Broadband Networks

Prof. Karandikar

Department of Electrical Engineering

Indian Institute of Technology, Bombay

Lecture - 30

RTP

So, in the previous lectures we had discussed the issues involved in voice over IP. So, what we were saying is that the data plane operations that are involved in voice over IP is that first the analog speech is converted into digital speech and then audio coding is performed to achieve the necessary compression and after which the packetization is done and then a voice packet is sent using UDP mainly as the transport mechanism over the internet.

Now, we had seen that if since the internet is usually the best effort and does not offer any explicit quality of service guarantees, there could be problems in terms of either the packet loss or the end to end delay and the delay jitters as well. What we saw that some amount of the packet loss up to let us say 10 to 15% can be tolerated depending upon the applications and even if some packet loss is there, forward error correcting codes FEC can help conceal these losses.

Similarly, if the network is lightly loaded, then the delay will not be that significant and moreover if the delays are in the range of 150 milli seconds to 400 milli seconds, then the human ears may not perceive such delays. So, some delays may be tolerable also and may not degrade the quality of the speech to that extent. So, some end to end delays may not affect the performance of the voice over IP also.

However, since the different packets may experience different queuing delays at various nodes in the internet due to the random queuing delays which will be encountered at different nodes; it may so happen that when packets arrive at the receiver, since different packets have experienced different delays, the periodicity with which the packets were generated at the transmitter that periodicity may get lost at the receiver and even though the maximum delays suffered by any packet maybe less than the expectable tolerable delay. Even then, since different packets are experiencing different delays due to this delay jitter that is there in the stream of the packets, there may be an appreciable degradation in the speech quality.

So, the delay jitter may become an important factor in voice over IP protocols compared to packet losses and the absolute delay encountered by a packet. So, we saw in the previous lecture that the technique which is used to encounter or combat this delay jitter is to buffer the arriving packets at the receiver for an appropriate amount of time and then play these packets out at the receiver.

Now, how do you determine the playback time? There are various techniques that can be used to determine these play backs times. In the fixed play out buffer algorithms, what you do is that you delay the packet in the buffer for an amount of time such that the total delay experienced by this packet from the transmitter including this play out buffer delay is equal to the maximum delay that any packet will encounter. That means the play out time of the i'th packet is determined by adding to its time of generation the maximum delay. That is if t_i is the time of generation of the i'th packet and d_{max} is the maximum delay, then the play out time p_i will be t_i plus d_{max} . So, that is how the play back time of the i'th packet can be determined.

But as we saw that the problem with the fixed play out buffer algorithm is that you may have to know the maximum delay that any packet will encounter in the networks and if you do not know this; then one may take a conservative estimate and keep a large value for the value d_{max}. So, if you do this, then there will an appreciable latency in the conversation. So, this latency may become an important factor.

However, if you do not keep a conservative estimate and if you keep the value of d_{max} to be low, then it may just happen that occasionally some packets may experience large delays and they may arrive at the receiver even after their playback time has passed. So if that happens, then the packets will be lost and if there is no control on this packet loss, then there will be degradation in the speech quality. So to combat this, one may not have the fixed play out buffer but one may have the adaptive playback buffer algorithms.

So, what you do in the adaptive playback buffer algorithms? In the adaptive playback buffer algorithms, you actually adaptively adjust the amount of time for which the packet has to wait in the queue and this adjustment is done by estimating the average delays encountered by the packets in the internet and the variances of these delays and then appropriately adjusting the delay of the packet in the playback buffer. By doing this, by appropriately adjusting those delays, one can achieve a balance or trade off between the latency of the play out on the one hand and the packet losses that may occur due to the packets arriving late at the receiver on the other hand.

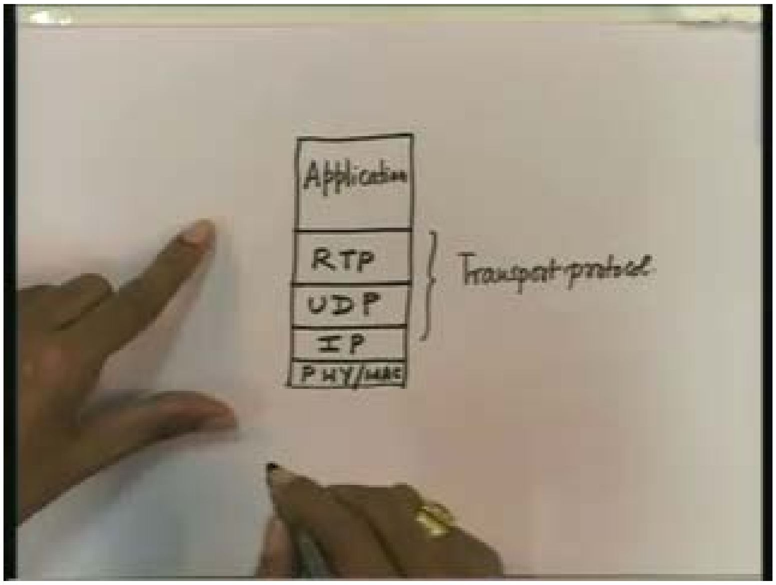
So, one can achieve this balance between these two by having appropriate version of the adaptive playback buffer algorithms. Several adaptive playback algorithms have been proposed in the literature and one of them we have already discussed which basically tries to estimate the average delay in the network by taking a moving average of the delays, a low pass smoother version of the delays and then adding and using this average delay to estimate actually the playback time.

Now, as we have seen since this delay jitter is an important problem; in order to estimate this playback time, it is necessary for the receiver to know the time of generation of the packets. So, because if you know the time of generation of packets and then if you know the time of reception of the packet at the receiver, then you can determine the delay of the packets and these delays then can be used to estimate the average delays in the network.

So basically, you need to know the, you need to timestamp these packets. So whenever, a packet is generated at the transmitter, you need to timestamp these packets. Now, the question really is that who does all this time stamping? Because, as we have seen that after the packetization, we had mentioned that since retransmission mechanism is not a possible option in the voice over IP for the real time services, TCP as the transport mechanism is ruled out and therefore the UDP is used. But UDP as the transport mechanism does not have any mechanism or any method of time stamping the packets.

So, this means you require a separate transport protocol for the purposes of the real time services over internet and that separate protocol is what is called the real time protocol or RTP. So, this is this is the RTP. So, we will discuss today some features of the RTP and how real time protocol would help achieve our goal. So, what are the features of the real time protocol or the RTP? Basically, if you see in the internet protocol stack, then RTP can be viewed as some kind of a transport protocol.

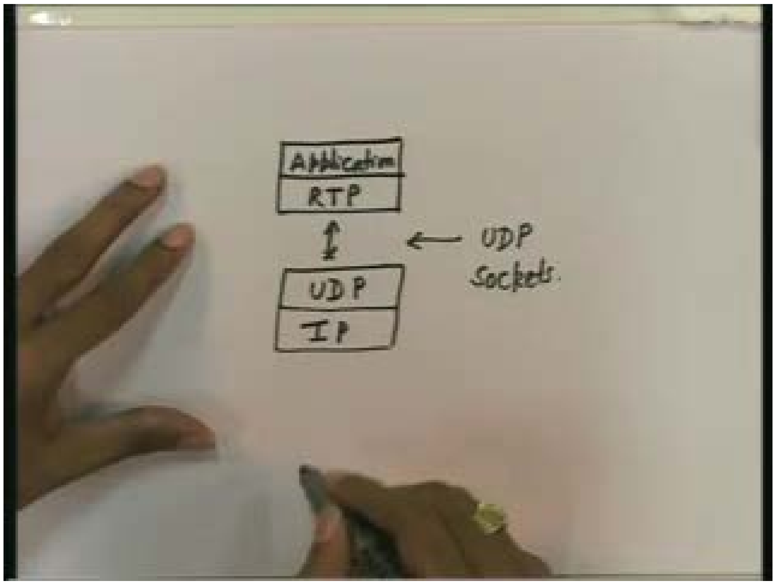
(Refer Slide Time: 9:26)



So, it would look like this that if you see that this is your physical layer and phy/Mac and this is your IP layer and this is your UDP layer and this is your RTP and this could be your application. So, as you can see here, both RTP and UDP, they are really the transport protocol and the primary objective of this transport protocol RTP is to provide time stamping of the packets and also to provide sequence number to the packets so that if any packet loss is there in the top spurts, then that can be detected.

So, if you view this in the internet protocol stack, then RTP is like a transport protocol. However, in most voice over IP telephones or in most voice over IP applications, RTP is typically bundled and used as a part of the application itself and uses UDP sockets to communicate with the rest of the protocols stack. So, if you take that approach as a programmers approach, then RTP can be viewed as a part of the application layer. If you take a pure network theorist approach, then RTP is more like a transport protocol.

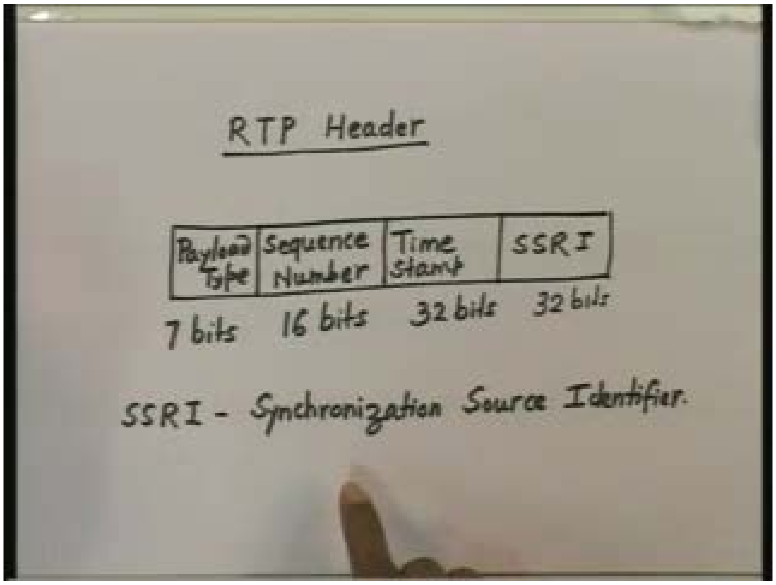
(Refer Slide Time: 11:12)



So, from a programmer’s point of view, RTP will appear something like this that you have a part of the application and RTP and this is used using UDP sockets to your UDP IP. So, this can be used using sort of UDP sockets. So, if you take this approach, then RTP acts like a voice over IP application itself and typically RTP will be a part of the data plane applications for the voice over IP.

Now, we will look at what is the RTP header and then from the RTP header, we will try to understand the functionality of the real time protocol or RTP. So, now let us look at what is there in RTP header.

(Refer Slide Time: 12:11)



RTP Header

Payload Type	Sequence Number	Time Stamp	SSRI
7 bits	16 bits	32 bits	32 bits

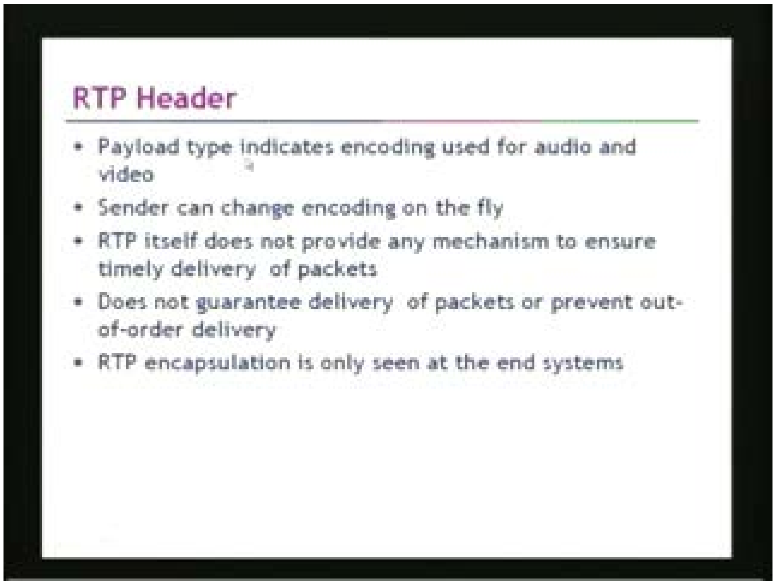
SSRI - Synchronization Source Identifier.

So, let me just have the RTP header which will actually tell you what is the functionality of the RTP header. So, there is a payload type, so this is typically 7 bits, then you have sequence number; so this is 16 bits, then you have time stamp, this is 32 bits and then you have SSRI which is 32 bits.

SSRI stands here for synchronization source identifier synchronization source identifier. So, this synchronization source identifier is different from the IP address. It actually identifies a particular RTP stream between two hosts. So, that is how it is actually get associated with a particular multimedia stream or RTP stream. So, it is distinct from the IP address and so sender actually chooses this SSRI - synchronization source identifier as a random number.

The other bits as you can see here that the payload types or the sequence number or the time stamps number, we will explain their functions shortly. So, let us look at the slide where it shows that what are the functions of the various RTP headers?

(Refer Slide Time: 14:06)



RTP Header

- Payload type indicates encoding used for audio and video
- Sender can change encoding on the fly
- RTP itself does not provide any mechanism to ensure timely delivery of packets
- Does not guarantee delivery of packets or prevent out-of-order delivery
- RTP encapsulation is only seen at the end systems

So, the payload type if you can see, the 7 bits of payload type, it indicates the encoding which is used for the audio and video. And this payload type is actually there are 7 bits for this. So, as you can see, since there are 7 bits for the payload types, 2 raised to power 7 that is about a 128; 128 types of payloads are possible.

Now, basically what is meant by these payloads? I mean it can be an audio packet or it can be a video packet. You can have different types of audio coding as we have seen and there can be a different type of video coding as well as mpeg 1 or mpeg 2 or mpeg 4 or H 263 kind of encodings, whether it is a PCM encoded audio or it is a DCPMN encoded audio or it is vocoder encoded audio and so on.

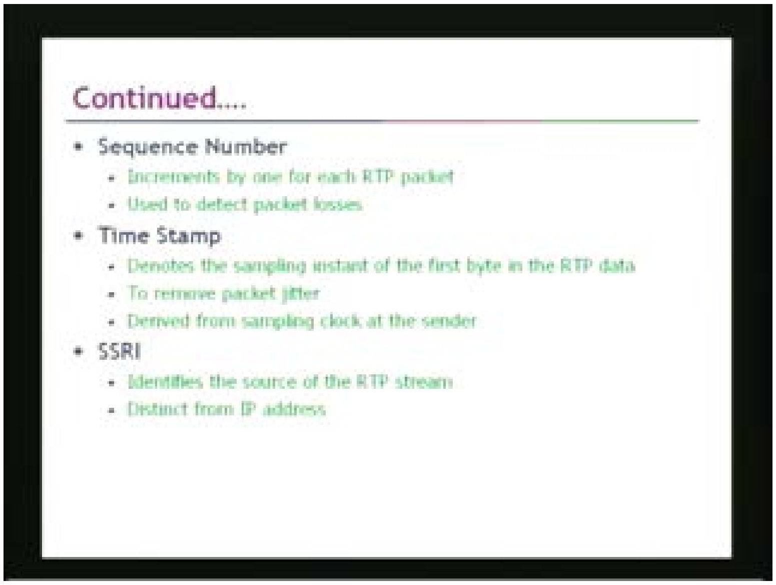
So, there are different types of encodings that can be there and the first 7 bits that will indicate what type of encoding has been used in the payload because it is important so that the receiver will use an appropriate decoder for decoding the payload type. As a matter of fact, the sender can change the encoding on the fly also because the sender changes the encoding on the fly all it needs to do is to indicate to the receiver by putting the appropriate bit which is meant for encoding technique in the header and then the receiver can then shift to an appropriate decoder and decode the payload data accordingly. So, that is the importance of this payload types.

Now, you can see the second thing is this of course I have already told mentioned that sender can change the encoding on the fly. Now, one thing that is important to be seen is to be observed is that RTP itself does not provide any mechanism to ensure the timely delivery of the packets. It does not also guarantee the delivery of packets or prevent out of delivery, out of order delivery and RTP encapsulation is only seen at the end point.

So, what does it mean to say it means to say that one of the most important things to be observed is that RTP itself does not provide any quality of service guarantees, it does not guarantee the delivery of the packets and it does not ensure that the packets will not be delivered out of order or anything like this. RTP encapsulation is seen only at the end points and it is primarily for the purposes of identifying the RTP streams to the synchronization source identifier and taking care of the delay jitter problems by appropriately time stamping the packets and also indicating to the receiver that what kind of payload the particular packet is carrying.

So, that is the important function, I mean this should not be confused that since RTP is a primary transport protocol for the voice, it can provide any quality of service guarantees or anything like that. It does not do that thing at all. So, we had seen the payload type thing. Now, let us look at the sequence number and the time stamp and the synchronization source identifier.

(Refer Slide Time: 17:18)



Continued….

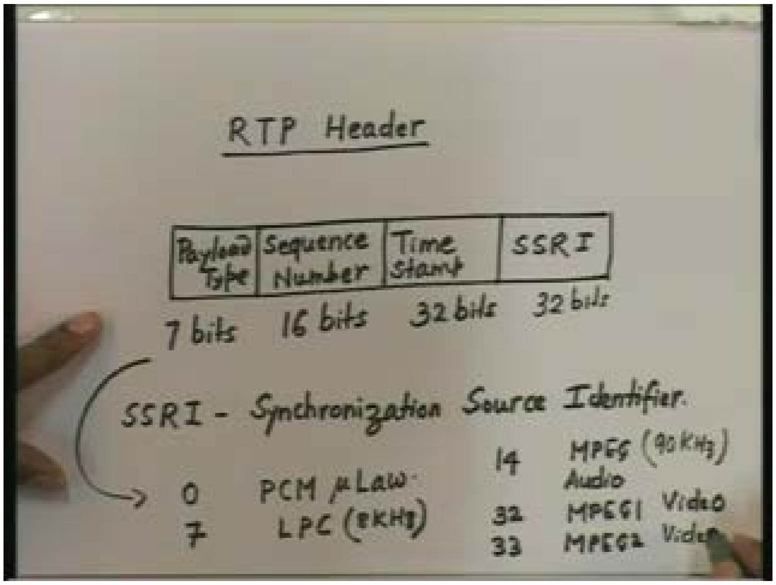
- Sequence Number
  - Increments by one for each RTP packet
  - Used to detect packet losses
- Time Stamp
  - Denotes the sampling instant of the first byte in the RTP data
  - To remove packet jitter
  - Derived from sampling clock at the sender
- SSRI
  - Identifies the source of the RTP stream
  - Distinct from IP address

So, as you can see here, the sequence number is 32 bits and the sequence number, it increments by 1 for each RTP packets. So, the sequence number field is actually 16 bits and you can choose the sequence number randomly and then you can increment by 1 for each RTP packets and what is the reason for the sequence number? It is actually used to detect the packet losses.

Time stamp as we have already discussed, it is very important to combat the delay jitter and the time stamp is 32 bit and it actually denotes the sampling instant of the first byte in the RTP data and this time stamp is actually derived from the sampling clock at the senders and as we have seen it is basically used to remove the packet jitter.

Synchronization source identifier, it is again 32 bits. So, sequence number is 16 bits, time stamp is 32 bits, synchronization source identifier is also 32 bits, it identifies the source of the RTP stream and as I already discussed, it is very distinct from an IP address. The source can have a different IP address. It is basically chosen randomly and it is associated with a particular RTP stream. A particular source may have a number of RTP streams and then each RTP streams will be associated with the synchronization source identifier.

(Refer Slide Time: 18:51)



RTP Header

Payload Type	Sequence Number	Time Stamp	SSRI
7 bits	16 bits	32 bits	32 bits

SSRI - Synchronization Source Identifier.

0	PCM μLaw.	14	MPEG (90 KHz) Audio
7	LPC (8KHz)	32	MPEG1 Video
		33	MPEG2 Video

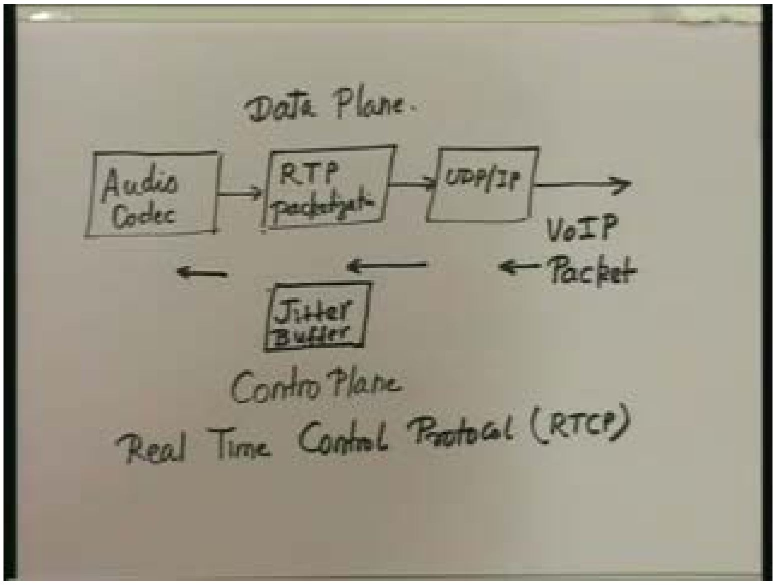
So, as we had seen here that we have 7 bits, we have 16 bits for sequence number, we have 32 bits for time stamp and we have 32 bits for synchronization source identifier. So, this constitutes the entire RTP header. So, this becomes 64 and 16 bit.

Now, as you can see some of this payload type as I was saying that some of them have been standardized. Some of the payloads, for example, if this payload type number is 0, then you can say that you are using a PCM mu law. If the payload type is 7, then you are using LPC kind of coding with 8 kilo hertz of sampling rate and if the payload type is 14, then you are using mpeg audio and which is like 90 kilo hertz and if it is payload type let us say 32, then it is basically for mpeg 1 video and if it is of 33, then this is mpeg 2 video.

So basically, some of these payload type numbers that have been standardized; of course there are a possibility since it is 7 bit, there is a possibility that you can have 128 types of payloads. Some of them of course are unused fields, one can assign as and when new encoding techniques are discovered or the sender or the receiver mutually agree to use their own proprietary types of encodings, then they can actually assign these payload type bits appropriately and use the decoder accordingly.

Now, so that is what we had seen the function of the RTP. So, you can see basically when we have in the voice over IP in the data plane, basically you will have the audio codec and audio codec will then give to sort of RTP packetization.

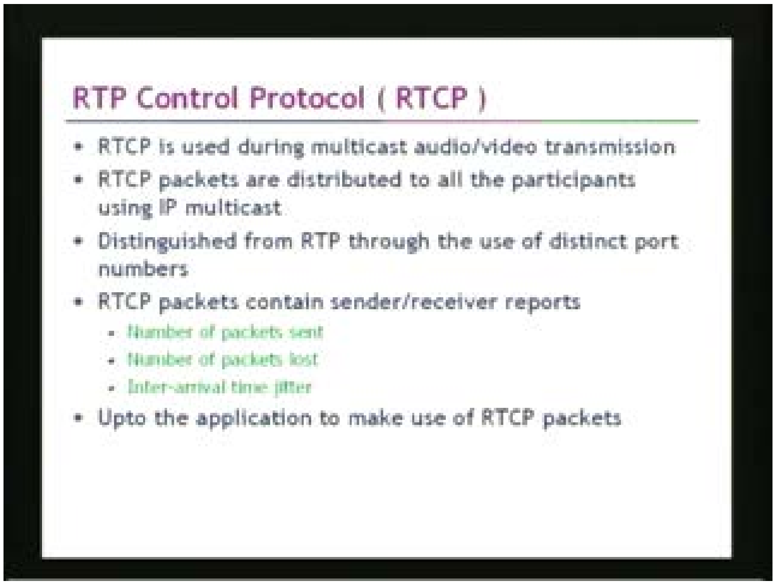
(Refer Slide Time: 20:53)



So, this is like the packet RTP packetization and then it will go to UDP IP protocol stacks and an IP voice over IP packet will come out and on the receiver, you can have here a jitter buffer which a jitter buffer which will actually remove the jitters, the delay jitters by using the information from the RTP.

So, this is typically you can say as the data plane for the voice over IP. So, there is a apart from the RTP on the control plane, there is another protocol which is to be used along with the real time protocol and that protocol is called RTCP or real time control protocol. So, let us see what are the features. So, this is so if you look at here, then on the control plane, we will have real time control protocol or RTCP. So, what are the features of the RTCP? So, we will just see in the slides that what is the function of the RTCP.

(Refer Slide Time: 22:38)



RTP Control Protocol (RTCP)

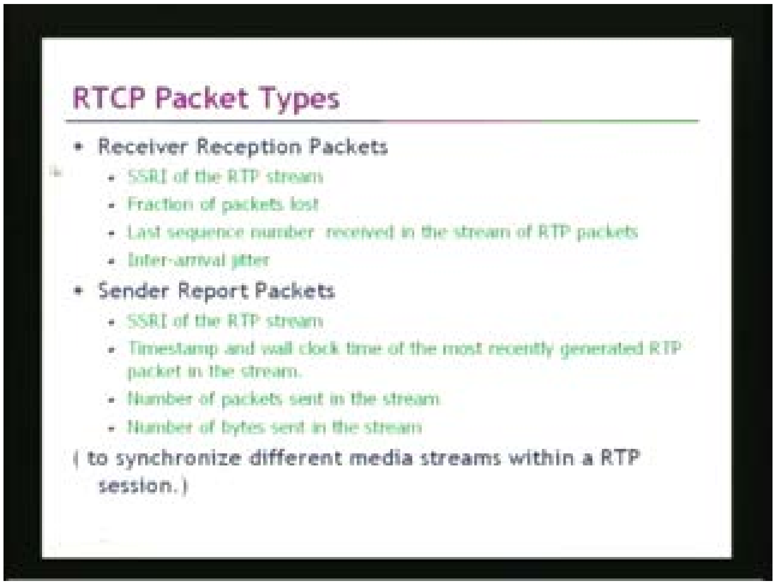
- RTCP is used during multicast audio/video transmission
- RTCP packets are distributed to all the participants using IP multicast
- Distinguished from RTP through the use of distinct port numbers
- RTCP packets contain sender/receiver reports
  - Number of packets sent
  - Number of packets lost
  - Inter-arrival time jitter
- Upto the application to make use of RTCP packets

So, RTCP is used during multicast audio or video transmissions and RTCP packets are distributed to all the participants using IP multicast and it is distinguished from the RTP through the use of distinct port numbers. The port numbers, UDP port numbers which are used for RTP and RTCP are different. Now, what is the function of the RTCP? The RTCP contains the sender and receiver reports. So, it contains the information like number of packets sent, number of packets lost, inter arrival time jitter at the receiver and so on.

So, this is like a statistics report which is sent by the RTCP from the sender and receiver and it is of course up to the application to make use of RTCP packets. RTCP does not dictate how this information like number of packets sent or number of packets lost or inter arrival time jitter excreta have to be used. Applications, different applications may come up with different algorithms and mechanisms to best make use of these informations.

So typically, what is there? In the data plane, we are having these packetised voice which is using RTP and UDP as the transport mechanism and then these voice over IP packets are being transmitted over the internet along with these in the control plane the sender and receiver they are also sending out these RTCP controlled packets and they are associated with the each RTP stream and they contain some informations like statistics number of packets sent, number of packets received, inter arrival time jitters and so on. There are different types of RTCP controlled packets. We will just have a look at them, what are those types of RTCP controlled packets.

(Refer Slide Time: 24:17)



RTCP Packet Types

- Receiver Reception Packets
  - SSRI of the RTP stream
  - Fraction of packets lost
  - Last sequence number received in the stream of RTP packets
  - Inter-arrival jitter
- Sender Report Packets
  - SSRI of the RTP stream
  - Timestamp and wall clock time of the most recently generated RTP packet in the stream.
  - Number of packets sent in the stream
  - Number of bytes sent in the stream

(to synchronize different media streams within a RTP session.)

As you can see here that the type of packets which are there are something called as receiver reception packets. So, they will contain information like SSRI - synchronization source identifier of the RTP streams, they will contain information like fraction of packets lost, they will contain information like last sequence number received in the stream of RTP packets and they will contain inter arrival jitters.

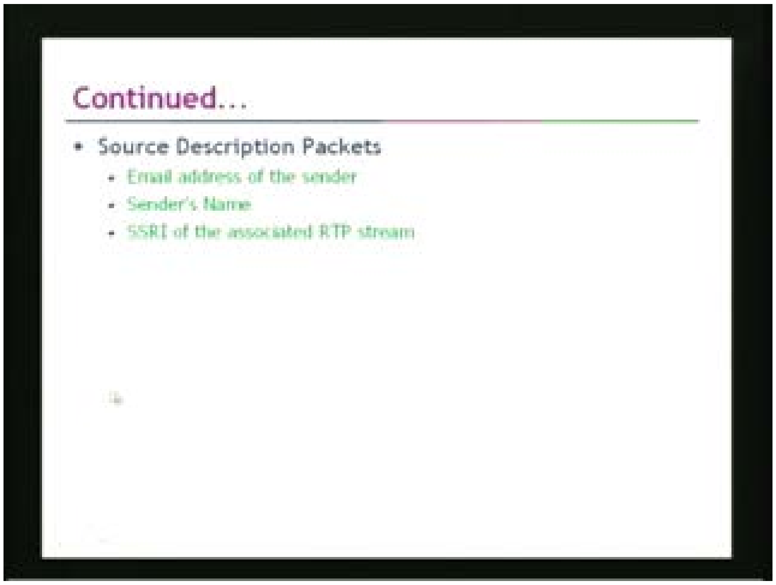
So, there are basically two types of packets. One is receiver reception packets, so they contain this information. The second packets are sender report packets, so they contain informations like synchronization source identifier of the RTP streams SSRI of the RTP stream, they also contain information like time stamp and wall clock time of the most recently generated RTP packets in the stream and they contains also informations like number of packets sent in the streams, number of bytes sent in the streams.

Now, this informations can be used to synchronize the time stamp and wall clock times and SSRI of the RTP streams also can also they can be used to synchronize different media streams within a RTP session. You can use this information to synchronize different media streams. You can also use informations like number of packets sent and number of packets received etcetera to have appropriate congestion control or rate control algorithms as well. So, that information we will see.

So, RTCP is having receiver reception reports, it is having sender reception reports and it is sending lot of informations and it is up to the application to make use of this informations. For example, if you observe through RTCP packets that at the receiver there are lot of packets getting lost, then it is obviously the internet is congested and then you can appropriately adapt the sending rate or you can appropriately change the quantization level.

So, that is one of the techniques that can be used by the application, this informations can be used by the applications in this way. Then, there are other types of packets.

(Refer Slide Time: 26:21)



Continued…

- Source Description Packets
  - Email address of the sender
  - Sender's Name
  - SSRI of the associated RTP stream

So, we have seen that there are receiver reception packets and there are sender reception packets; another type of packets that are used is what is called as source description packets. So, they will contain some information like email address of the sender, the sender’s names or synchronization source identifier of the associated RTP stream and so on. So, these are what are called as source description packets or SDPs.

So, as we have seen RTP is part of data plane of the voice over IP and RTCP is a part of the control plane primarily to assist the RTP protocol and to assist the application. So, that is it on the control plane.

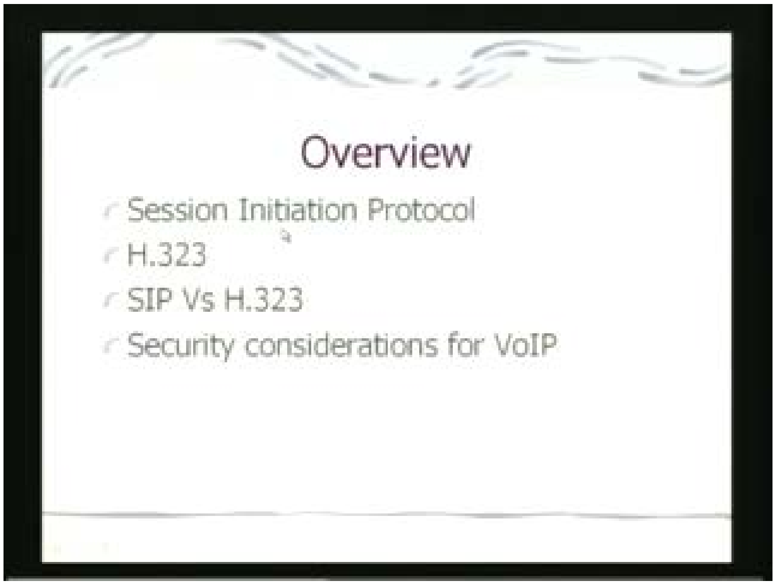
So, essentially if you review what we have learnt so far, then we will see that the analog speech is converted to digital speech and by using some compression techniques, we can reduce the data rate of the streams through an appropriate audio coding or video coding mechanisms if it is a multimedia stream and also video and then we use RTP and UDP as a transport mechanism time stamp each packet put sequence number and so on and then send this stream out over the internet which is actually the best effort internet and at the receiver when then these packets are received, then you can use jitter buffer and an appropriate play back buffer algorithms which could use an adaptive play back buffer algorithms to remove the delay jitters that maybe present in this stream of packets and then play these packets out at the receiver.

So, this way you can have internet telephone or a voice over IP telephone the data plane part of this. Now, as we know that in the typical circuit switched networks, before actually you can make a call, you have to dial the destination number and through a signaling process and once the destination number is dialed and you the destination responds back with the connect, then your call gets connected and then you can start transmitting the informations.

Now, similar things here we need to do because what we have discussed so far is more concerned with the transmission part or the data plane parts. We have to now understand how before actually we can start speaking to the other party, what kind of signaling we need to have and what kind of signaling protocols we need to establish.

So, we will now study some kind of signaling mechanisms that we need to use. So, as we had seen that we need to then understand some of the signaling protocols, so we will see some of the signaling protocols now and so let me just tell you that there are two types of signaling protocols that are currently in use.

(Refer Slide Time: 29:18)



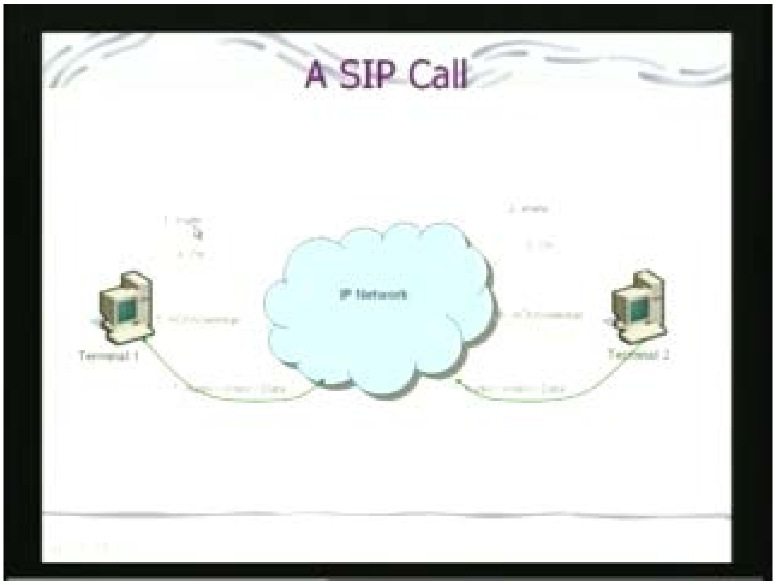
Overview

- Session Initiation Protocol
- H.323
- SIP Vs H.323
- Security considerations for VoIP

One is called session initiation protocol which is SIP. This is the most commonly used protocol. It was proposed by IETF that is the internet engineering task force and another protocol is H.323 which was actually proposed by ITU.

So, what we do will do is that we will first get an overview of the session initiation protocol, the SIP and then we will look at H. 323 protocols and then we will see SIP versus H. 323, we will see what are the pros and cons of SIP and H. 323 and then finally, we will also look at security considerations for the voice over IP. So, now let us look at session initiation protocol or the SIP.

(Refer Slide Time: 30:03)



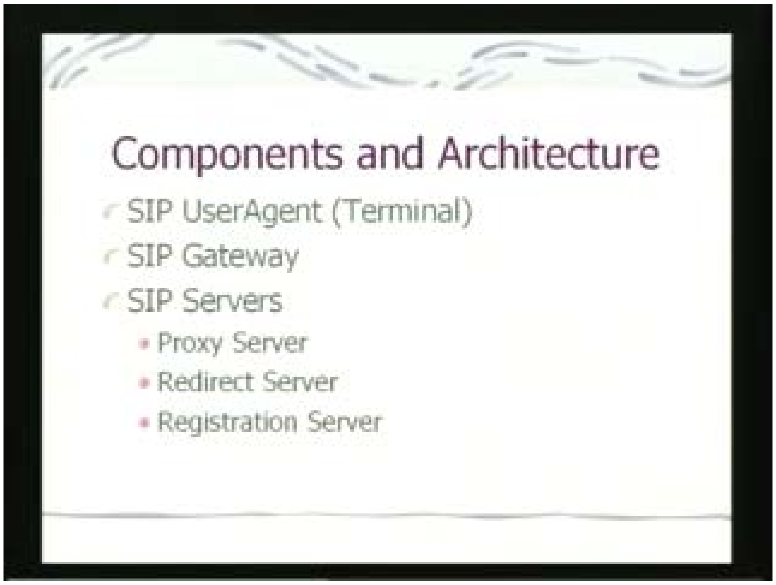
A SIP Call

So, this actually shows how do you do a signaling by using a SIP protocol. So, there are 4 kinds of messages which are there. One is called as invite, one is called invite, then there is okay, then there is acknowledge and then of course, you can transmit the audio and video. So basically, a terminal will first send an invite message to the internet which it wants to make a call.

So therefore, it will send an invite message and once it has sent an invite message, you will get an acknowledge message from the internet and once you got an acknowledge message, it will send a message and then after that you can start transmitting the audio or video. So, as you can see here that this invite message will go which was initiated from terminal 1, it will go to the IP networks and then it will go to the terminal 2. If the terminal 2 is ready to accept the call, it will send the acknowledge message which will transmitted through the IP networks back to the terminal 1 and then the terminal 1 will send an okay call. So this way, the call can be established and and then the audio or video data can be transmitted.

So, what are the various components and architectures for the SIP? There is what is called as the SIP user agent. So, a SIP user agent is actually your voice over IP terminal. Then there is a SIP user gateway and then there are various kinds of servers which are various kinds of proxy servers. So, these are the various kinds of servers which are called as Proxy servers or the Redirect servers and the Registration servers.

(Refer Slide Time: 31:30)



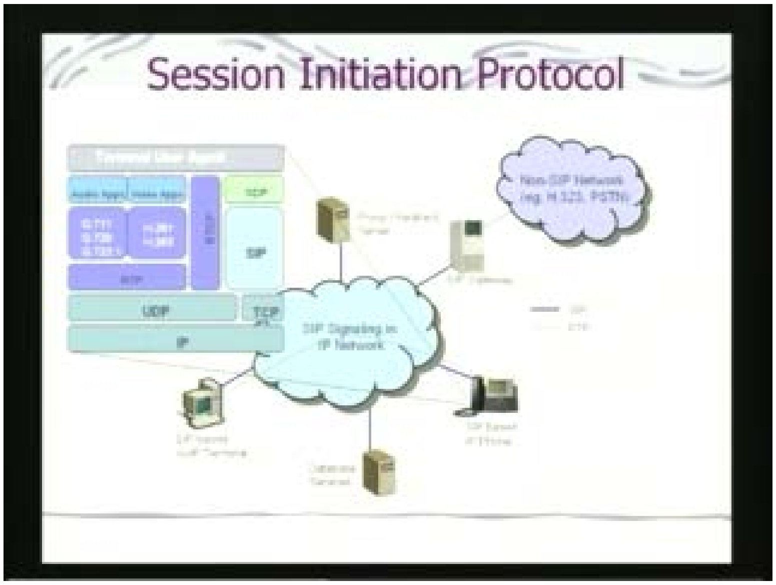
Components and Architecture

- SIP UserAgent (Terminal)
- SIP Gateway
- SIP Servers
  - Proxy Server
  - Redirect Server
  - Registration Server

So, there are 3 kinds of servers. So, you can see in this diagram; there is the SIP based voice over IP terminal which is usually actually your user agent, then there are proxy or redirect servers and of course, then there are SIP gateways. SIP gateways actually may connect a SIP based network to a non SIP based networks like H. 323 networks also or so on. That is the function of the gateway.

So, there are 3 major components as we have discussed. One is the SIP user agent which is actually your terminal, then there is a SIP gateway which actually is used to connect between a SIP network and non SIP network maybe a public switch telephone networks like a ordinary circuit switch based telephone networks or maybe another voice over IP or internet telephony based networks which is based on H. 323 and so on. So, it is a SIP gate and then there are proxy servers or registration servers or redirect servers. So, registration servers maybe used to register your call and proxy servers to redirect your SIP request.

(Refer Slide Time: 32:43)

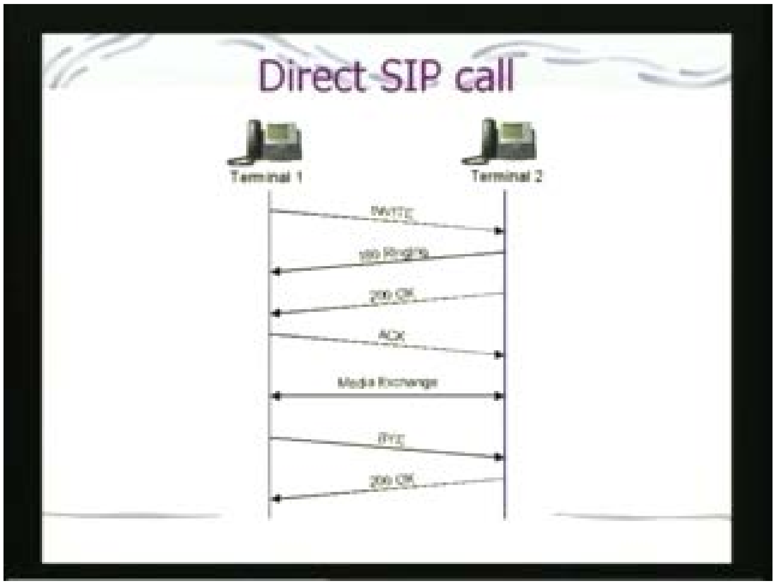


Session Initiation Protocol

So, as you can see here that this actually shows the protocol stack here. You can see here that there is a terminal user agent which is actually generating the audio and video applications and then you have as you can see in this diagram, the blue. We have this g 71 or g 729 or g 723. These are various kinds of codecs that can be used. Then you have this RTP encapsulation and finally UDP and IP and then you can have sort of a SIP based phone. So, this is the… on the other hand you can see here, SIP is separately shown as the signaling protocol and SIP is using TCP as the transport mechanism.

So, this is a typical voice over IP protocol architectures which will be sitting on the SIP phones. So, one is on the data plane which is your various kinds of audio codec applications which we saw in the slides. g 711 g 723 g 729, these are various kinds of audio coding standards which could be there and then you code the data, you encapsulate it using RTP and then send it using UDP. But on the other hand, the signaling protocol, SIP messages, SIP packets will be sent using TCP as the transport mechanism. So, they will be basically sent over TCP IP networks and your voice packets are being sent over an UDP IP network.

(Refer Slide Time: 34:06)

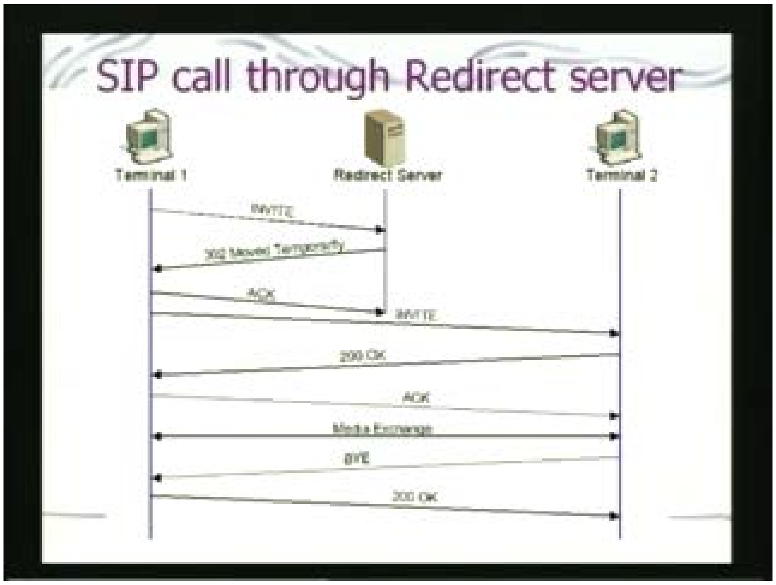


Direct SIP call

So, this is like a typical SIP call as we have seen. You send an invite, then it rings, then you send an okay message, finally you acknowledge, you have media exchange and then when you have to release the call you send a bye message which is then acknowledged by sending an okay message.

So, this is how a direct SIP call when you do not have any server. If you have a proxy server, then you send an invite message through a proxy server, the proxy server will relate to terminal 2 through another invite you get 180 ringing back then you get an okay message back. Finally, you acknowledge, then the media exchange and then the bye that is the call release sessions.

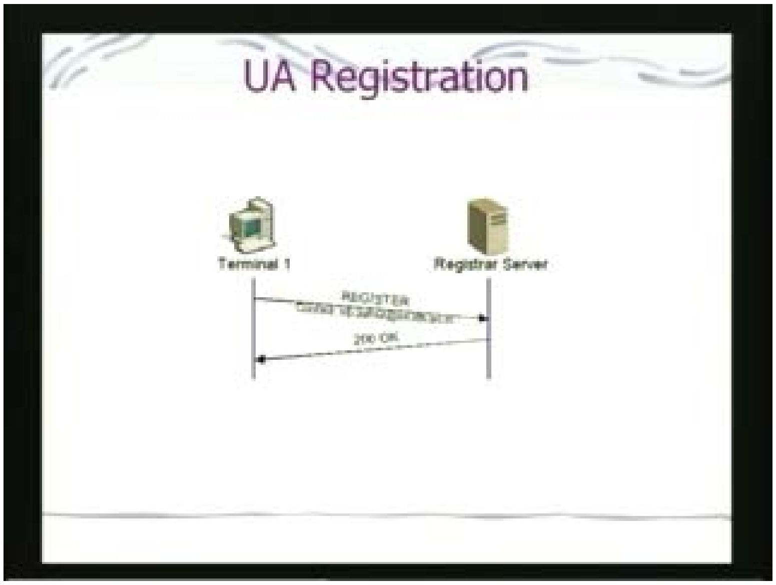
(Refer Slide Time: 34:50)



SIP call through Redirect server

This is a SIP call through a redirect server. If you discover that terminal 2 has moved temporarily, then the call needs to be redirected. So, that information also you will get it from the redirect servers. The rest of the other steps actually, they remain the same. This is of course the user agent, registration. So, terminal 1 may want to register with the registration servers and then once the registration is complete, you get an acknowledgement by sending an okay message.

(Refer slide: 35:07)



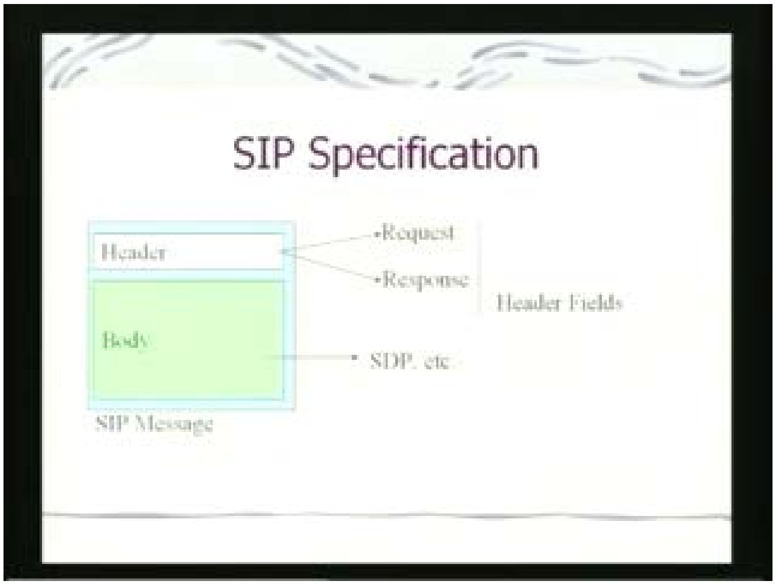
UA Registration

So, one thing however one needs to understand that whether it is a SIP signaling or whether it is a H. 323 signaling, these signaling messages are basically used to communicate or indicate to the receiver or to the destination that a particular user agent or terminal is interested in making a call and when that user agent or when that terminal accepts that invitation or accepts its willingness to talk, then it can send a acknowledge or message and then finally you can have a media exchange using RTP as the transport mechanism.

But one thing we must understand that the SIP does not guarantee the quality of the service guarantees. So, it is only merely doing the signaling messages to communicate to the destination or the reception party that a particular sender is interested in talking. It does not offer any quality of service guarantee. So, the fact that a SIP call has gone through or the call establishment process has been completed successfully does not mean that later on the quality of the speech or the quality of the call that will go through will be good. So, that is never guaranteed. So, that is one of the things in the voice over IP distinct from the traditionally public switch telephone networks.

Remember that in public switch telephone networks, when your call goes through the call establishment procedures, automatically the resources are allocated in the switches and a time slots are allocated to you and a quality of service is always guaranteed. So, resource reservation and the call establishment signaling is inter linked in the traditionally circuit switched telephone networks. However, that is not the case in the voice over IP case; the SIP is completely different, it is just used for establishing the call.

(Refer Slide Time: 37:12)



SIP Specification

Header

Body

SIP Message

•Request

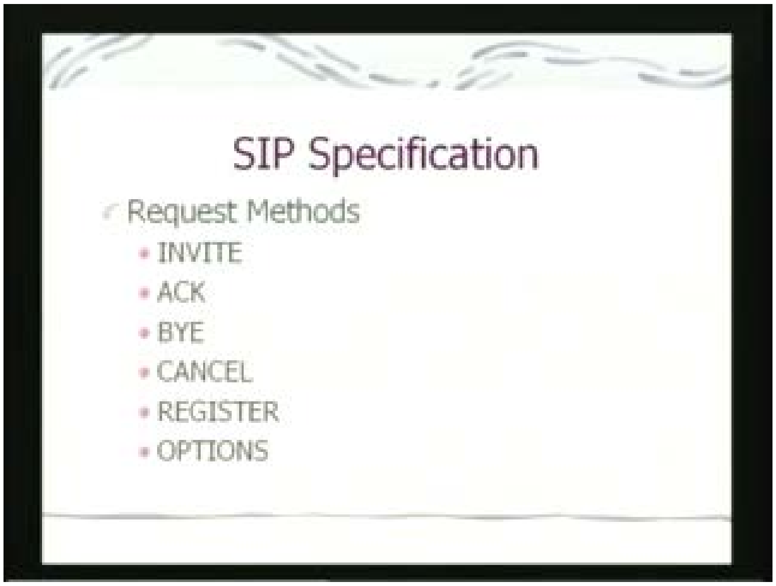
•Response

Header Fields

• SDP, etc.

So, this is the SIP message as you can see; there is a header and of course there is a body, there are two types of SIP you can have. One is either the request as there invite and all these messages or is the response which is like ack and bye and so on. Then this is the SDP. So, you can see here, the request methods. Example is invite, ack, bye, cancel, register or there are different options.

(Refer Slide Time: 37:32)

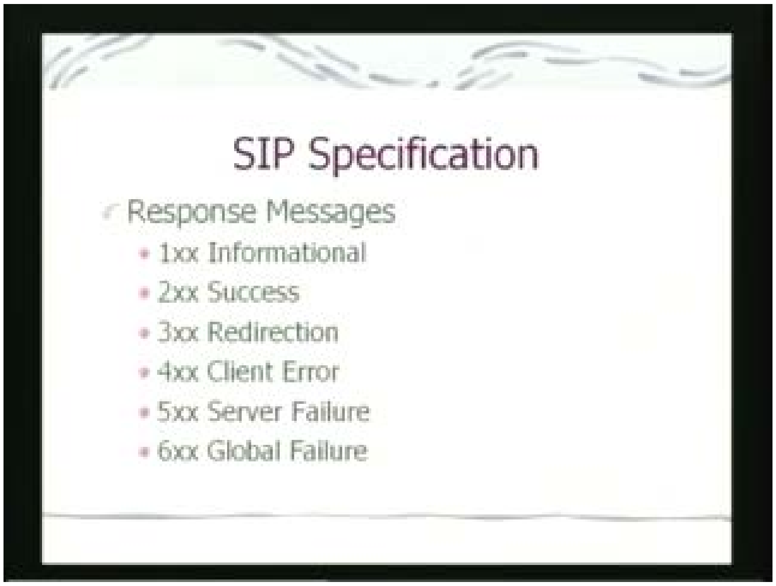


SIP Specification

- Request Methods
  - INVITE
  - ACK
  - BYE
  - CANCEL
  - REGISTER
  - OPTIONS

Response messages could be some kind of informational or success or redirect or client error or server failures and global failures and so on.

(Refer Slide Time: 37:38)

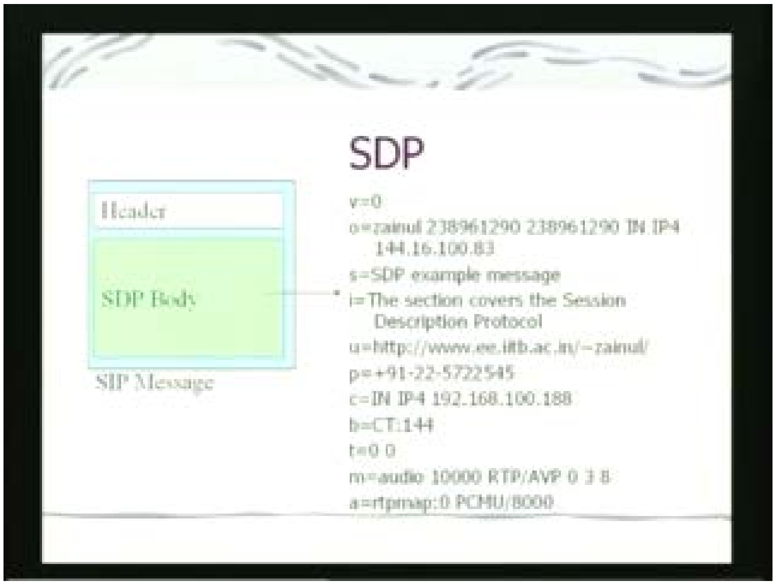


SIP Specification

- Response Messages
  - 1xx Informational
  - 2xx Success
  - 3xx Redirection
  - 4xx Client Error
  - 5xx Server Failure
  - 6xx Global Failure

So, these are listed as 1xx, 2xx, 3xx. Depending upon various kind of informational messages or success messages or redirection messages; you can have various types of informational messages with appropriate numbering, various types of success messages with appropriate numberings, various types of redirection messages with appropriate numberings and so on.

(Refer Slide Time: 38:19)



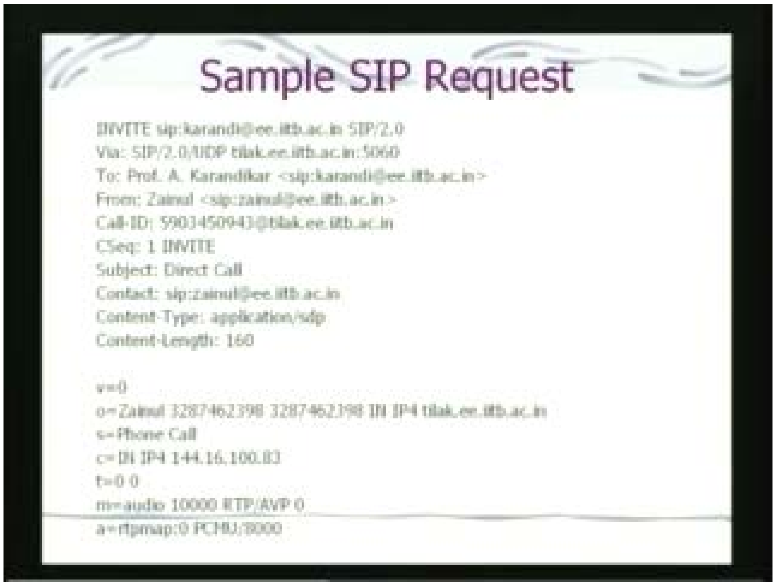
SDP

Header
SDP Body
SIP Message

```
v=0
o=zainul 238961290 238961290 IN IP4
    144.16.100.83
s=SDP example message
i=The section covers the Session
    Description Protocol
u=http://www.ee.iitb.ac.in/~zainul/
p=+91-22-5722545
c=IN IP4 192.168.100.188
b=CT:144
t=0 0
m=audio 10000 RTP/AVP 0 3 8
a=rtpmap:0 PCMU/8000
```

SIP headers are: as you say, general headers are there, request headers, the respond headers and the entity headers. So, this is the SDP body, the SIP body which will have this message for example http WWW dot ee dot iit dot ac dot in, the phone number, the IP address and what is the kind of audio, the media types and so on. What is the version and so on. These are the different types of informations that can be there in the body.

(Refer Slide Time: 38:37)



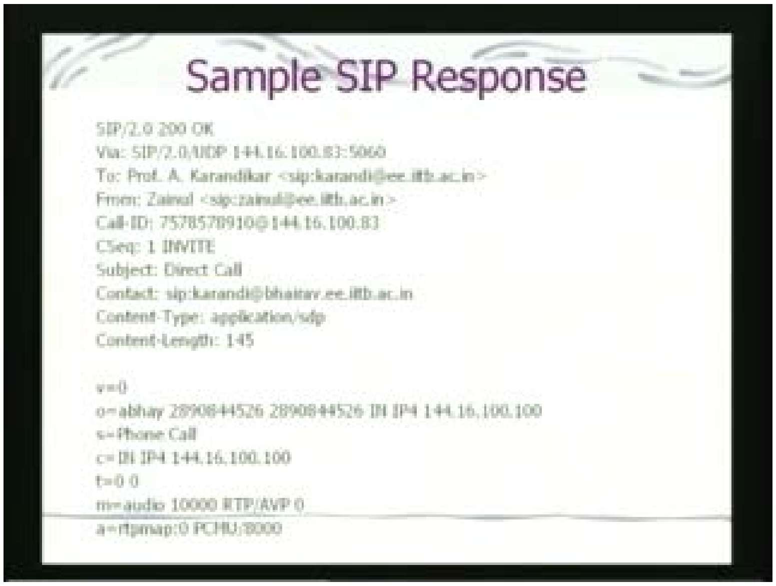
Sample SIP Request

```
INVITE sip:karandi@ee.iitb.ac.in SIP/2.0
Via: SIP/2.0/UDP tilak.ee.iitb.ac.in:5060
To: Prof. A. Karandikar <sip:karandi@ee.iitb.ac.in>
From: Zainul <sip:zainul@ee.iitb.ac.in>
Call-ID: 5903450943@tilak.ee.iitb.ac.in
CSeq: 1 INVITE
Subject: Direct Call
Contact: sip:zainul@ee.iitb.ac.in
Content-Type: application/sdp
Content-Length: 160

v=0
o=Zainul 3287462398 3287462398 IN IP4 tilak.ee.iitb.ac.in
s=Phone Call
c=IN IP4 144.16.100.83
t=0 0
m=audio 10000 RTP/AVP 0
a=rtpmap:0 PCMU/8000
```

So, this is like a sample SIP request which may be sent from let us say Karandi dot ee dot iit dot ac dot in via this from zainul. So, what is the call id, which is the type of message it say, invite and so on. So, this maybe a sample SIP request which may look like and this is like a sample SIP response which will look like.

(Refer Slide Time: 39:03)



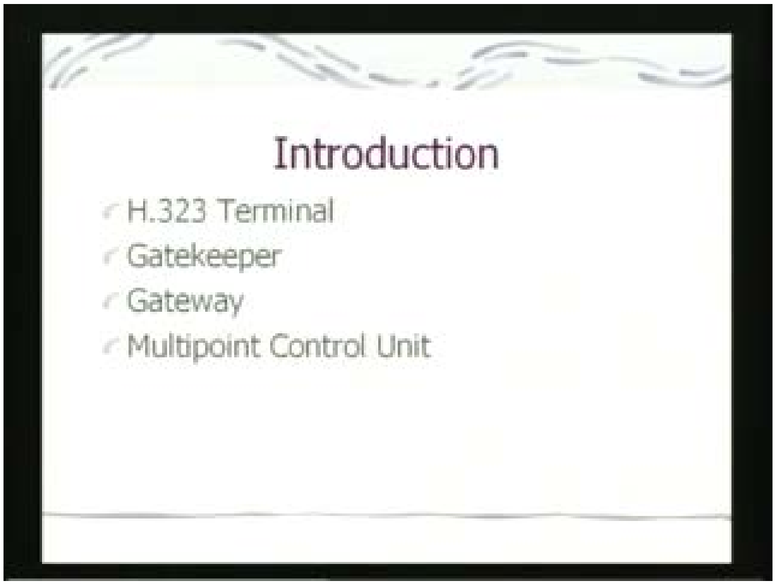
Sample SIP Response

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP 144.16.100.83:5060
To: Prof. A. Karandikar <sip:karandi@ee.iitb.ac.in>
From: Zainul <sip:zainul@ee.iitb.ac.in>
Call-ID: 7578578910@144.16.100.83
CSeq: 1 INVITE
Subject: Direct Call
Contact: sip:karandi@bhairav.ee.iitb.ac.in
Content-Type: application/sdp
Content-Length: 145

v=0
o=abhay 2890844526 2890844526 IN IP4 144.16.100.100
s=Phone Call
c=IN IP4 144.16.100.100
t=0 0
m=audio 10000 RTP/AVP 0
a=rtpmap:0 PCMU/8000
```

So, we will now review about H. 323. We have already seen about the SIP. Essentially, it is a call establishment signaling where you send messages to invite a particular party to talk and then the party responds back by sending an appropriate ack message. So, this is very simple. It is nothing much complicated in the SIP protocol. But as I already pointed out that SIP is generally not used for the resource reservations unlike traditional circuit switch telephony networks where the resource reservation and the signaling is interlinked.

(Refer Slide Time: 39:43)

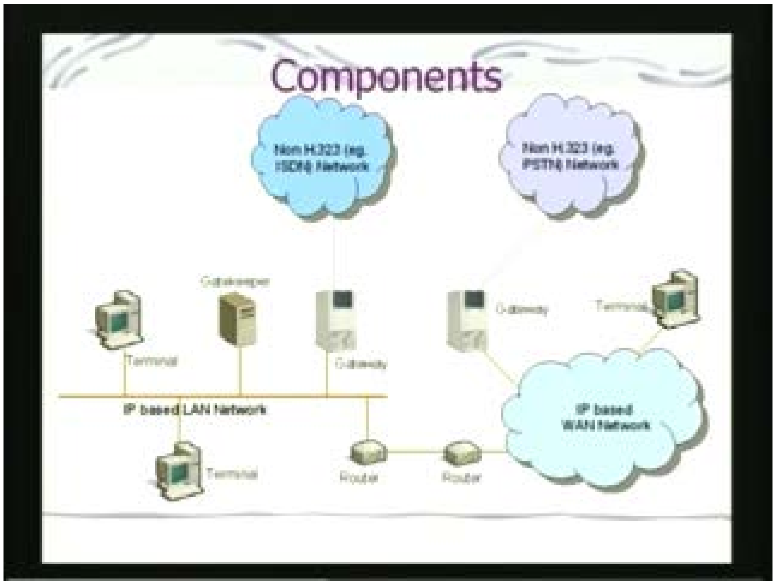


Introduction

- H.323 Terminal
- Gatekeeper
- Gateway
- Multipoint Control Unit

So, we look at H. 323. So, H. 323 is also having a similar concept like H. 323 terminal; there is a gatekeeper, there is a gateway and there is a multipoint control unit. So, these are the 4 components in an H. 323 based networks. So, you can see here, this is the gateway like the SIP gateway is used to connect between a H. 323 based network and a non H. 323 network. So, the gateway has the same functionality; whether it is in the SIP gateway or whether it is in the H. 323 gateway, so they have the same functionality.

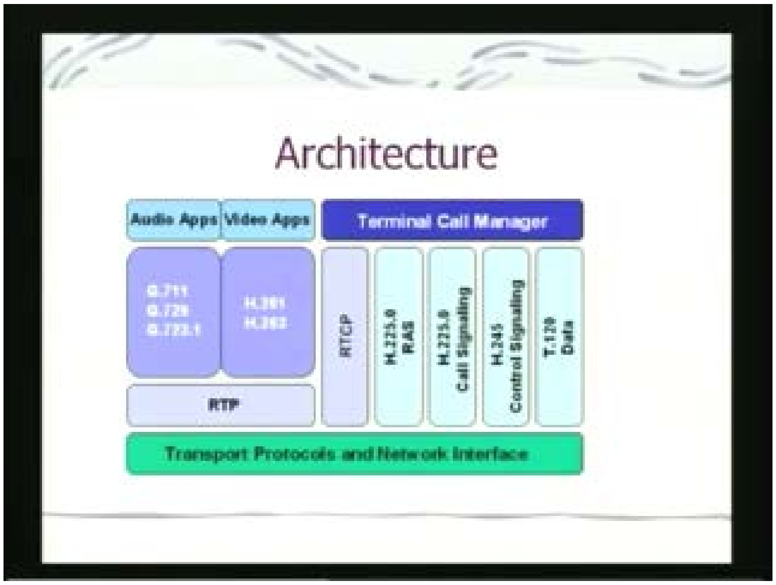
(Refer Slide Time: 40:15)



Components

Then you have the terminal which is like a H. 323 terminal and a gatekeeper which is like a registration server in a SIP based network. So, gate keeper contains all the registration or authenticated related informations.

(Refer Slide Time: 40:28)

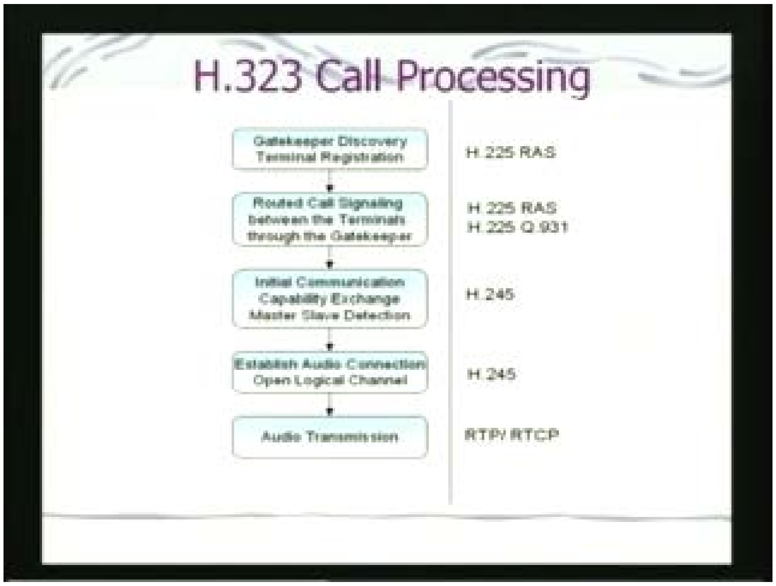


Architecture

So, this is the typical architecture. So, as you can see, as far as the data plane is concerned which is like shown here as audio applications or video applications; audio applications for example is using g 711, g 729 or g 723 types of encodings and RTP.

So, as you can see that as far as the data plane part is concerned, this is the same whether you are using a SIP based phone or a non SIP based or H. 323 based architecture. So, H. 323 is actually a combination of several standards. So, you have H. 225 which is like registration and authentication servers. Then you have H. 225 call signaling, H. 245 which is control signaling and then T. 120 which relates to data. So, there are different components of H. 323 standards: 225, 245 and so on.

(Refer Slide Time: 41:23)



H.323 Call Processing

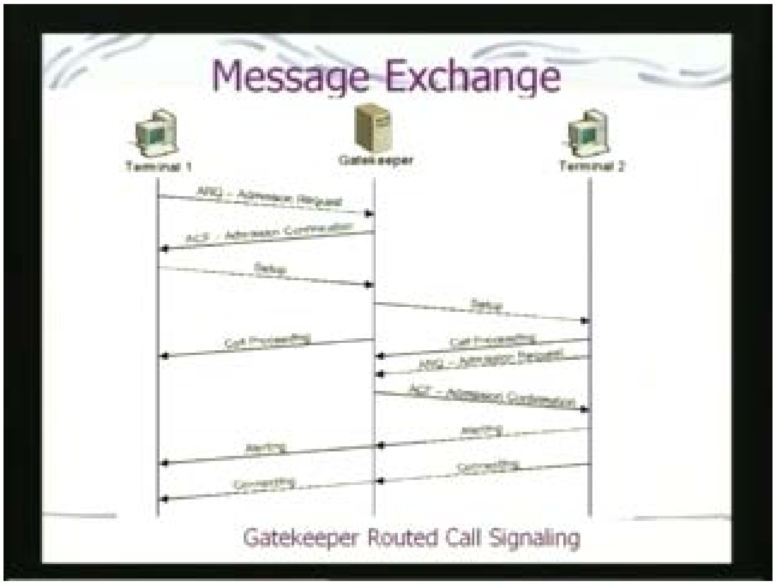
Gatekeeper Discovery Terminal Registration	H.225 RAS
Routed Call Signaling between the Terminals through the Gatekeeper	H.225 RAS H.225 Q.931
Initial Communication Capability Exchange Master Slave Detection	H.245
Establish Audio Connection Open Logical Channel	H.245
Audio Transmission	RTP/ RTCP

So, this is actually the call processing. So, the gatekeeper, you discover a gatekeeper and the terminal registers with the gatekeepers, then the call signaling is routed between the terminals through the gatekeepers, then initial communications capability exchange can takes place and then finally you establish the audio connections over open the logical channel and then start transmitting the audio using RTP.

There may be that is there in the SIP also, in H. 323 also that capability media capability exchanges also may takes place between the sender and the receiver as a part of this signaling. Apart from the fact that the sender communicates to the receiver to initiate the conversations, these media capability exchanges are also important because the sender might encode the data using some kind of encoding technique and the decoder may not have the capability to decode that data.

So, it is very important that before you start transmitting this multi media streams over the internet, both sender and receiver come to a conclusion about the capabilities of their encoding and decoding techniques which exists at 2 end points. So, that the capability exchange messages are also done as a part of these H. 323 or SIP messages.

(Refer Slide Time: 42:31)



Message Exchange

Gatekeeper Routed Call Signaling

So, this is a typical gatekeeper routed call signaling. This is very similar to the SIP based thing. So, SIP based SIP versus H. 323, differences come in terms of mostly in terms of architecture. SIP architecture is sort of similar to a http kind of a client server architectures. H.323 architecture is more akin to a telecom kind of architectures.

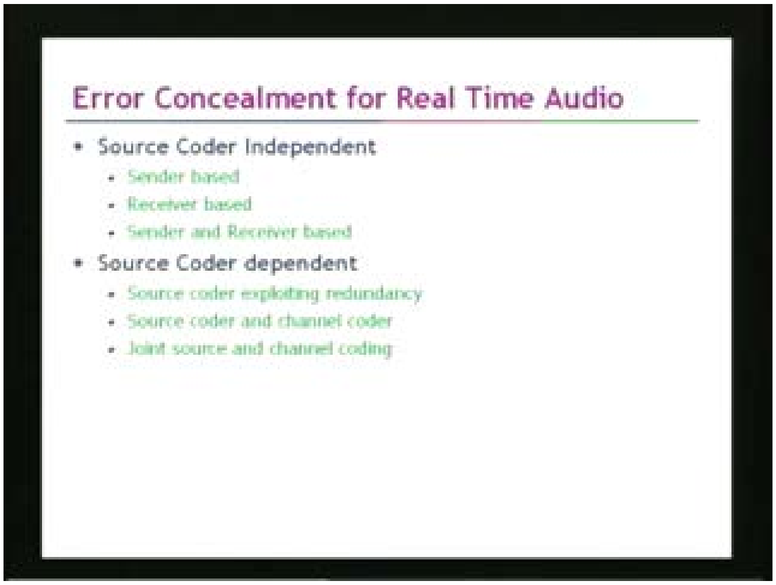
Then, the other differences that actually come in SIP is in some sense it is simpler and less complex. And therefore, it has gained much more popularity. It is an open standard by IETF and due to some of the complexity which are associated with ITU standards like H. 323; H 323 has not gained that much support among the voice over IP or internet telephony community. These are the major differences. Otherwise, the 2 have the same notions of establishing the call signaling procedures for the voice over IP.

But before we proceed further, I again want to point out that if you want to reserve the resources in the internet, then you have to undergo a separate resource reservation protocol which maybe a RSVP and so on and the SIP usually will not be a part of that. So, SIP if you have to undergo SIP along with the RSVP, then recently it has been proposed that you can use a QoS version of a SIP which is called as Q SIP. So, that versions can also be used to sort of support to do the call establishment as well as have the quality of service.

So, now what we had seen so far in our studies of voice over IP, let us recapitulate and then we will see the other aspects of the voice over IP. So, we have seen that the RTP packetization is primarily done to remove the effects of the delay jitter and by using a jitter buffer and an adaptive playback buffer algorithms and SIP or H. 323 primarily used as call establishment signaling procedures.

Now, as we have already pointed out that there could be problems for real time services due to both packet losses as well as the end to end delay. So, we will now see how this problem of the packet losses can be combated. So, we have to basically see how we can have packet loss concealment in voice over IP for the real time audio. So, this is how we will see. So, let us look at some of the techniques which are there.

(Refer Slide Time: 45:40)



Error Concealment for Real Time Audio

- Source Coder Independent
  - Sender based
  - Receiver based
  - Sender and Receiver based
- Source Coder dependent
  - Source coder exploiting redundancy
  - Source coder and channel coder
  - Joint source and channel coding

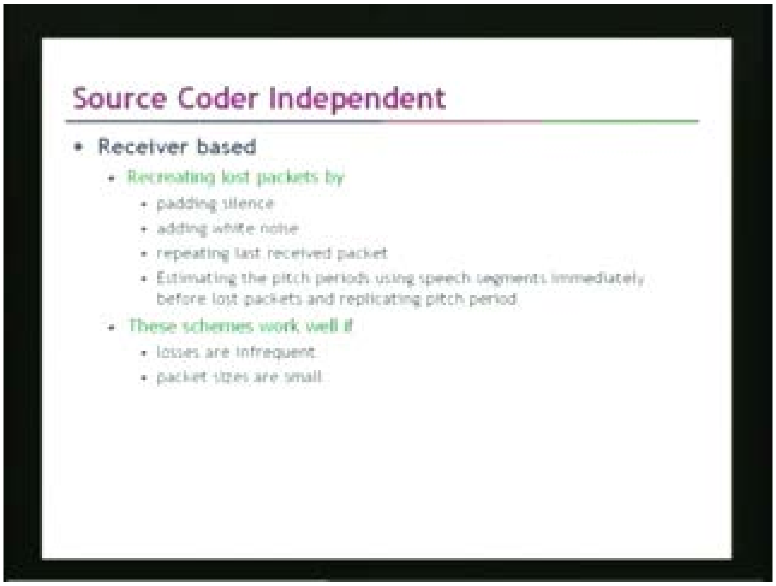
One is there are so, all the error concealment techniques which are there for the real time audio, they can be divided into 2 types. One is the source coder independent, so it is independent of whatever is the encoding technique which has been used at the source. Another one is like source coder dependent, so it is actually dependant upon what is the kind of encoding techniques which has been used at the receiver.

Now, if the source code independent techniques, they can be sender based or receiver based or they can be both - joint sender and the receiver based. Now, source coder dependent techniques which actually exploits the redundancy which maybe there in the source coders; you can have a source coder and a channel coders and you can have a joint source and channel coding, so we will actually see all these things.

Now, first let us look at the source coder independent technique and on which the receiver based techniques. Now, basically the idea is something like this that if the packets are lost at the receiver and if your technique is purely receiver based, it is not sender based at all; then what the receiver can do to combat these packet losses, small packet losses that may occur?

So, one of the popular techniques which is typically implemented as a part of the voice over IP application and which is like receiver based is that repeat recreating the lost packets by either padding the silence or by adding some white noise or even repeating the last received packets and then or estimating the pitch periods using the speech segments immediately before the last packets and then replicating the pitch periods.

(Refer Slide Time: 47:07)

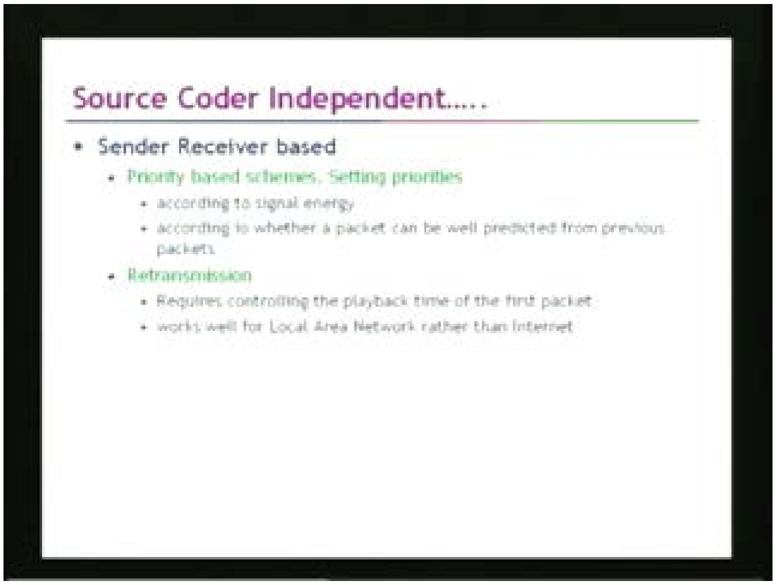


Source Coder Independent

- Receiver based
  - Recreating lost packets by
    - padding silence
    - adding white noise
    - repeating last received packet
    - Estimating the pitch periods using speech segments immediately before lost packets and replicating pitch period
  - These schemes work well if
    - losses are infrequent
    - packet sizes are small

As a matter of fact, one of the popular techniques is to replay the previous played packets. That is one of the most commonly or popularly used techniques which can be used based on the receiver. However, these schemes, the receiver based schemes really as you can observe will work well only the losses are very infrequent and the packet sizes are small; only packet sizes are small and losses are infrequent, then only these schemes will work well, the receiver based and source coder independent.

(Refer Slide Time: 47:47)



Source Coder Independent…..

- Sender Receiver based
  - Priority based schemes, Setting priorities
    - according to signal energy
    - according to whether a packet can be well predicted from previous packets
  - Retransmission
    - Requires controlling the playback time of the first packet
    - works well for Local Area Network rather than Internet

There could be a if you consider source coder independent, then there could be a combination of sender and receiver based. So, what can be done is that you can have a priority based schemes, so you can set priorities and these priorities can be set according to signal energy or according to whether a packet can be well predicted from the previous packets.

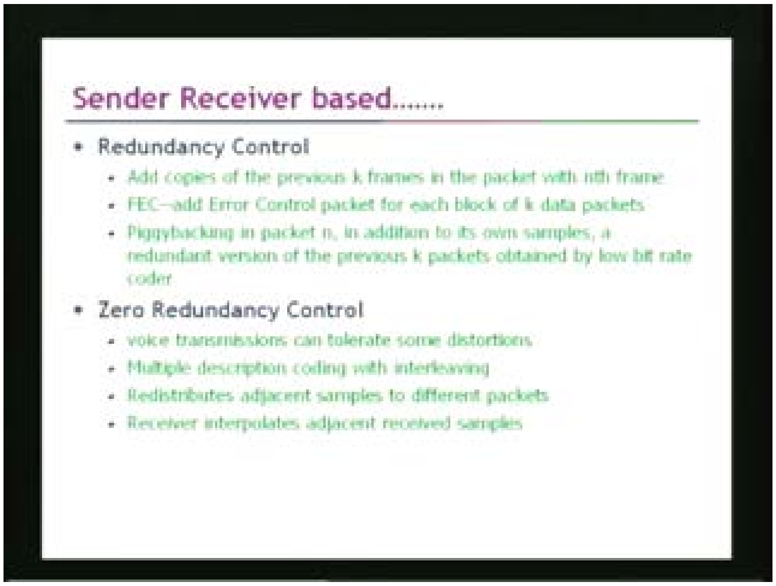
So, if the packet is well predicted from the previous packets and obviously it has low priority and therefore the attempt could be really to process the higher priority packets and this is indicated to the receiver that these are higher priority packets or low priority packets.

As a matter of fact, if the intermediate routers are able to give priorities to these packets depending upon their priority levels; then in the event of congestions of the packet losses are occurring, then higher priority packets may get through and these higher priority packets this priority can be determined whether the packet can be well predicted from the previous packet or not.

As I have already mentioned that typically when the packet losses occurs, retransmission mechanism is not really suitable for voice over IP applications. But sometimes retransmissions can really work well if the round trip time from the sender to the receiver, the round trip time is very small. If the round trip time is very small, then even the retransmission can really work because what may happen is that packets may still be in the play out buffers and you can retransmit the packets within that time itself.

So, typically therefore retransmissions will work if you are having voice over IP conversations over the enterprise networks. Typically, over local area networks where the round trip delays are not that high. So, retransmission really as you say requires controlling the playback time of the first packet and it really works well for local area networks rather than internet.

(Refer Slide Time: 49:48)



Sender Receiver based.......

- Redundancy Control
  - Add copies of the previous k frames in the packet with nth frame
  - FEC—add Error Control packet for each block of k data packets
  - Piggybacking in packet n, in addition to its own samples, a redundant version of the previous k packets obtained by low bit rate coder
- Zero Redundancy Control
  - voice transmissions can tolerate some distortions
  - Multiple description coding with interleaving
  - Redistributes adjacent samples to different packets
  - Receiver interpolates adjacent received samples

Then there are sender receiver based techniques again which could be based on redundancy control. So, redundancy control is basically using some kind of forward error correcting coding techniques. So, what you do is that in one of the techniques for example, what you can do is that add copies of the previous k frames in the packets with the n'th frames or you can use an error control packets for each block of k data packets. For every k packets, you can use an error control packets.

Actually, these error control packets can be generated by using xoring the contents of all the previous k packets. So, if you do for example, if you use a simple technique like this that if you xor the contents of the packets of the previous k packets and then make this k plus one'th packet; if you do bit by bit xoring of the various contents and then make this k packet, then if one of the packets is lost, then you can recover that lost packet. However, if more than one packet is lost, then obviously you cannot recover.

So, in order that packet losses can be recovered you can keep the value of k smaller. However, if you keep the value of k smaller, this redundant packet will increase and therefore the overheads will increase and as a result the bandwidth redundancy will further go down. So, there is a trade off basically.

Another technique is that piggy backing in packet and in addition to its own sample, a redundant version of the previous k packets which can be obtained by some kind of low bit rate coders; so, that technique can also be used. The other one is of course the zero redundancy control. Now, zero redundancy control uses some kind of inter leaving.

(Refer Slide Time: 51:36)

So, what you can do is that suppose your having some speech, consecutive speech segments. So, instead of putting this consecutive speech segments in the same packet, what you can do is that you can distribute these segments into different packets. So, if you distribute these speech segments into different packets, then even if some packets are lost, you may be able to recover from that loss without having any considerable degradation in the quality of the speech. So, this is what is called as zero redundancy control using some form of interleaving.

So, there are various techniques for recovering from these losses and all of them can work well if the packet losses are not very high and are within certain limits. So, then also one can recover and have a very good voice over IP quality.