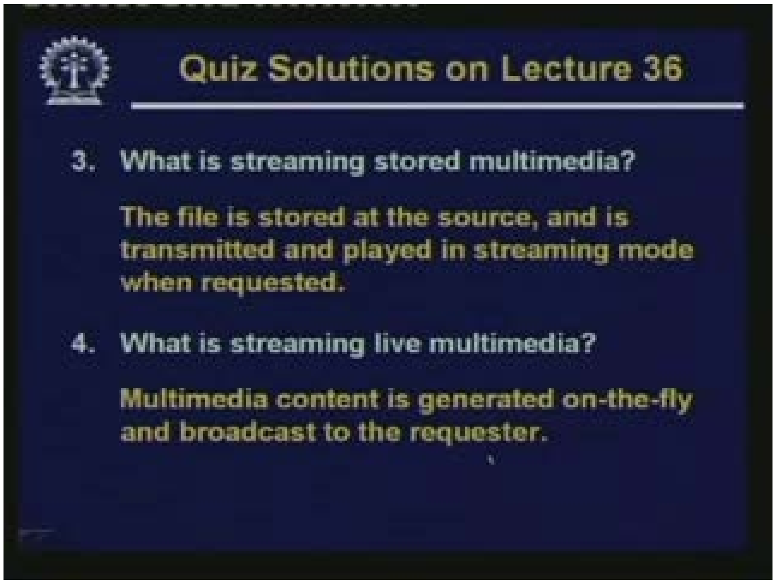
(Refer Slide Time: 50:35)



What is a multimedia application?
Now a multimedia application is one, we have repeatedly said which deals with multimedia content text, audio, video, image etcetera. Now again I have mentioned we view a look at websites today where in the web pages there are mixed contents audio, video. Now it depends that the video, suppose there is a place where a video clip is running on your page. Now it depends on the way the server is handling it whether you are downloading the whole clip and then playing or is it running in the streaming mode. There are technologies like flash they support this kind of streaming. So there are tools and technologies available where you can built a web page today which has support for streaming media. So when we develop some application you have to keep all these things in mind.

What are the factors that affect the quality of streaming multimedia contents?
Now broadly we said that there are two things one is delay sensitivity. Well if the delay is long but the delays of all packets are the same then it is okay if it is not real time traffic. But if the delay is variable then we can have jitters. Similarly if you have packet losses then it is better to drop the packets if the losses are not too many rather than try to recover from the lost packets and playing the correct stream. Because that can also introduce some unnecessary delays in the playback.

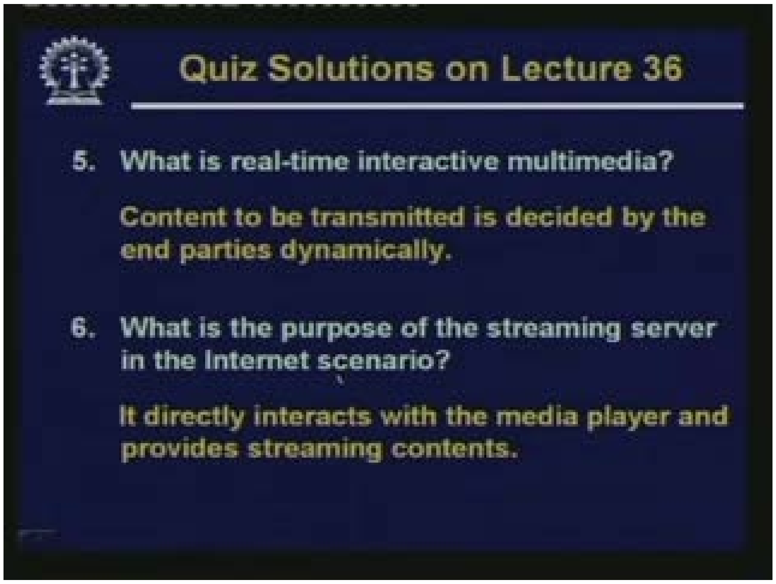
(Refer Slide Time: 52:27)



What is streaming stored multimedia?
Now this I mentioned here the file is stored at the source from the server from where you are downloading and is transmitted and played in streaming mode when requested. So we are not downloading the whole file and then starting to play. So as soon as the downloading starts the playback starts in the streaming mode.

What is streaming live multimedia?
Here the multimedia content is generated on the fly and broadcast to the requester. Well we had given a couple of examples like live news broadcast or a live sports even which is or a live concert which is being so called webcast. There is a term for it webcast. Webcast thing of an event which is occurring live. So it is a streaming live multimedia session.

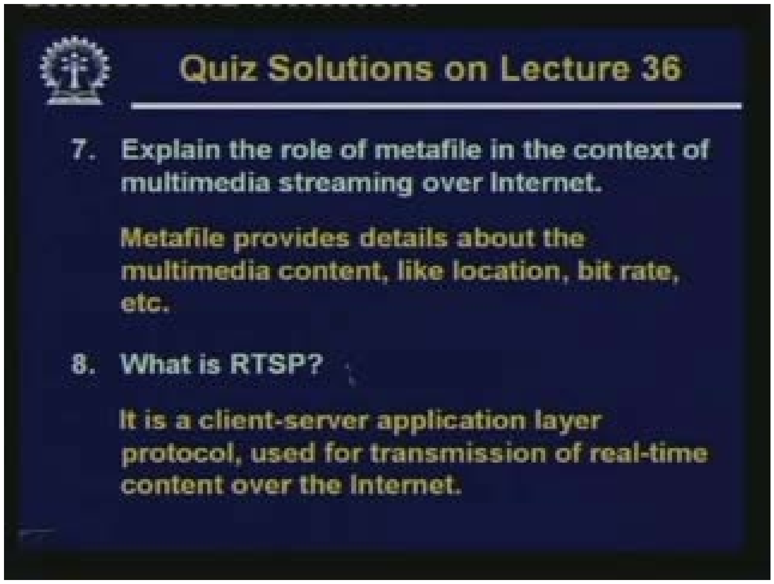
(Refer Slide Time: 53:30)



What is real time interactive multimedia?
Well voice over IP is an example of it the content to be transmitted is decided by the end parties and dynamically when to transmit, this is decided by the end parties.

What is the purpose of this streaming server in the internet scenario?
Now this streaming server directly interacts with the media player and provides streaming contents. Now instead of having streaming server you can have an alternate situation where this media files are stored on the same web server and they are requested using http. If you request the media files using http then you are forced to send over TCP and the properties of TCP that is mentioned, it tries to ensure the correctness if a packet is lost. If the packets are out of order it will wait till the correct packet comes back and then it will forward it to the application. But multimedia applications cannot afford to wait for the correct packet to arrive. So http access to the media files may not be a very good solution. That is why we use a separate streaming server where the media files are located. And the media files are accessed using some kind of a proprietary protocol like RTP or RTCP is example where the overheads of http and TCP are not there.

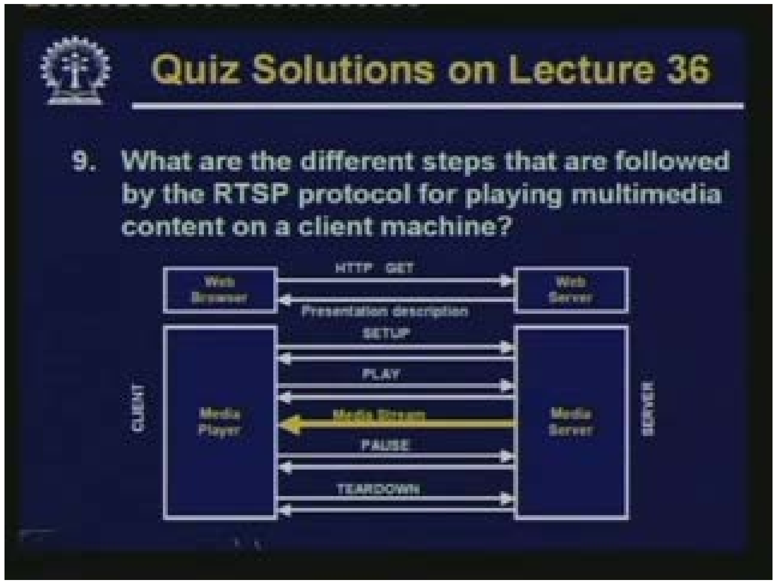
(Refer Slide Time: 55:14)



Explain the role of metafile in the context of multimedia streaming over internet.
Now as I mentioned metafile actually contains some additional information about the multimedia content like the metafile will contain information about the media whether it is an audio video or a combination of both. If there are multiple encoding streams available low quality, high quality, medium quality, it will contain details of everything depending on your available bandwidth. You can dynamically select one of them. So all these things are available in the metafile. So you can provide these things.

Then the next question is what is RTSP?
Real Time Streaming Protocol this we have said this is a client server application layer protocol used for transmission of real time contents over the internet. So RTSP supports the facilities that real time transmission demands and the different kinds of control signals that are available under RTSP are specifically suited to these kinds of requirements or needs.

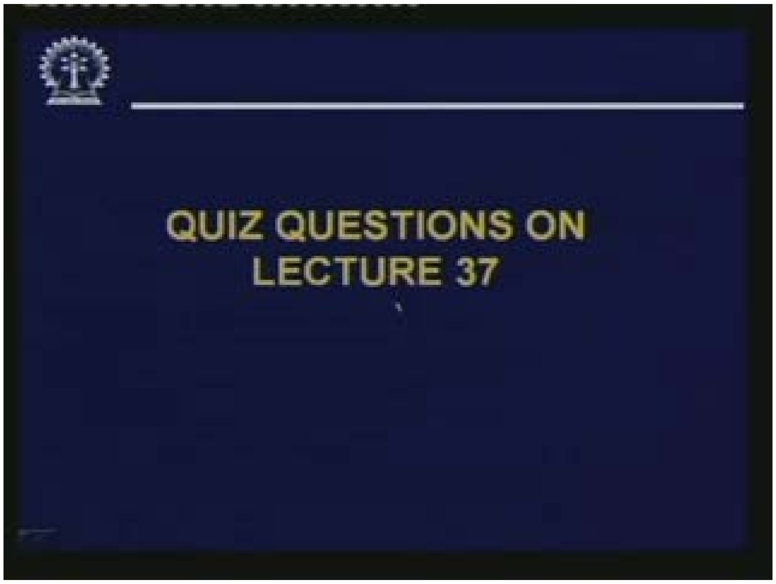
(Refer Slide Time: 56:40)



What are the different steps that are followed by the RTSP protocol for playing multimedia content on a client machine?

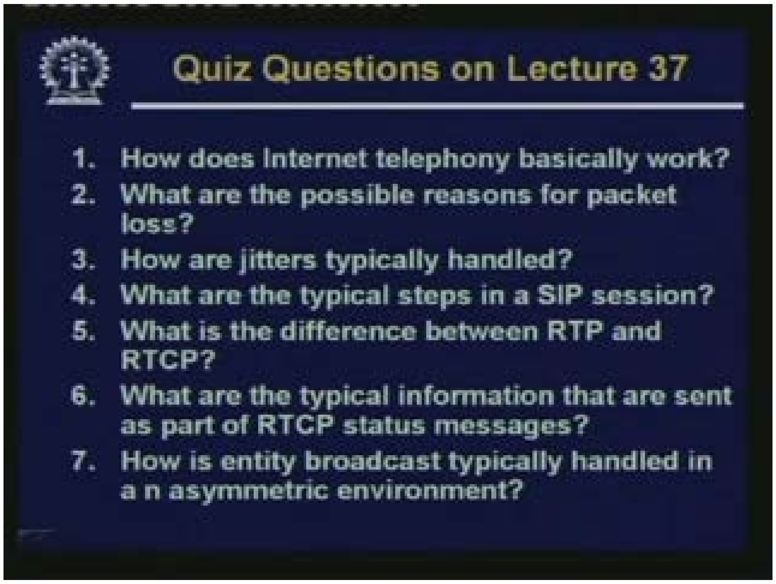
Well this diagram we had seen already on last class I am just quickly going through it once. In RTSP before actually starting a session the web browser sends a request to the web server and gets back the Meta file after getting back the Meta file the Meta file is sent to the media player. Now the media player after analyzing the Meta file decides to start interaction with the media server. It first carries out a set up phase where the connection is established then play. So after this play request and play acknowledge comes back the media stream is actually starting to flow. So now the playback has started. Now media player can send a pause request paused request can be used to stop a media to do a fast forward rewind. This kind of things so the media server can be informed what to do next which packet to send next. And finally when the session is over there can be a tear down message which can break the session or terminate the session.

(Refer Slide Time: 58:08)



So now let us look at some of the questions from today's lecture.

(Refer Slide Time: 58:14)



How does internet telephony basically work what are the possible reasons for packet loss?
How are jitters typically handled?
What are the typical steps in a SIP protocol or a SIP session?
What is the difference between RTP and RTCP?
What are the typical information that are sent as part of RTCP status messages?
How is entity broadcast typically handled in an asymmetric environment?

So with this we come to the end of today's lecture. In our next lecture we shall be starting our discussion on web chlolous search engines. These are again some very interesting application which run on the net and helps the general users in locating or finding the information they want very fast. So in our next class we shall see how this web chlolous and search engines really work internally. Till then good bye.