Broadband Networks

Prof. Dr. Abhay Karandikar

Electrical Engineering Department

Indian Institute of Technology, Bombay

Lecture - 29

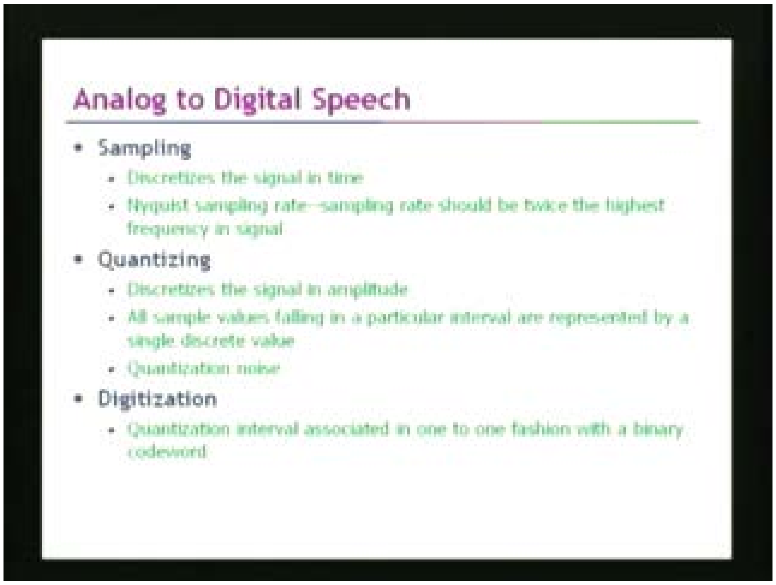
Voice over IP

So, today we will discuss about voice over IP and internet telephoning and the basic issues which are associated with voice over IP protocols. So, now as you know that the communication over packet switch networks or over internet is in the form of packet. So, the first step that needs to be done for carrying voice over internet is to first digitize the voice into the digital signals and then packetize these bits into the packets and then use some transport mechanism to transport these packets over the internet. Apart from transport mechanism, we would also need signaling mechanism to establish the sessions between the sender and the receivers.

So, overall component in terms of the voice over IP are the transport and the transmission mechanisms in the data plain and controlled and signaling mechanisms in the control plains. So, let me just briefly review the analog to digital conversions. You might have already studied analog to digital conversions in some other context or in some other course. But in this lecture, let me just briefly review some of the issues which are there in carrying the voice over IP in terms of analog to digital conversions.

So, let us just briefly review the A to D conversion and also the compression that is needed for carrying the voice over the internet protocols. Remember one thing that if you do not have a high bandwidth internet connections, then we may be suffering from scarcity of bandwidth and therefore a compression may be needed in the voice signals for achieving a high efficiency. So, we will also look into the fact that what are the various compression mechanisms that are used for the transport of voice packets over the IP.

(Refer Slide Time: 4:08)

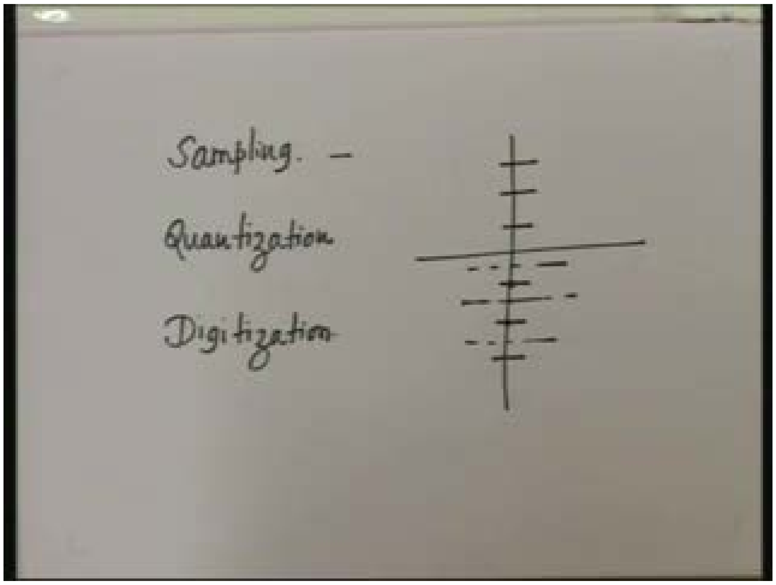


Analog to Digital Speech

- Sampling
  - Discretizes the signal in time
  - Nyquist sampling rate--sampling rate should be twice the highest frequency in signal
- Quantizing
  - Discretizes the signal in amplitude
  - All sample values falling in a particular interval are represented by a single discrete value
  - Quantization noise
- Digitization
  - Quantization interval associated in one to one fashion with a binary codeword

So, now let us start with the analog to digital conversion. Now, as you know that in any analog to digital speech conversions there are three steps that are required and these three steps are: one is sampling and another one is quantization and the third one you can say is the actual digitization or discretization.

(Refer Slide Time: 4:09)



Now, in sampling as you can see, the sampling really discretizes the signal in time domain. So, as you can see here that sampling actually discretizes the signal in time domain and as from the Nyquist sampling rate that analog signals should be sampled at a rate which is twice the highest frequency in the signal. So, an analog speech signal which is a low path signal typically lying between say 30 hertz to 3 kilohertz or 4 kilohertz would be sampled at a rate which is twice the highest frequency in signal that is the 4 kilohertz. So, this is the Nyquist sampling rate and analog signal is sampled at Nyquist sampling rate and then you get various samples.

Now, once you have got the samples, then you undergo a process of quantization. So, while sampling can be viewed as discretization of the signal in the time domain, the quantization can be viewed as discretization on the signal in the amplitude domains. So, quantization is actually discretizing the signals in amplitude and all sample values which falls in a particular interval, they are represented by a single discrete value.

So, you have various quantization levels and the signal sample values for example, you can have quantization levels like this that so that means if a sample value falls above this and but below this, then it may be quantized to this level. Similarly, if it falls below this but sample value above this, then it may be quantized this level and so on.

So, as a result quantization is actually a discretization of the signal in the amplitude domains. Now, you can see that since this is an approximation of the signal amplitude to the nearest level, it introduces some kind of an error and that error we call it to be quantization errors.

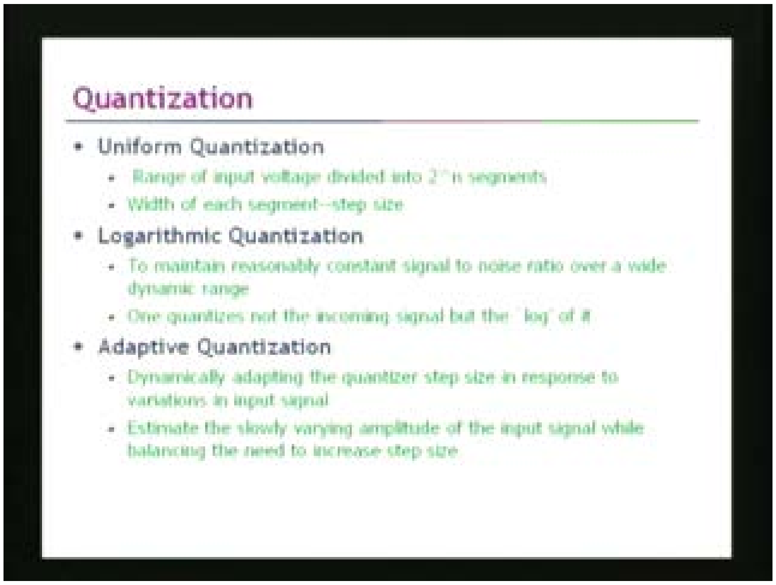
So, first step in analog to digital speech conversion is the sampling. In sampling, we sample the signal which is like discretization of the signal in the time domain and this sampling is done at Nyquist sampling rate and after the samples are obtained, we round them to the nearest level of quantization. So, this is like discretization this is discretizing the signal amplitude to fixed levels. So, this is the quantization.

Now, once the quantization is done, then you have the digitization in where each quantization interval is associated in one to one fashion with binary code words. So, it may be just be possible that there are maybe 256 quantization levels let us say, between plus 5 and minus 5 voltages and if there are 256 quantization levels, then each quantization level maybe associated with 8 bits.

So, let us say that we have an analog signal which is band limited to 4 kilohertz from 0 to 4 kilohertz which is a low pass signal. Then as per the Nyquist sampling rate, it could be sampled at a rate of 8 kilohertz or 8000 samples per second and once you get this 8000 samples per second, each sample is then quantized to one of these 256 levels and after that each quantization level is represented by 8 bit. So, as a result you get a bit rate of 8000 into 8 bits per second which gives you 64 kilo bits per second. So, that is how you get analog to digital speech conversions.

Now, as you know that in quantization, what we have done in the quantization that the amplitude levels where the signal is likely to occur that is let us say between minus V volts to some plus V volts where these two V volts is the range between which the signal amplitudes can vary; then this quantizations can be either uniform quantizations that means these two signal amplitudes have been divided into uniform levels. So, we call it to be a uniform quantizations or it could be a non uniform quantizations where the width of each segment is not uniform. So now, let us look at the uniform quantization.

(Refer Slide Time: 8:54)



Quantization

- Uniform Quantization
  - Range of input voltage divided into 2^n segments
  - Width of each segment--step size
- Logarithmic Quantization
  - To maintain reasonably constant signal to noise ratio over a wide dynamic range
  - One quantizes not the incoming signal but the 'log' of it
- Adaptive Quantization
  - Dynamically adapting the quantizer step size in response to variations in input signal
  - Estimate the slowly varying amplitude of the input signal while balancing the need to increase step size

So, in uniform quantization as you know that the range of input voltage, the range of input voltage which may lie between let us say between minus V volts to some plus V volts, it is divided into 2 raised to power n segments so that each quantization actually is represented by n bits and then width of each segment is called as a step size. So, this is how in this case is the width of each segment which we are calling it to be a step size.

Now, uniform quantization is actually the normal thing to do. But if it so happens that the probability that the signal amplitudes hovers between the low amplitude values if that is higher than the probability that it lies it in the high amplitude zones; then what maybe done is that we may resort to non uniform quantizations.

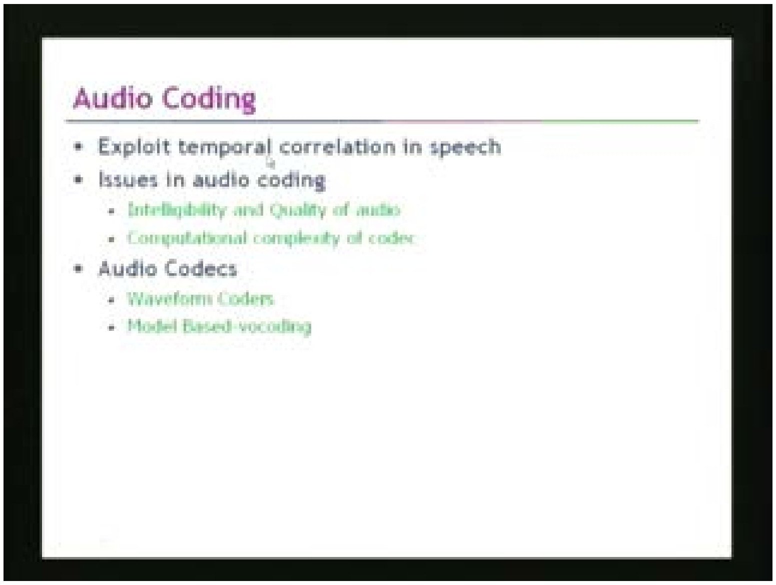
Now, what is non uniform quantization? In non uniform quantization; the segment size, the step size are not fixed but the low amplitude levels, there you may quantize into more number of levels and the high amplitude levels you may quantize in less number of levels. So, this results in into non uniform quantizations. Logarithmic quantization is kind of non uniform quantizations where we can maintain reasonably constant signal to wide noise ratio over a wide dynamic range.

So, what usually you do is that you can have a uniform quantizer. But you quantize not the incoming signal but the log of the signal. So you can actually quantize not the incoming signals but the log of it and as a result you can get logarithmic quantizations. Then we have adaptive quantizations where the step size is really not fixed. But you dynamically adopt the quantizer step size in response to variations in the input signals.

So, what you can do is that you can estimate the slowly varying amplitude of the input signals and then if the input signal is lying between the low amplitude levels, then you can have the more number of steps between the low amplitude levels. However, amplitude the signal amplitude is varying between let us say plus V to minus V volts, then you can have more number of levels between plus V to minus V.

So, really speaking, the step size you can adopt depending upon what are the voltage ranges between which the signal amplitude is likely to vary. Now, once the quantization and the digitization process is over that is you have done analog to digital conversions, as I have already pointed out that there maybe a need to do a compression in the speech, in the audio coding as a part of the audio coding and for doing this compression, we might want to exploit the temporal correlations that may be there in the speech segments. So, by exploiting the temporal correlations that may present in the speech segments, we would like compress the speech segments.

(Refer Slide Time: 12:00)



Audio Coding

- Exploit temporal correlation in speech
- Issues in audio coding
  - Intelligibility and Quality of audio
  - Computational complexity of codec
- Audio Codecs
  - Waveform Coders
  - Model Based-vocoding

So, what we will do is that as a part of the audio coding is we will exploit the temporal correlations in the speech. So, when we do this, when we achieve the when we try to achieve the compressions; there are two issues which come up in the context of audio coding. One thing is that we have to worry about the intelligibility and quality of audio and secondly we have to worry about the computational complexity of the codec which is one.

So, while we can exploit the temporal correlations in the speech, we should also take care of the fact that the speech quality is not degraded, the speech quality remains intelligible and at the same time the algorithms that we use for exploiting the temporal correlations and thereby achieving the compressions, they should not be computationally very high and computationally complex. They should be actually computationally simple because these compressions we would be doing actually while online transmissions.

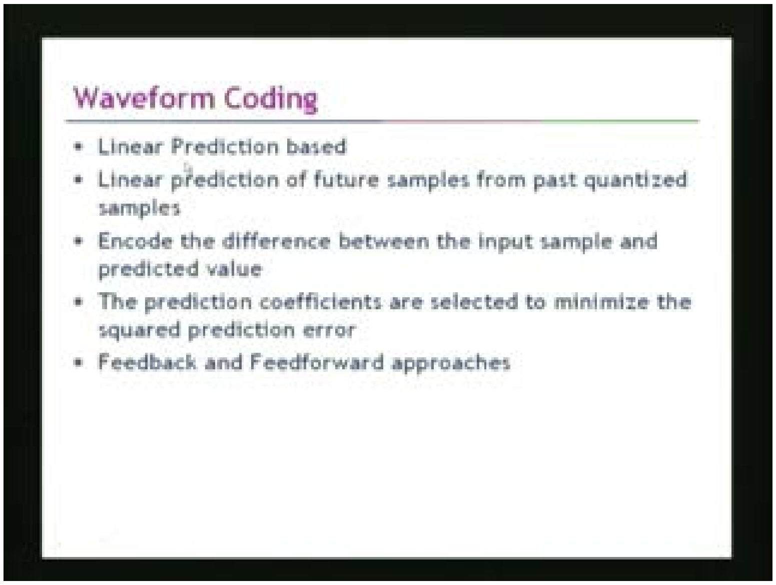
So, what are the forms of the audio codecs? Let us look at what are the forms of the audio codecs. There are there could be 2 forms of audio codecs: one is what we call as the waveform coders and second one is the model based coders or what is called as the vocoders. Now, waveform based coders typically work on the speech segments themselves. So, the speech signal is sampled is quantized and digitized and then compressed. So, those are called waveform coders.

However, the model based coders or the vocoders; they actually are based on the model of the vocal tract and they are also based on the principle of speech synthesis through a model of the vocal tract. So, that is the fundamental difference between the waveform coders and the model based vocoders.

Obviously, as we will see that since the vocoders is the principle of vocoders is based on speech synthesis through a model of the vocal tract; we will see that that will achieve a greater degree of speech compressions than the waveform coders. But then, at the same time the disadvantage of the vocoder would be that the quality of the speech produced would not be a tone quality voice it would be more like a synthetic voice.

So now, let us look at first what are the typical waveform coders that are used in practice and then we will look at the vocoders.

(Refer Slide Time: 14:31)



Waveform Coding

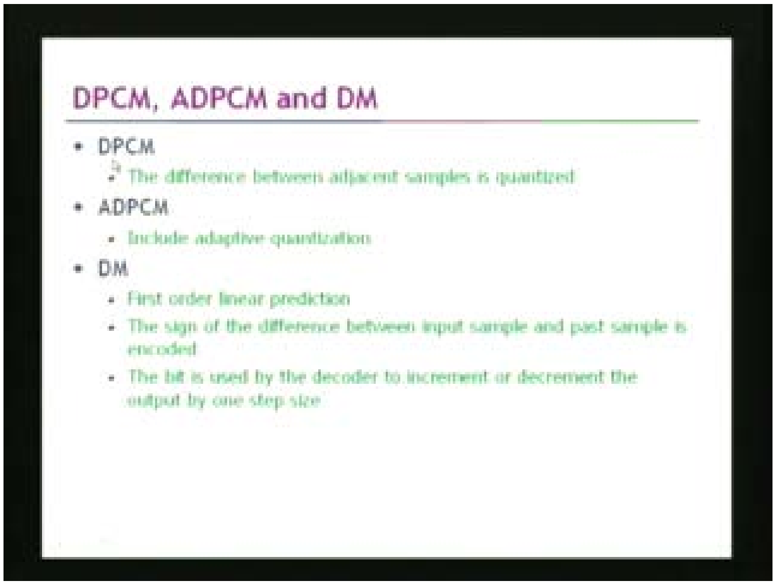
- Linear Prediction based
- Linear prediction of future samples from past quantized samples
- Encode the difference between the input sample and predicted value
- The prediction coefficients are selected to minimize the squared prediction error
- Feedback and Feedforward approaches

Now, waveform coding as you can see here that the most commonly used waveform coding could be which can achieve a greater degree of compression is like linear prediction based where you do the prediction of the future samples from the past quantized samples. So basically, this exploits the fact that there is a correlation between the samples and therefore you can predict the future samples from the past quantized samples.

Now, instead of then quantizing the samples themselves, what you do is that you encode the difference between the input samples and the predicted value and then the prediction coefficients are selected to minimize the prediction errors. So, what really you do is that you quantize the difference between the input samples and the predicted value and not really the samples itself.

So, by doing this obviously, you can achieve a greater degrees of compression, you will require lower bit rate to transmit the information because what really are now you are transmitting the information is the difference between the predicted value and the original value and note that if you transmit this error; then since the receiver can predict the samples and by using this error informations, it should be able to more accurately represent the original sample.

(Refer Slide Time: 15:52)



DPCM, ADPCM and DM

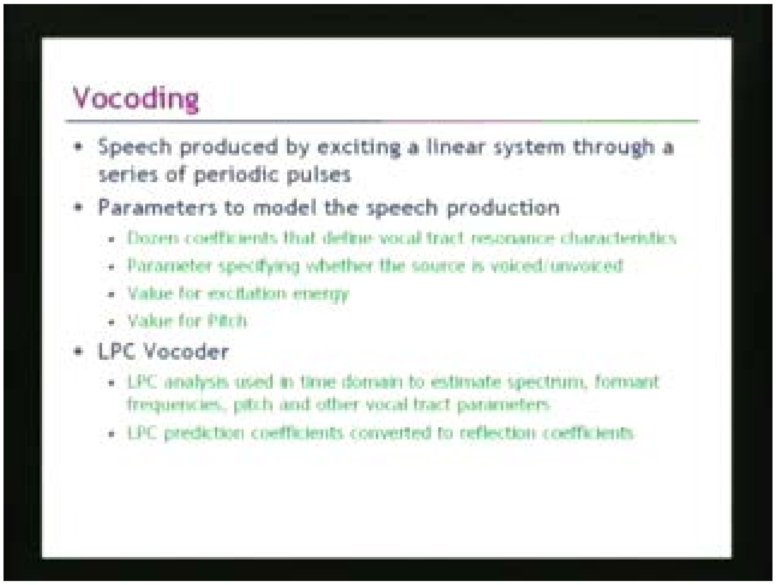
- DPCM
  - The difference between adjacent samples is quantized
- ADPCM
  - Include adaptive quantization
- DM
  - First order linear prediction
  - The sign of the difference between input sample and past sample is encoded
  - The bit is used by the decoder to increment or decrement the output by one step size

Now, the popular waveform coders are that differential pulse code modulations as it is called as the DPCM. In DPCM, typically the difference between the adjacent samples is quantized. So, DPCM is what I just told you that difference between adjacent samples is quantized. Then we can have an adaptive differential pulse code modulation. The difference between ADPCM and DPCM is that ADPCM includes adaptive quantizations and then we also have a delta modulation.

Now, delta modulations is more like a first order linear predictions and here the sign of the difference, sign of the difference between the input sample and the past sample is encoded. So, you just need one bit to represent the sign. Either it is positive, then you can represent it by 1, if it is negative, you can represent it by 0. And then, this bit is used by the decoder to increment or decrement the output by one step size.

So this way, at the receiver, depending upon the bit; the step size will be increased in the positive or negative directions and as such you will get staircase waveforms which will be closely following the original signals. So, that is what is called as the delta modulation.

(Refer Slide Time: 17:18)



Vocoding

- Speech produced by exciting a linear system through a series of periodic pulses
- Parameters to model the speech production
  - Dozen coefficients that define vocal tract resonance characteristics
  - Parameter specifying whether the source is voiced/unvoiced
  - Value for excitation energy
  - Value for Pitch
- LPC Vocoder
  - LPC analysis used in time domain to estimate spectrum, formant frequencies, pitch and other vocal tract parameters
  - LPC prediction coefficients converted to reflection coefficients

Now, so these are the popular forms of the waveform coders; either as the differential pulse code modulations or adaptive pulse code modulations or delta modulations. Now, as I have already pointed out that since the waveform coders directly work on the speech segments; the typical data rates that you can get, as you know, with pulse code modulations you can get data rate of 64 kilo bits per second and with ADPCM and DPCM you can get data rate of 32 kilo bits per second or 16 kilo bits per second kind of data rates.

But if you have to really go for low bit rate speech code, then the coding technique that is used is what is called as the vocoding or vocoder. Now, as I have already pointed out the principle of vocoder is speech synthesis through a model of the vocal tract. So, now what is done in the vocoding is that speech is produced by exciting periodic pulses through a linear system. So, this linear system actually models the vocal tract and the idea is that the voice sounds will be produced by exciting this linear system through periodic pulses and of course this period would make, the time difference between the two pulses may correspond to the pitch of the speech.

So, by excitation of the periodic pulses of this linear system which actually represents the model of the vocal tract, you can actually produce the sound. So, that is the basic principle. So, what is done at the transmitter? When the transmitter actually analyzes the speech through some kind of linear predictive filters only, so it analyzes the speech and instead of transmitting the quantized values of the speech signals or the digitized values of the speech signals, it actually transmits the coefficients of the models and also the values of the excitation signal energies.

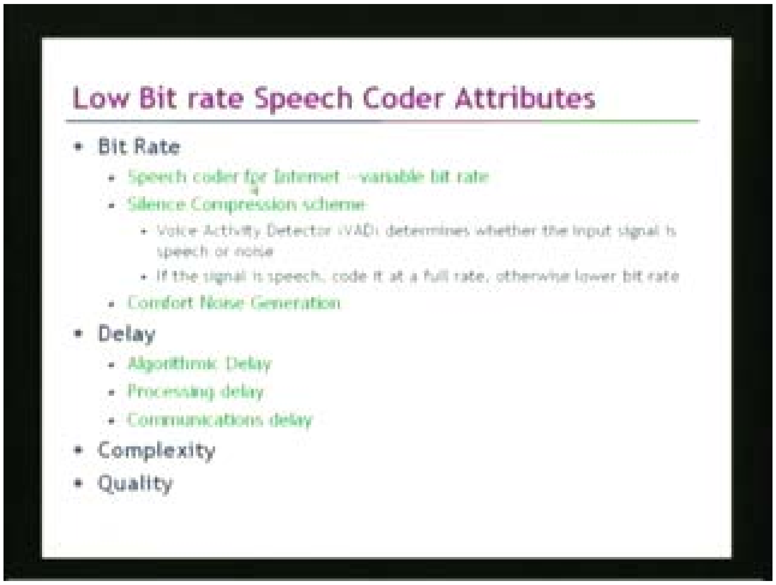
So, as a result the number of parameters or the number of bits that are required to be sent to the receiver decreases considerably and thereby we achieve a low bit rate speech coding. So, that is precisely the reason why we can get a higher degree of speech compressions by using vocoders.
So, the parameters which are used to model the speech productions are maybe about a dozen coefficients that can define about vocal tract resonance characteristics, then a parameter which is specifying whether the source is voiced or unvoiced, the values of excitation energies and so on.

So, these are the typical parameters which are sent to the receiver so that a model of the vocal tract can be constructed and the values of the excitation signal energies are also given which when fed into this model will produce exactly the same sound which was produced at the transmitter. So, this is the principle of speech production at the receiver by using vocoder kind of techniques.

Now obviously, you achieve a very great degree of compressions by using these vocoders. Now, remember that some of these parameters which have been transmitted to the receiver; some of them are more important, some of them maybe less important and as you will see that in the context of packet communications, some of these parameters which are more important maybe given a higher priority packets transmissions and parameters which are not so important maybe given as a low priority in the packet transmissions.

So, what are the, what are really the attributes of a low bit rate speech coders? As we can see that the bit rate is one of the most important.

(Refer Slide Time: 21:15)



Low Bit rate Speech Coder Attributes

- Bit Rate
  - Speech coder for Internet – variable bit rate
  - Silence Compression scheme
    - Voice Activity Detector (VAD) determines whether the input signal is speech or noise
    - If the signal is speech, code it at a full rate, otherwise lower bit rate
  - Comfort Noise Generation
- Delay
  - Algorithmic Delay
  - Processing delay
  - Communications delay
- Complexity
- Quality

So, speech coder for the internet generally will produce the variable bit rate, since we are using the compression techniques. We will also use a silence compression technique through voice activity detector which determines whether the input signal is speech, which determines whether the input signals is speech or noise.

So, that is very important by using this voice activity detector, you will determine whether the input signal is speech or noise and if it is noise, then you do not encode that part of the speech segment. So, thereby also you can result a reduction in the bit rate. So, the voice but however since you are not transmitting anything during the silence part that is during the unvoiced part. So, at the receiver it may generate some irritating feeling and therefore typically at the receiver, you will generate what is called as the comfort noise.

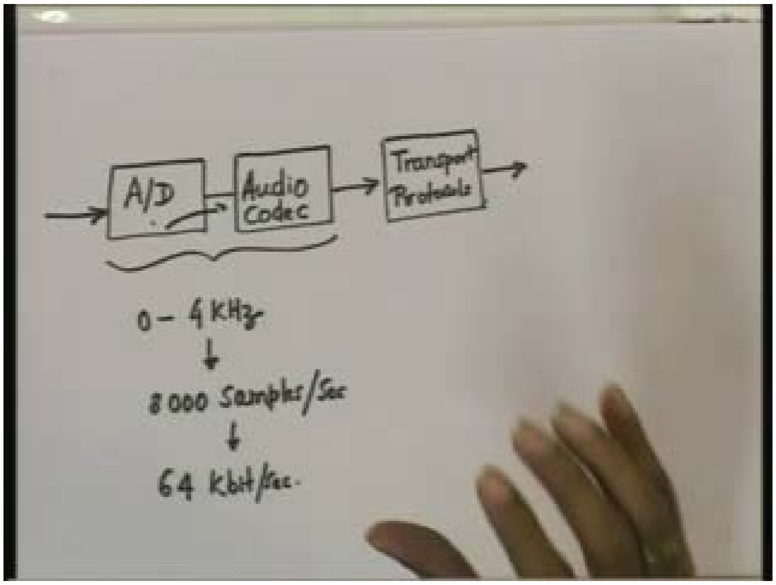
So, there will be a voice activity detector at the transmitter and at receiver, you will require corresponding comfort noise generations so that which can fill in between these silence spurts. So, the other attribute is delay. How much is the delay that is required in the speech coders? There could be several forms of delay like algorithmic delay or processing delay or the communications delay and of course, the other parameters are the complexity of the speech coders and also the quality.

So, we have already discussed; what are the attributes that should be there for the speech coder. One is of course, the bit rate is most important. We should get as low bit rate as possible, we should try to achieve a very low bit rate speech coder for the internet. But apart from the bit rate, as we have already seen that the delay which will be there as the speech codec, the quality the intelligibity of the speech and the computational complexity are also important issues.

So, there could be trade-off between these various attributes and depending upon the applications, depending upon the bandwidth which is available over the internet; one can have a choice of speech coders in a particular voice over IP context. So, now we have studied the analog to digital conversion part.

Now, let us look at what are the other important aspects of the voice over IP. So, let me just sketch brief block diagram of the voice over IP phone that may be present. So, it would be something like this that you have the speech, you got this analog to digital conversions and some kind of audio codec.

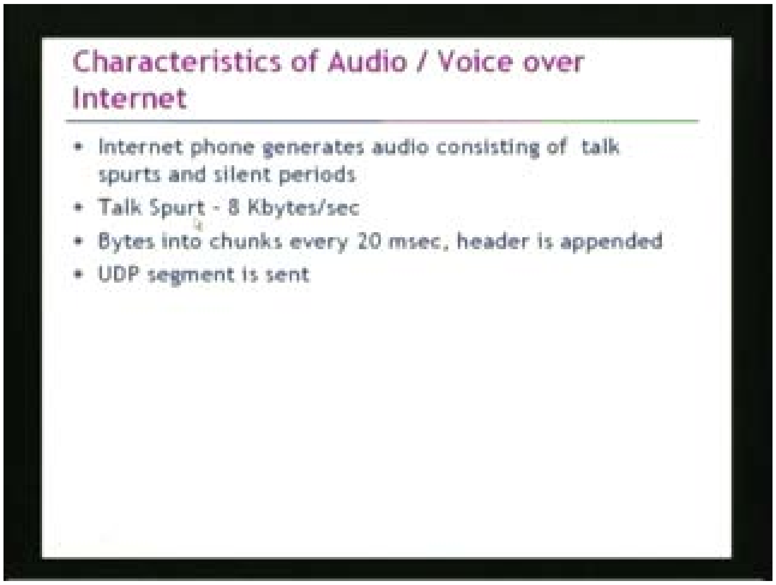
(Refer Slide Time: 24:03)



So, this is like the audio codec part. So, both is actually doing the analog to digital conversions and the compressions and the audio codec. Basically, analog to digital conversions can also be incorporated as a part of the audio codec itself. Now, once you get this, you basically are having the transport protocol for packetization because you are then transmitting these voice samples or voice digits into the packets. So, this is the transport protocols.

Now typically, you can assume that your voice is having from 0 to 4 kilohertz and let us say that you have sampled it at the rate of 8000 samples per second and then you have resulted into 8 bits for each sample and as a result you may have resulted into 64 kilo bits per second. Let us say that we are not using any compressions. So let us, for the time being assume that we are not using any compressions, we are using PCM kind of compressions.

(Refer Slide Time: 25:40)

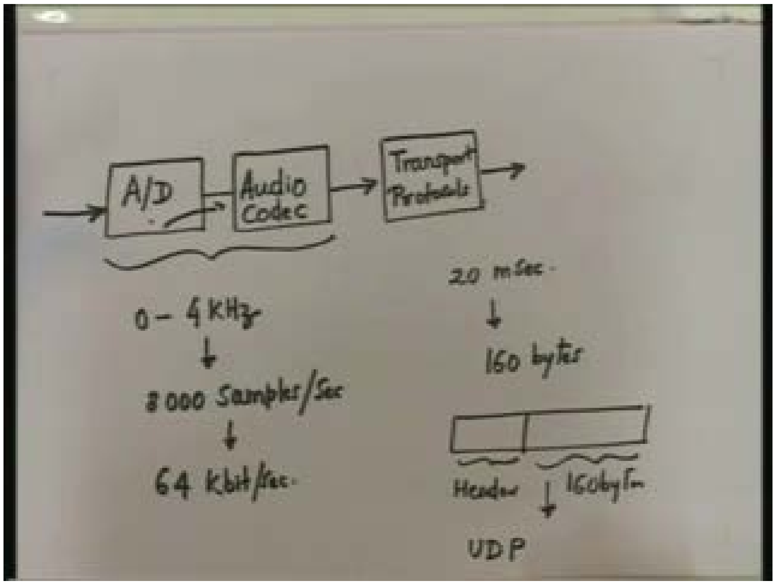


Characteristics of Audio / Voice over Internet

- Internet phone generates audio consisting of talk spurts and silent periods
- Talk Spurt - 8 Kbytes/sec
- Bytes into chunks every 20 msec, header is appended
- UDP segment is sent

So, if you do this; then, so what is typically done in the internet phone, you have the talk spurts which is encoded using 64 kilo bits per second which is like 8 kilobytes per second and then these bytes are put into the packets say every 20 milliseconds a header is appended and then we send it using UDP as the transport protocol. So, in this scenario what we see is that you will encode about of 20 milliseconds.

(Refer Slide Time: 26:05)



So, when we say 20 milliseconds and if you are having 64 kilo bits per second; then you will generate about 160 bytes of data. So, what you can do is that you can put 160 bytes here, you can append a header and then you can send using a UDP. So, this you can give it to a UDP - user datagram protocol for transmissions. Now, this will be typically the transport of the packet voice over the internet. Now, several issues come up here, we will we will take up many of the issues.

So, the first thing is that when the voice is getting digitized and when the bits are coming; whether we should take the speech segment for 20 milliseconds or 5 milliseconds or 50 milliseconds, so there are issues. Now, the issues are that if you wait for let us say 50 milliseconds, then obviously you encounter some kind of delay and because you will have to wait for that much data to get accumulated at the transmitters. So, you encounter that much of the delay. But what is the advantage?

The advantage is that when you get this payload and append a header, remember that the header or the overhead associated with the header is typically the same irrespective of the payload size. So obviously, if the payload size is more, the overhead associated with the packet will be smaller. So, ideally we would like this payload to be as large as possible in the packets. But then the downside of that is that we will have to wait for a long enough time and since we will have to wait for a long enough time, we would encounter the latencies and the delays which will be unacceptable for the voice communications.

So therefore, a trade-off needs to be achieved between the delay that would occur in accumulating the packets and the overheads that would be associated with the headers. So, typical applications, they will take the bytes for each 20 milliseconds, put them into the packets append header and then send.

Now, another question that arises is that what kind of transport protocols should be used for transmitting these voice packets; should we use TCP or should we use UDP. So, as I just pointed out that typically the transport protocol that is used in the voice over IP communication is the user datagram protocol or UDP. So, the question really is that why UDP. Why we would like to use the the UDP as a protocol? So, one thing that is important to understand that in the TCP, first of all the TCP provides you the reliable transports mechanisms and UDP provides you unreliable transport mechanisms.

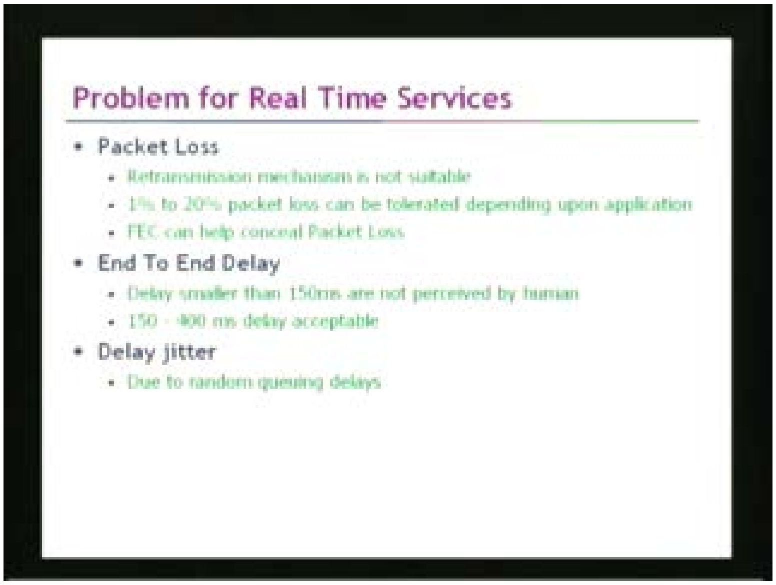
So, how does TCP provides reliable transport mechanism? Since TCP provides a reliable transport mechanism, it also looks for the acknowledgements of the packets and if the packets have been lost or if the packets have been received in error, then TCP allows for retransmission of the packets again to the receiver. Now, note that in the real time communications like voice, retransmission of packets is not possible and therefore you do not need mechanisms like retransmissions and retransmission of packets and the reliability of this kind in the voice communications. So therefore, the TCP is not used typically, the only the UDP as the transport mechanism is used.

So, if some packets have been lost in the internet, either due to congestions or due to errors; then they would be considered as lost and thereby there may be an effect in the speech quality. But you will typically not look for retransmission of those packets. So, that is why the packets are sent using UDP as a transport mechanisms. So, now what are the components of the data plain that we have discussed?

So, first we have seen that will require audio codec. So, the audio codec will actually generate a lot bit rate speech digits and then these digits would be packetized and then we sent using UDP as a transport mechanism over the internet and at the other side of course, you can receive these UDP packets, you can collect those depacketize those packets, collect those bits, convert them into speech signals again and then play them out. So, this is the normal process.

But is it really then possible to achieve an acceptable voice over IP communication by using this kind of mechanisms? What are the issues that are involved and how to improve the speech quality? That is what will be the part of our discussions that we will do in the in the next few minutes. So, now let us look at what are the real problems for the real time services.

(Refer Slide Time: 31:21)



Problem for Real Time Services

- Packet Loss
  - Retransmission mechanism is not suitable
  - 1% to 20% packet loss can be tolerated depending upon application
  - FEC can help conceal Packet Loss
- End To End Delay
  - Delay smaller than 150ms are not perceived by human
  - 150 - 400 ms delay acceptable
- Delay jitter
  - Due to random queuing delays

If you try to transmit it like the way which I have just explained, so one thing is that so remember one thing we are assuming right now that these packet transmissions or voice over IP or internet telephony which we are doing it over the internet, our assumption is that our internet is not offering any quality of service guarantees.

So, we are assuming that the packet switch network that we have got does not offer any quality of service guarantees. It is the best effort networks. So, that is our assumption and what we are saying is that over these best efforts networks, we are trying to achieve the transport of these real times services using the mechanisms that are there at our disposals. So, that is what we are saying.

Now, assuming that the internet is the best effort networks or is a non QoS networks, then what are the problems that would occur for the real time services. So, we will look at that and then we will see how we can address those problems.

Now, as you see that the packet loss can occur because the network in a non QoS network. But as we have just already pointed out that retransmission mechanism is not suitable. Now typically, in a best effort network; when the packet loss occurs, the packet loss is taken care by using suitable retransmission mechanisms. But as we have just observed for the real time services, a retransmission mechanism is not suitable.

But the good thing is that in this packetized voice communications, 10 to 20 percent of the packet losses can be tolerated depending upon the applications. So, that is a good part of this that 10 to 20 % of the packet loss can be tolerated. It will not affect or degrade the speech quality as much and therefore really speaking, somehow if the load on the internet is light, if it is not heavily loaded and if the packet losses can be contained to not a very significant fraction of the total packets transmitted; then it should be possible for us to have a very good intelligible speech without having appreciable degradations.

So, that is the thing that we should hope for. But even if there is some packet loss which is above this, then I have already pointed out that since retransmission mechanism is not suitable, the mechanism that will be used for voice over IP applications will be forward error correcting codes or FECS. Typically, forward error correcting codes will be used to conceal these packet losses. I will describe it in detail how FECS can be used to conceal the packet losses. But what we are saying is that let us say that if there is a packet loss, then we will have mechanisms like FECS to conceal these packet losses.

Now, second thing that problem arises is of the end to end delay. Now, as you know that delays smaller than 150 milliseconds, they are not perceived by the human beings. So, that is another good thing that if the end to end delay is of the order of 150 milliseconds; then typically, they are not perceived by the human beings. Depending again upon the applications, delays in the range of 150 to 450 milliseconds may be tolerable.

So, while the packet loss can be contained by using forward error correcting codes, the delays even if there are packet delays of the order of 150 to 450 milliseconds depending upon the applications; if it is a streaming audio kind of applications, then larger delays like 450 milliseconds may be acceptable. If it is a two way interactive phone conversations, then lower delays like 100 milliseconds or 150 milliseconds, they are desirable.

So, the end to end delays may not be a great problem if these delays are again within acceptable or tolerable limits. Now, assuming that the internet is lightly loaded, let us say even though internet is not offering any quality of service guarantees, suppose that the internet is lightly loaded and therefore packet losses are contained to let us say 10% and let us say the delay is less than 150 milliseconds; then the question that we should ask ourselves is is it possible to have acceptable and good voice over IP conversations over such a non QoS enabled internet.

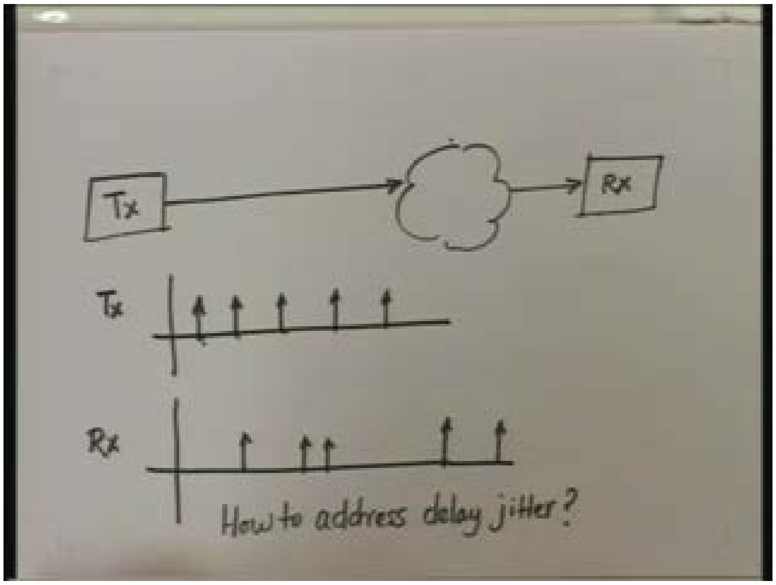
Now, can we do like this as we assumed that we digitize the speech using audio codec, packetize it and then transmit it using an UDP segment over this best effort but lightly loaded internet? Is it possible to have that acceptable voice communications? Unfortunately, the answer is that no. Why? It may be possible to have the end to end delays smaller than 150 milliseconds that may be possible to have smaller than 150 milliseconds.

Unfortunately, another serious problem that arises is due to the fact that different packets will experience different delays. If all packets are experiencing the same delay of 150 milliseconds, then that is not a problem. Unfortunately, since different packets are likely to experience different delays; that results in the problem of delay jitter and this delay jitter actually degrades the speech quality. So, that is one of the greater issues that we need to address that how to address the problem of the delay jitter in the voice over IP communication.

So now, we will actually see that since the retransmission mechanism is not possible in the packet loss; how to address the problem of packet loss in the voice over IP conversation, how to address the problem of the delay jitter which occurs due to random queuing delay and different packets experiencing the different delays? So, now we will address all these problems.

So, now let us look at how to address each of these problems individually or separately. So, let us look at first the problem of the delay jitter. So, we will see how the problem of the delay jitter can be addressed. Now, as you know that the problem of the delay jitter is that it happens because at the transmitter so and this can be of course through the internet and then we have the receiver. At the transmitter, the packets may have been generated in certain order. So, this is at the transmitter.

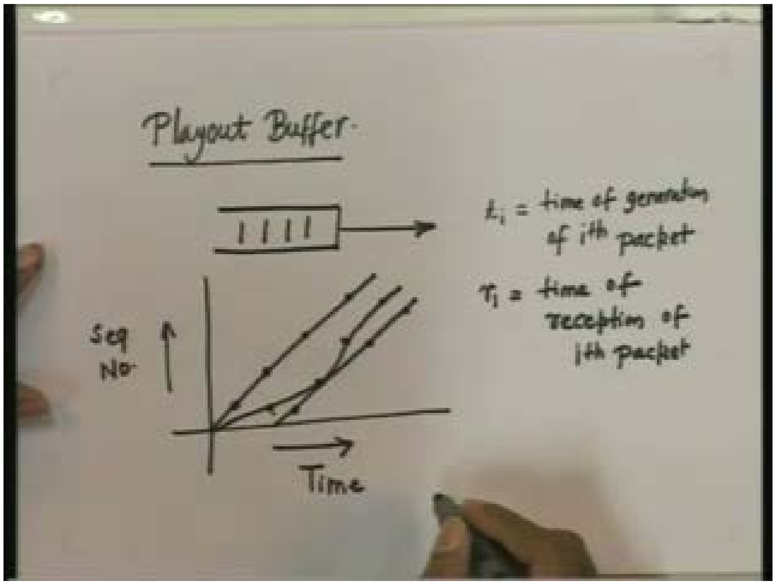
(Refer Slide Time: 37:41)



However at the receiver, since each packet is received at a receiver with different delay; then this packet may, for example may be received here, this packet may be received here, this packet may be received here and this packet may suffer a large delay and similarly this packet may suffer this delay.

Now, as you can see here that the periodicity with which the packets were generated at the transmitter, that periodicity is completely lost in the receiver here. This periodicity as you can see is completely lost. So, if these speech segments are given to the receiver for play out, then the speech quality will be definitely degraded. So now, the question really is that how to address the problem of the delay jitter. So, the question really is that how to address this delay jitter problem. Now, the answer to the question of the delay jitter is that we can use at the receiver what is called as the playout buffer. So, the solution for this is to use the playout buffer.

(Refer Slide Time: 39:37)


Playout Buffer
t_i = time of generation of ith packet
r_i = time of reception of ith packet
Seq No.
Time

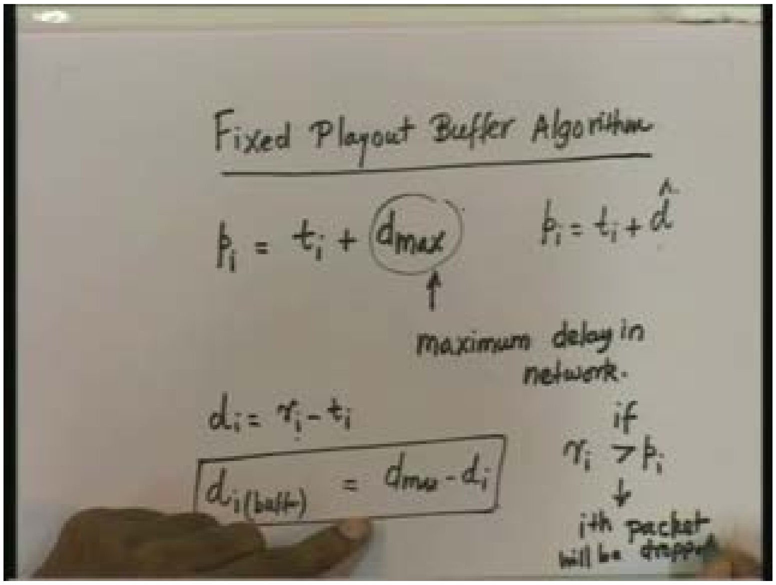
Now, what is a playout buffer? The playout buffer is actually a buffer which is used at the receiver where the arriving packets are stored and then these packets are played out at an appropriate time. So, the packets are allowed to wait in the buffer for some amount of time and then played out at appropriate time. That is why that is called as the playout buffer.

Just to give you an example of fixed playout buffer algorithms. So, the fixed playout buffer algorithm will work something like this that let us say that this is I denote this with a time and this with let us say sequence number. Let us say the transmitter; the packets are generated like this. Let us say the first packet is generated here, the second packet is generated here, the third packet is generated here and so on and let us say that these packets, they arrive at different times. So, this is the time of arrival.

Now, what you can do is that in a fixed playout buffer algorithm, you can actually start playing the packets like this. So, the first packet you play it here. So, this packet which was transmitted at this time, it arrived here but it was played here. So, this packet has to wait in a queue for this much amount of time. This packet does not have to wait because it really suffered a large amount of delay in the network. This packet which was transmitted here, it arrived here. This packet may have to be waiting for this much time and similarly for this packet and so on.

So, this way each packet can wait some amount of time, appropriate amount of time in the playout buffer and then the packets can be played out. If you really see that let us say that t_i is the time of generation of i'th packet, let us say that t_i is time of generation of i'th packet and r_i is the time of reception of i'th packet, then as you can see here that in the fixed playout buffer algorithm, in the fixed playout buffer algorithm, as you can see the packet p_i will be played at the time t_i plus d_{max} where d_{max} happens to be the maximum delay in the network.

(Refer Slide Time: 42:33)



Fixed Playout Buffer Algorithm

$p_i = t_i + d_{max}$ $\qquad p_i = t_i + \hat{d}$

↑ maximum delay in network.

$d_i = r_i - t_i$

$\boxed{d_{i(buff)} = d_{max} - d_i}$

if $r_i > p_i$
↓
ith packet will be dropped

As you can see, actually the i'th packet suffered a delay d $_i$ equal to r $_i$ minus t $_i$. This was the time at which i'th packet arrived and this was the time at which the i'th packet was transmitted. So, d $_i$ is equal to r $_i$ minus t $_i$ was the delay of the i'th packet. The playout time of the i'th packet is t $_i$ plus d $_{max}$ and therefore the packet, the i'th packet has to wait in the buffer for the amount of time. If I write d $_i$ which is buffer which is equal to d $_{max}$ minus d $_i$; it had to wait in the buffer for this much amount of time. So, different packets will have to wait in the buffer for the different amounts of time.

So, now as you can see here that one thing that is important is that you need to know the bound on the maximum delay that is the d $_{max}$; that needs to be known. Now, in actual practice this value of d $_{max}$ may not be may not be known in a network which provides quality of service guarantees, it would be desirable to give a quality of service in terms of the maximum delay that the network can provide.

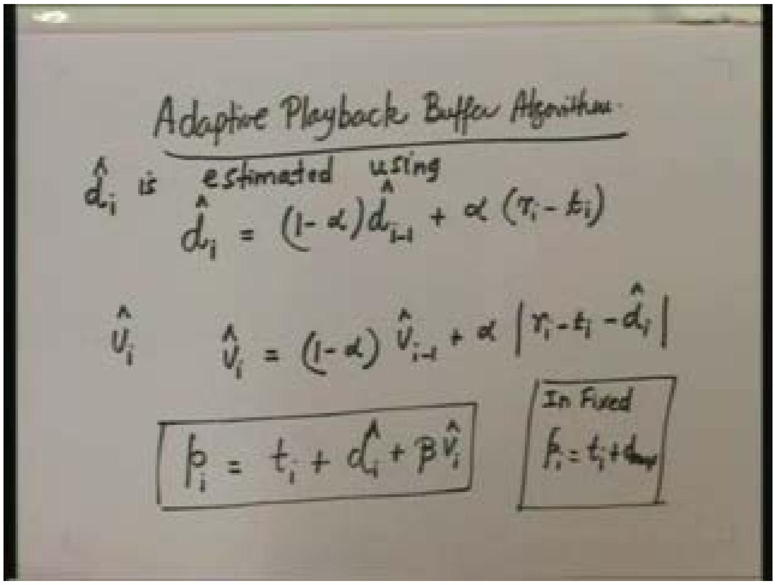
So, however as we have seen in practice, it may not be possible to do that and therefore moreover this value of d $_{max}$ may not be known at the receiver. So, one of course can take a conservative estimate and one can take the value of d $_{max}$ to be very large. Now, the problem here however is that if you keep the value of d $_{max}$ to be very large, then what really happens is that it increases the latency because the packets then really have to wait in the playout buffer for the corresponding amount of time till their time to playout comes.

So, this may actually introduce latency. While this may not be a serious problem in the streaming audio applications, actually this may become a significant issue in the two way interactive phone conversations. So, what is alternative? And the alternative is that instead of using this d $_{max}$ instead of using this d $_{max}$, one must use some other values maybe p $_i$ is equal to t $_i$ plus some d hat, some d value it should be used. The question really is that what should be the value of this d hat that needs to be used. If d hat is kept equal to d $_{max}$ for a very large value, then it increases the latency.

Now, let us suppose that if I keep the d hat value to be some value and i'th packet arrives or suffers a larger delay in the network and therefore it arrives at a time r $_i$ where r $_i$ happens to be larger than p $_i$. So, if the i'th packet arrives such that r $_i$ happens to be larger than p $_i$, then i’th packet will be dropped. The i'th packet will be dropped if it arrives after its playback time.

Now, there is a trade-off here, the trade-off here is in between the latency and the packet lose rate. Somehow, if we can adaptively adjust the value of the delay d hat in such a manner that the number of packets which arrive after their scheduled playback time if that is minimized, at the same time, this value of this d hat is not that large such that the latency is not affected; then we can have a large number of packets being played out and having not much degradation in the speech quality and also having an acceptable latency. So, those algorithms are actually called as the adaptive playback buffer algorithm.

(Refer Slide Time: 47:07)



Adaptive Playback Buffer Algorithm

$\hat{d}_i$ is estimated using

$$\hat{d}_i = (1-\alpha)\hat{d}_{i-1} + \alpha(r_i - t_i)$$

$\hat{v}_i$

$$\hat{v}_i = (1-\alpha)\hat{v}_{i-1} + \alpha\left|r_i - t_i - \hat{d}_i\right|$$

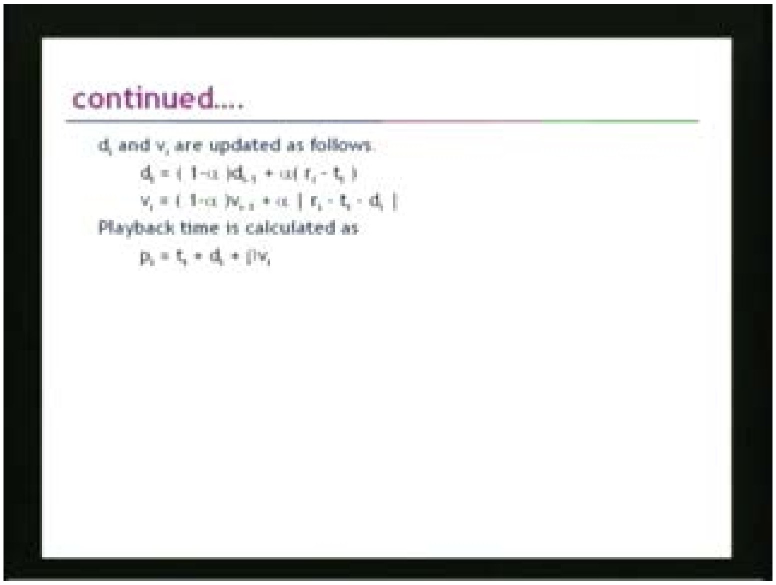
$$\boxed{p_i = t_i + \hat{d}_i + \beta\hat{v}_i}$$

In Fixed

$p_i = t_i + d_{max}$

So, what is done in the adaptive playback algorithm is the delay that the estimated delay is used instead of the instead of the maximum delay. So, I will just present one adaptive playback buffer algorithms that have been used in the literature.

(Refer Slide Time: 47:35)



continued....

d_i and v_i are updated as follows

$$d_i = (1-\alpha)d_{i-1} + \alpha(r_i - t_i)$$

$$v_i = (1-\alpha)v_{i-1} + \alpha \mid r_i - t_i - d_i \mid$$

Playback time is calculated as

$$p_i = t_i + d_i + \beta v_i$$

So, that adaptive playback algorithms works like this that you estimate the delay d_i hat for the i’th packet to be 1 minus alpha of d_i minus 1 hat into alpha of r_i minus t_i - this is the actual delay. So, this is like what we are doing is that the delay d_i hat is estimated using this equation. So, this is really a moving average. So, this is a low pass filtered versions and as you can see here that alpha is a constant which lies between 0 and 1.

If it is kept close to 1, then this d_i hat is more a reflection of the current delay. On other hand, if this alpha is close to 0, then this actually filters out any temporary fluctuations in the delays. So, this is how d_i hat is estimated. You also have the variance v_i hat. If the variance v_i hat is estimated as 1 minus alpha v_i minus 1 plus alpha into r_i minus t_i minus d_i hat. So, this is like the variance and then the playback time p_i is given as t_i plus d_i hat plus some beta times the variance.

So, instead of using, as we have seen that in the fixed playback buffer algorithms, what we were doing it in the fixed playback buffer algorithm, so we are doing p_i is equal to t_i plus d_{max}. So, instead of saying t_i plus d_{max}, we are using p_i plus d_i hat plus beta times v_i hat. This is the estimated values of the delay and of course, the constant times the variance we are using. So, what we are really doing is we are estimating the delays and the variance of the packet delay and adaptively adjusting the playback times. The objective of course is to minimize the packet loss and also to a sort of minimize the latencies.

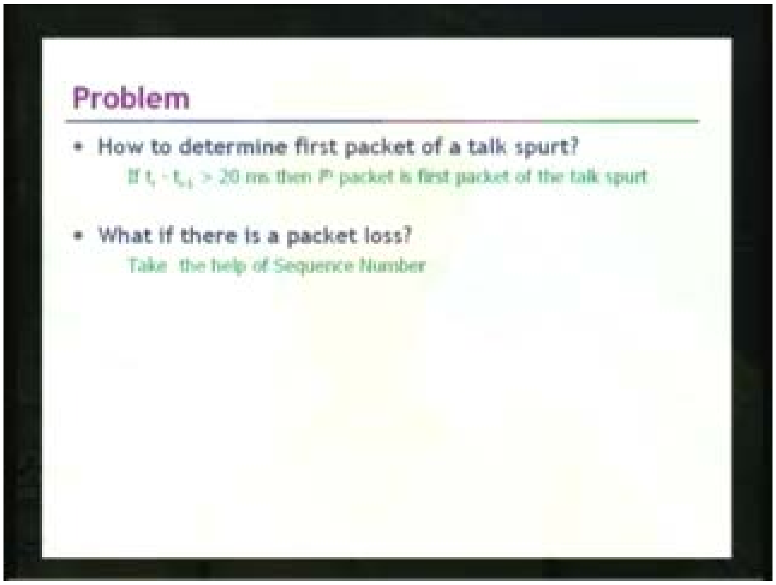
Now, the question really arises now, as we have seen here; so again to recapitulate our voice over IP thing, what we have said is that we will digitize the voice, we will compress the voice, we will have an audio codec, we will packetize the voice maybe by collecting the speech segments of 20 milliseconds and after doing this, we will transmit it over the UDP segments over a best effort internet.

But as we have already seen, since we transmit it over a best effort internet, we will be faced with a problem of combating this delay jitter and to combat the delay jitter, we will have to queue those packets in the playback buffer at a receiver and then play them out using an appropriate playback buffer algorithms.

Now, as we have seen in both the algorithms, we consider two playback buffer algorithms; one was the fixed playback buffer algorithms and another one was the adaptive playback buffer algorithms. And, what we have seen in the fixed playback buffer algorithms, we may require an accurate knowledge of the d_{max} or maximum delay; in the adaptive playback buffer algorithms, we can adaptively adjust the playback time so as to minimize the packet loss and the latencies.

But in both these cases, as we have actually seen what really we need is that we need to know the time of generation of the packet, t_i needs to be known, as we have seen the t_i should be known. So, then the question really is that we need to timestamp the packets. So, that is one aspect that we need to sort of timestamp the packets.

(Refer Slide Time: 51:44)



Problem

- How to determine first packet of a talk spurt?

  If $t_i - t_{i-1} > 20$ ms then i^{th} packet is first packet of the talk spurt

- What if there is a packet loss?

  Take the help of Sequence Number

Another thing that would arrive here is that as that we will typically in the playback algorithms, in the adaptive playback algorithms, we will determine the playbacks time of the first packet of the talk spurt only. The later on of course, you can determine as how these packets are spaced between the talk spurts. So, in adaptive playback buffer algorithms, typically we will adjust the playback time of the first packet only. So now, suppose if some packets are lost, so how will you determine that it is the first packet if the packet has been lost? So, we will determine that is t_i minus 1, if it is greater than 20 milliseconds then i'th packet is the first packet of the talk spurts.

So, what if there is a packet loss? So, if there is a packet loss, then we would require sequence numbering of the packets and we will take the help of sequence number. So basically, what we are saying is that in the transport protocol, we need time stamping the packets and we also need sequencing of the packets. Then the question is that we need some more information other than what is available with the UDP. So, that means we require a separate transport protocol which would enable the transport of the packet voice and that separate transport protocol is called as real time protocol or RTP. So, we will study in the next lecture the RTP and then how to carry the packets using RTP and other signaling aspects of the voice over IP.