INFORMATION CODING
Linköping University

TSIN02 Internetworking

Lecture 8 – Source coding and channel modeling

Outline

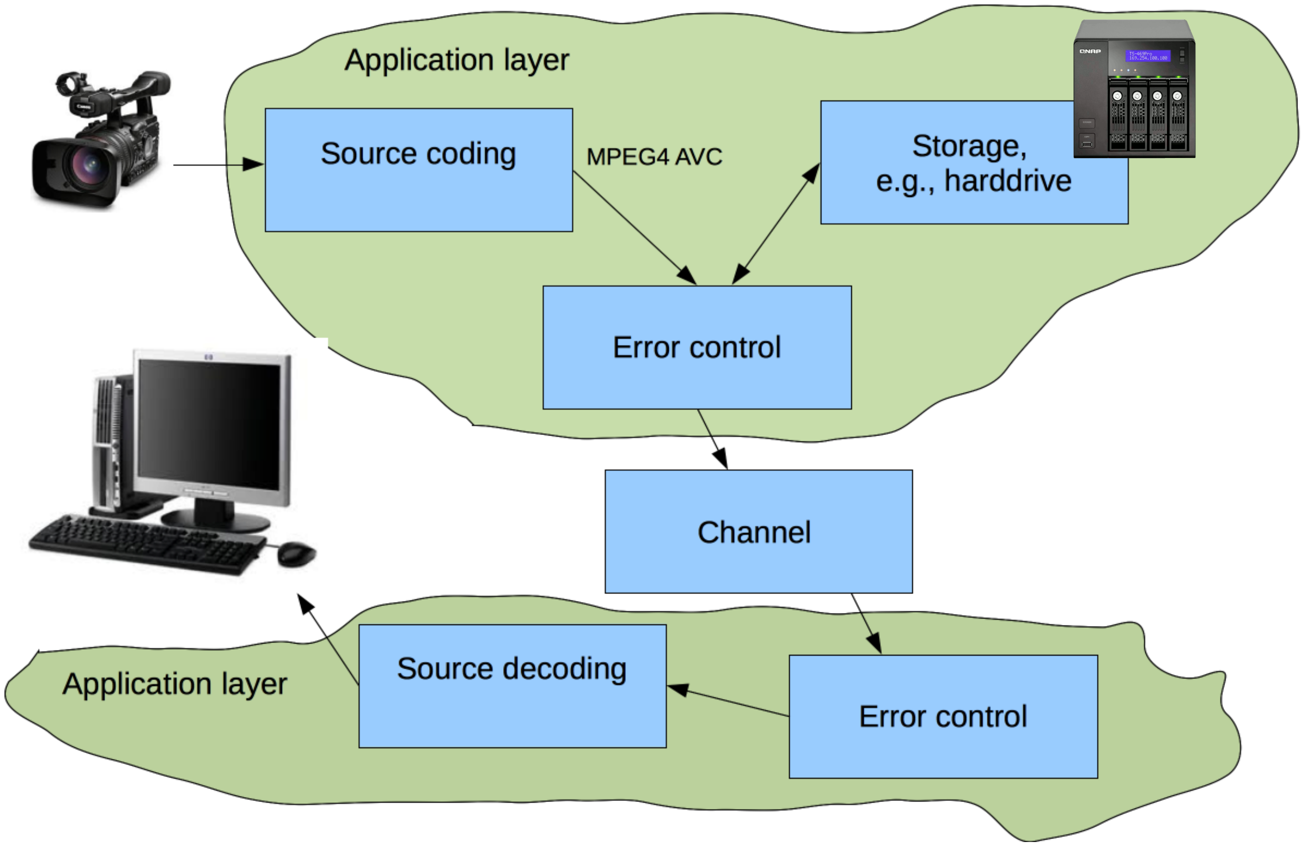
- Sampling and quantization
- lossy vs lossless compression,
- Distortion, MSE, SNR
- JPEG, MPEG/H.265
- Shannon limit for signal sources
- Channel models
- Shannon limit for channels
- The “best-effort” channel
- Gilbert-Elliot packet loss model
- “Transforming” the best-effort channel
- Error concealment
- Prioritized services, DiffServ, IntServ

Questions to be dealt with

- We do not want to send unnecessary data for our application => new concept: source coding (representing the information with less bits)
- The network leads to packet losses (congestion, router packet drops) and delay

How do we handle these issues?

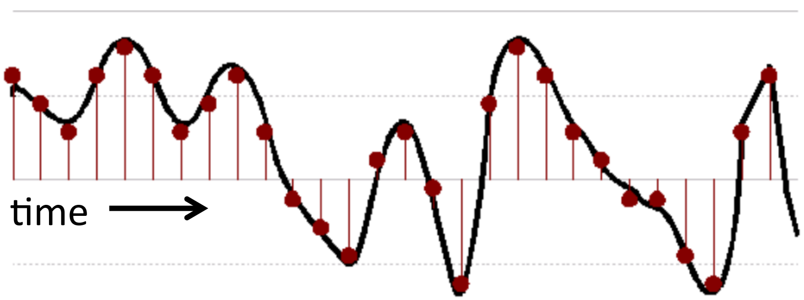
Video example



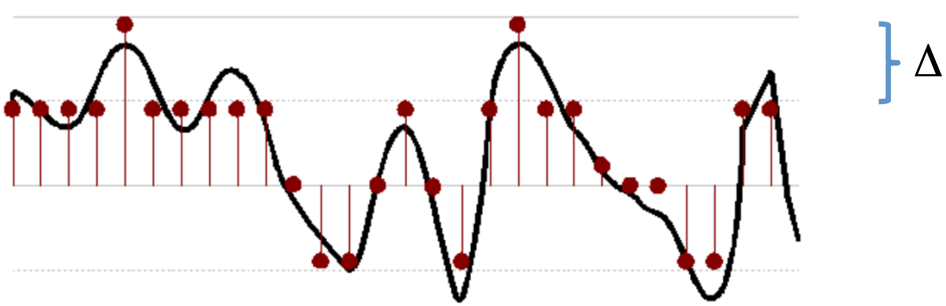
Source coding
(short introduction)

1. Assume a time-continuous, amplitude-continuous signal. **Sample** it to produce a discrete time, amplitude-continuous signal

Speech
music, video,
sensor readings…



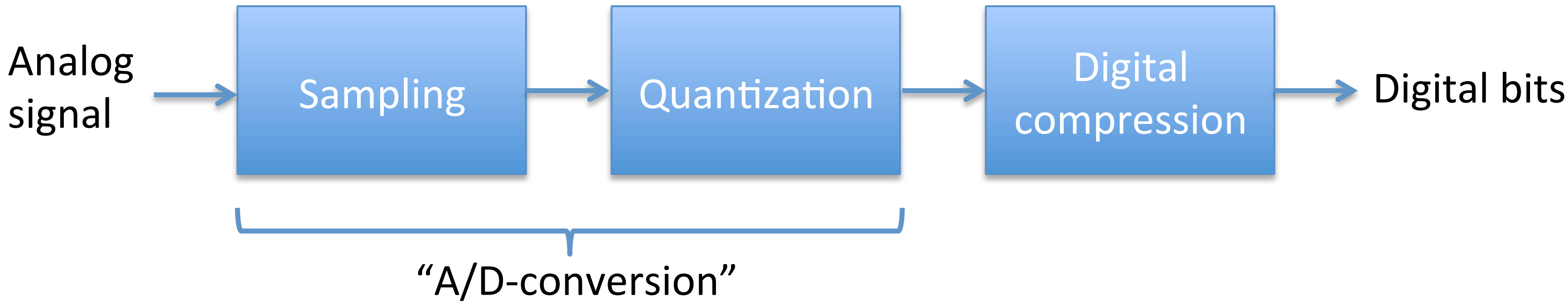
2. Round off amplitudes to discrete values - **Quantization**



3. Use **digital compression** to efficiently represent the amplitudes

Source coding cont'd

- Thus, "natural signals" e.g. audio, video which are analog, are turned into efficient digital signals through a 3-step procedure.
  - Sampling, quantization, digital compression



- Sampling – (practically) distortion-free
  - Sampling rate must be higher than 2*bandwidth of the signal (Nyquist rate)
- Quantization – introduces quantization error (distortion)
- Digital compression
  - lossless: invertible
  - lossy: non-invertible (distortion)

Quantization error

Distorsion, MSE, SDR

- It can be shown that the distortion D (defined as the *mean-squared-error, MSE*) due to the quantizer becomes

  $$D \approx \Delta^2/12 = 1/12\,(A/M)^2 \quad \text{where}$$

  Δ is the quantizer step (uniform quantization), A is the total range of the signal and M is the number of levels in the quantizer.

- For each additional bit in the quantizer, the step size is divided by 2, and thus the distortion is lowered by a factor of 4 (corresponding to 6 dB)
- This leads to the following formula for the *signal-to-distortion ratio (SDR)**: $SDR \approx 6R - const$ [dB]

  R: bits per sample, const ≈ 7.4 for uniform quantizer and Gaussian distributed samples)

***SDR** is also often denoted **SNR** (signal-to-noise ratio)

Example: Audio coding

- Telephony
  - Analog bandwidth: 3.4 kHz
  - Sampling rate: 8000 samples/s
  - Number of quantized values: $2^8 = 256$
  - Thus, a data rate of 64 kb/s
  - *SDR* ≈ 45 dB after A/D-conversion (non-uniform quantizer!)
  - After lossy digital compression (e.g. GSM) 2 – 6 kb/s

- CD music
  - Analog bandwidth: 20 kHz
  - Sampling rate: 44100 samples/s
  - Number of quantized values: $2^{16} = 65536$
  - Thus a data rate of 44100*16*2 = 1.411 Mb/s (stereo)
  - *SDR* ≈ 89 dB after A/D-conversion (uniform quantizer)
  - After lossy digital compression (e.g. AAC): 96 – 256 kb/s

Digital compression – lossless example

Assume our signal samples are quantized to 4 levels (A1..A4) and that these appear with the following probability:

A1 : 80%,

A2: 10%

A3: 5%

A4: 5%

Let us compare two binary coding schemes (fix-length vs variable-length):

A1: "00"	A1: "0"
A2: "01"	A2: "10"
A3: "10"	A3: "110"
A4: "11"	A4: "111"

Which one is better?

The number of bits per sample for the first scheme is 2. The number of bits per sample **on average** for the second is 1.3 thus a *compression factor* of 2/1.3 = 1.54. (see whiteboard).

INFORMATION CODING
Linköping University

Digital compression – lossy example

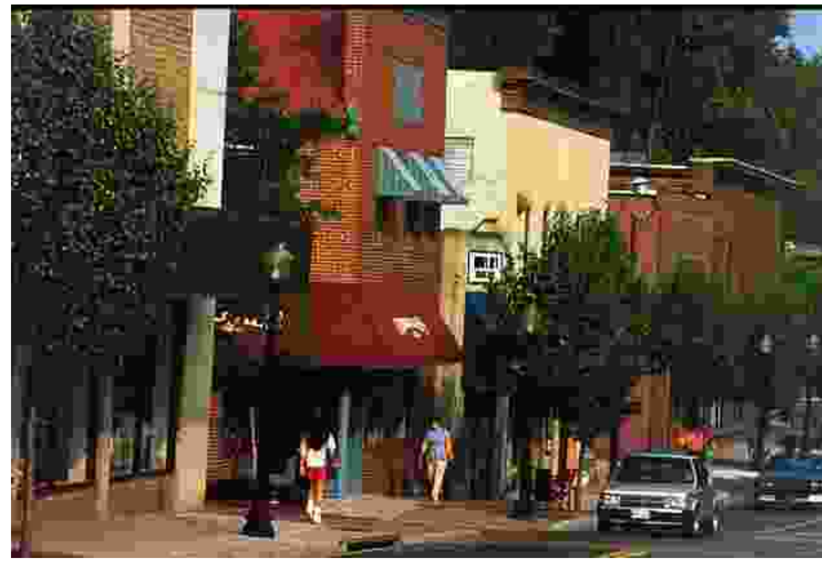
Image compression: the image is not recreated perfectly

Original

Joint Photographic Experts Group (JPEG) reconstruction,
¼ of the number of bits describing the original

Image compression cont'd

Original

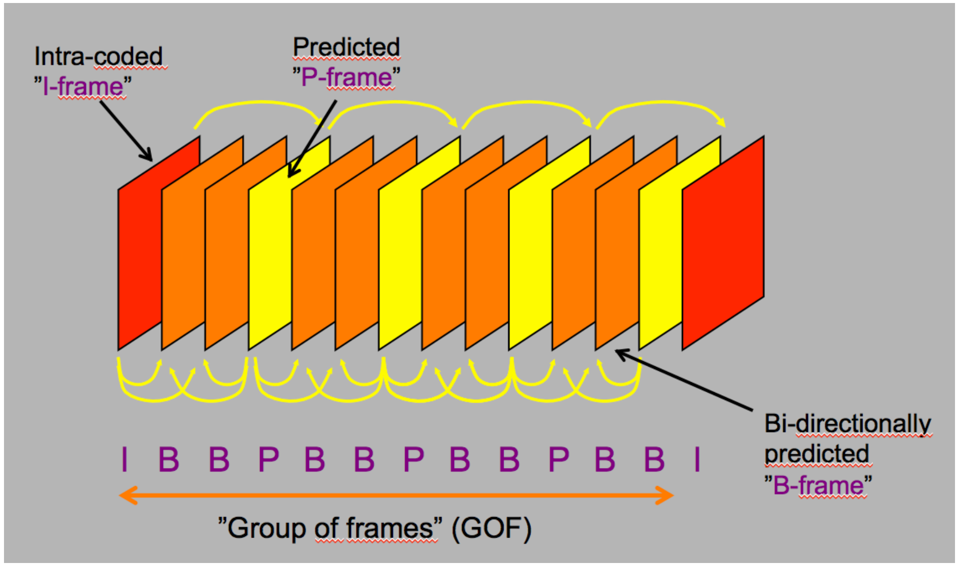


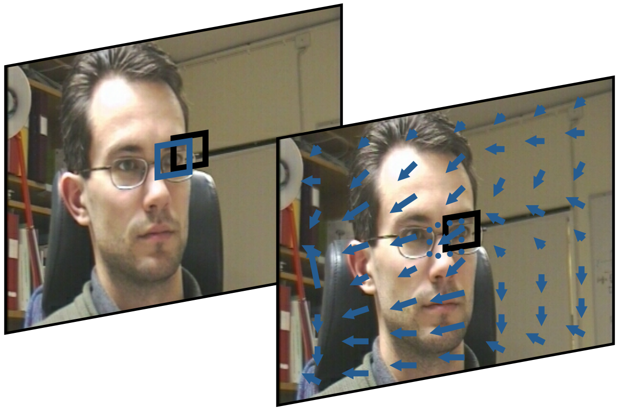
Joint Photographic Experts Group (JPEG) reconstruction,
1/25 of the number of bits describing the original

INFORMATION CODING
Linköping University

Video compression

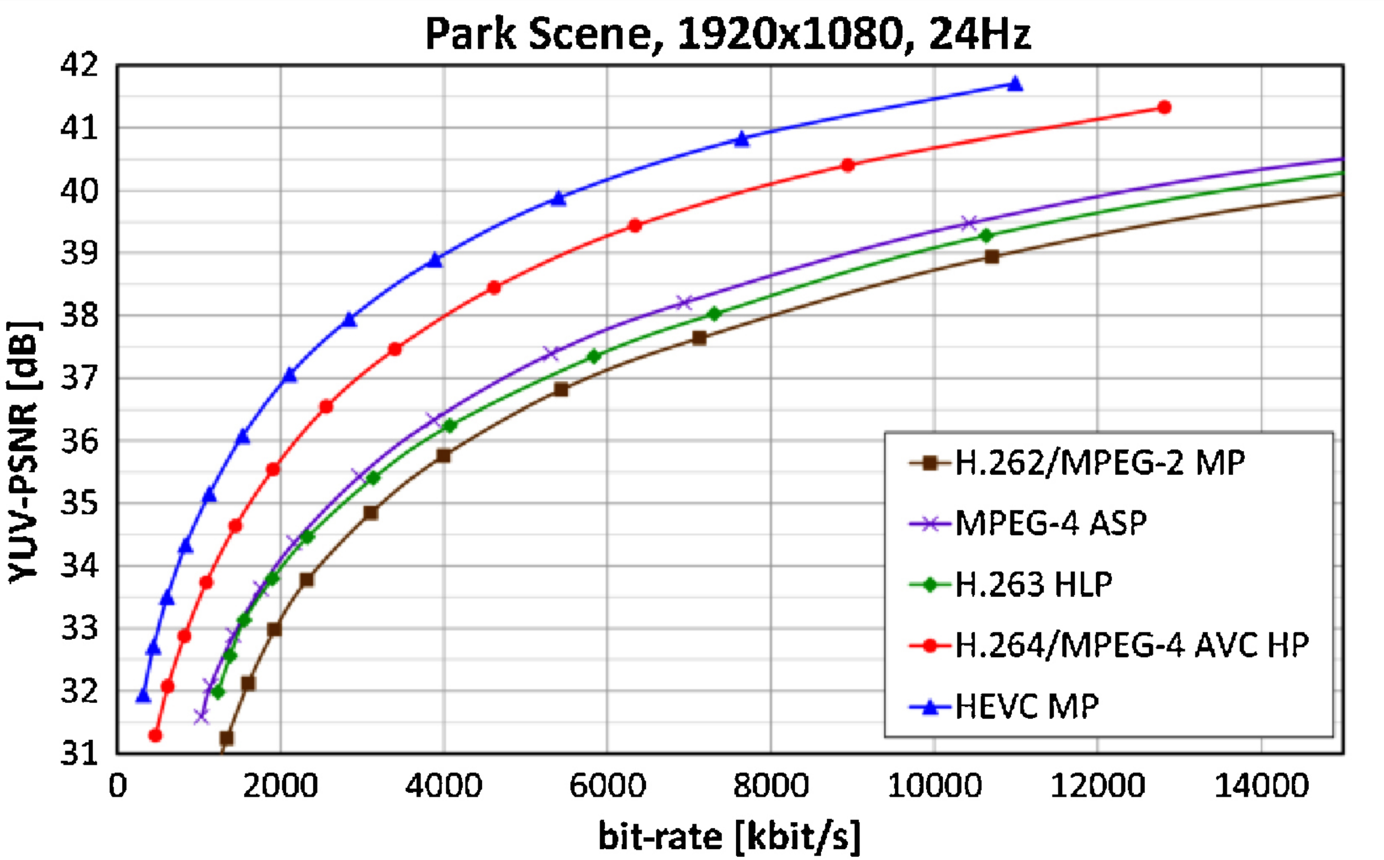
- Most common standard is H.264 (a.k.a MPEG-4 AVC)
- Being replaced now by the new ISO/ITU standard: H.265 (a.k.a.HEVC = High Efficiency Video Coding")
- Basic principle is to transmit the difference between the prediction of the frame and the frame itself
- Most simple prediction: previous frame => send difference between the two frames.
- Better prediction is obtained by motion estimation (performed on blocks of pixels). The motion vectors need to be transmitted as well
- Prediction is done both forward and backward (B-frames).





INFORMATION CODING
Linköping University

H.265 – example performance



PSNR = “Peak SNR”. Good quality when > 40 dB

Shannon limits for signal sources

- Claude Shannon developed a theory (*Information Theory, 1948*) through which he was able to compute the theoretically lowest number of bits needed to represent an analog or digital signal source.

- For an analog (Gaussian) source with equally strong frequencies within the bandwidth B and a tolerable signal-to-distortion (S/D) ratio, he shows that the lowest number of bits per second is,

$$R_s = B \cdot log_2(S/D) \approx 0.33 B \cdot SDR \qquad (SDR = S/D \text{ in dB})$$

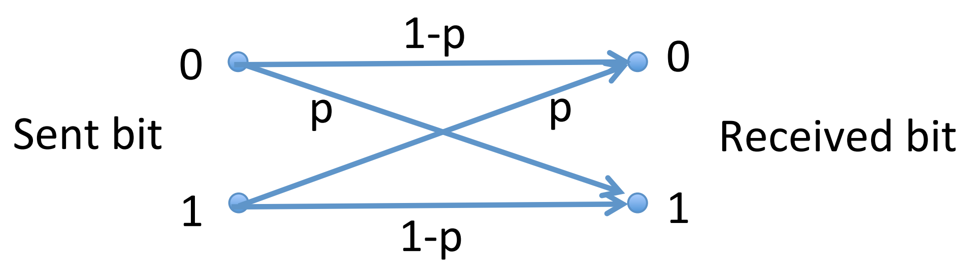
- Furthermore, for a digital signal (with symbols having a discrete number of values) where the values have the probabilities p_i and the symbols are statistically independent and identically distributed (i.i.d), the least number of bits per symbol is,

$$R = - \Sigma p_i \, log_2 p_i$$

- To achieve the Shannon limits (the *entropy*) it is required theoretically that messages are infinitely long. In practice the theory works also for limited length messages.

Channel models

- Transmission channels such as radio, copper and fiber links are characterized by random bit errors, which are typically independent of each other.
  - A common mathematical model is the Binary Symmetric Channel (BSC) model with error probability p:



  - The error probability from 0 to 1 is the same as the error probability from 1 to 0 (i.i.d. transitions).

- Packet networks are “best effort channels”

Shannon limits for channels

- Similar to Shannon limits for data compression, there is a maximum value of message bits per channel bit that a channel can communicate reliably.
- Assume a channel that is capable to transmit binary bits where bit errors are independent of each other and occur with the probability p (BSC model).
- The Shannon limit (*channel capacity*) for this channel is:
  $C = 1 - h(p)$ where $h(p)$ is the binary entropy function: $h(p) = -p\log p - (1-p)\log(1-p)$
- Provided that arbitrarily long messages are allowed (arbitrarily long delay), the message is transmitted without error!
- Basically, Forward Error Correction (FEC) is the technique used to handle the errors and obtain the capacity. (FEC was covered in Lecture 2).
- Compare with packet channel with ARQ (automatic repeat-request) which gives a channel capacity of $C = 1\text{-}P$ where P is packet loss probability.
- A detailed comparison (FEC vs ARQ) requires that the relation between bit error probability p and packet error probability P is known (depends on the channel model). E.g. for a BSC we have that $P = 1\text{-}(1\text{-}p)^N$ where N is the packet size.

The “best-effort” channel

- Packets can get lost
- Packets may not be delivered in sequence order
- Packets can get delayed
- Delay can vary from packet to packet (jitter)

Packet loss

- Packet loss can occur due to,
  - Transmission error
  - Buffer overflow in router
  - Packet is delayed past "realtime"

Packet loss measurements

- J. Bolot, “Characterizing end-to-end packet delay and loss in the Internet”, Journal of High-Speed Networks, 1993:

  Packet losses are independent if the traffic uses a small fraction of the available bandwidth

- V. Paxson, “End-to-End Internet Packet Dynamics”, IEEE/ACM Transactions on Networking, 1999:

  An order of magnitude difference in packet loss probability if the foregoing packet was lost

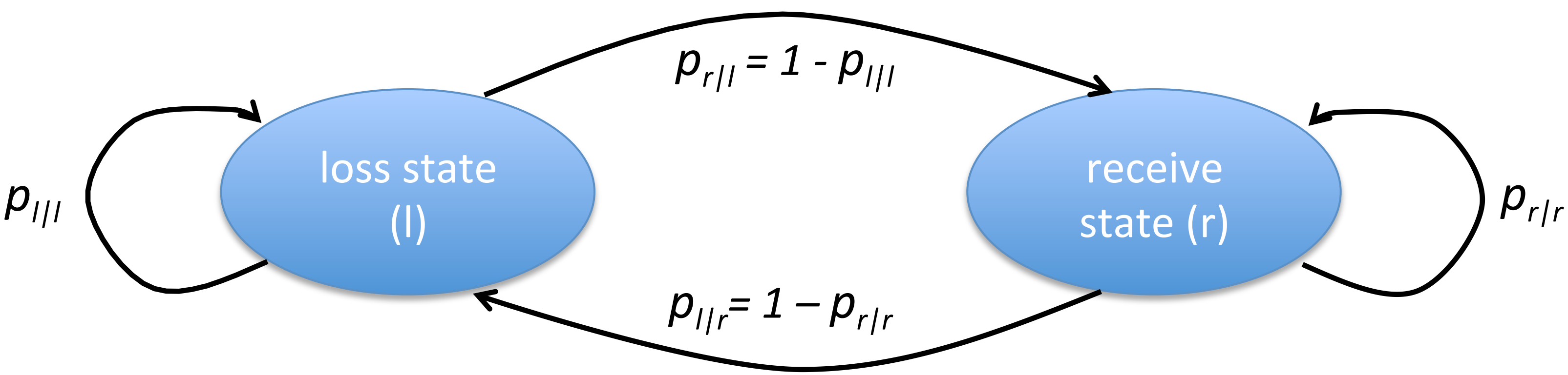
- M. Yalnic et al., “Packet loss correlation, in the Mbone multicast network”, GLOBECOM Conference, 1996:

  A majority of the losses occurs independently, but some very big bursts of losses were observed. Most losses are on access networks

Network settings are different. Difficult to draw conclusions. However, both independent losses and loss bursts seem to be important to model

Loss bursts – The Gilbert-Elliot model

- The G-E model assumes that the connection alters between a good (receive) state and a bad (loss) state with certain probabilities
- Time is discretized (e.g. per packet)
- We use the following notation; Probability that a loss state happens at next time instant given that the connection is in a receive state: $p_{l|r}$
- This can be described with a state diagram:



G-E stationary distribution

- Observing the G-E state changes it can be found that the probability that connections are good is p_r and that they are bad is p_l where,

$$p_l = \frac{p_{l|r}}{p_{l|r} + p_{r|l}}$$

$$p_r = \frac{p_{r|l}}{p_{l|r} + p_{r|l}}$$

- Special case: $p_{l|r} = p_l$, $p_{r|l} = p_r$ (independent states)

Average distortion

- Given the G-E packet loss model we can compute the average distortion in the following way:

$$D = p_l D_l + p_r D_r$$

Where D_l is the distortion related to a lost packet and D_r is the distortion related to a received packet.

Humans experiencing distortion/loss

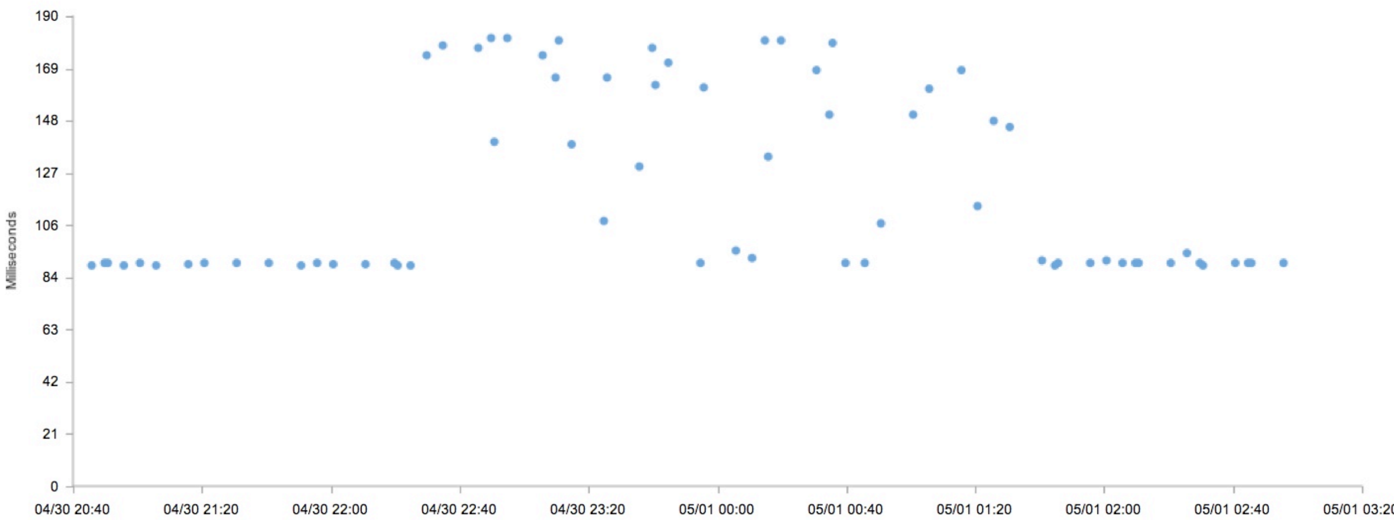
- If some words are lost, a text may be impossible to understand.
- Similarly for a (compressed) photo album image
- For speech and video, some lost data could be accepted by users
- It is difficult to come up with a mathematical distortion measure corresponding to how humans experience speech, audio, and video distortion. Mean square error is commonly used, but does not correspond that well in all applications

Delay

- **Delay** is due to several factors: Processing, Propagation, Transmission, Algorithmic, Congestion
- **Processing delay**: by various software and hardware. Becomes smaller with fast software/hardware
- **Propagation delay**: The time it takes to propagate a signal over the network. Limited by the speed of light, *Propagation time= distance/ (propagation speed)*
- **Transmission delay**: *Transmission time= (nr of bytes to send)/ (transmission speed)*
- **Algorithmic delay**: Algorithm limitations, e.g.:
  - We collect a number of bits into a longer message and pass it down the IP stack. Even with very fast computers, this delay cannot be avoided.
- **Congestion delay**: Packets are temporarily stored in buffers or are retransmitted (e.g. ARQ used in the TCP protocol).

INFORMATION CODING
Linköping University

Delay statistics



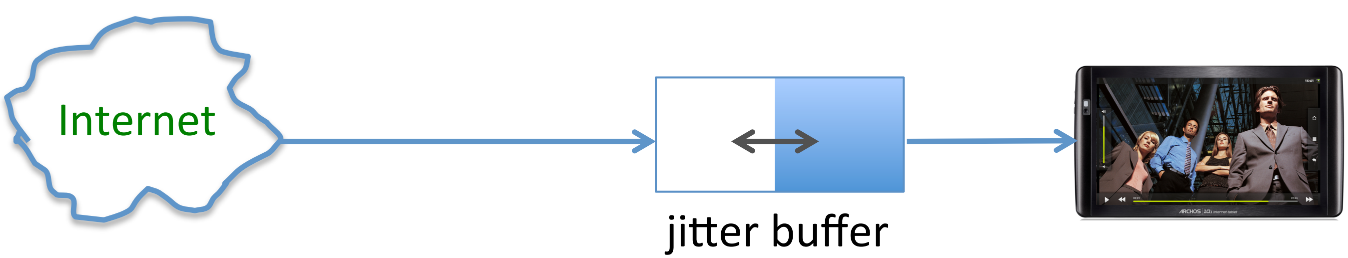
Example scatterplot for ping response time*

*http://blog.catchpoint.com/2011/05/02/network-latency-web-latency-catchpoint-the-plumber/

INFORMATION CODING
Linköping University

Jitter

- Variation in the delay of received packets is called *jitter*. With a jitter buffer and delayed playout, severely delayed packets are given a chance to catch up, and do not have to be considered lost.
- Jitter buffers with too long playout delay threaten real-time applications. There is thus a tradeoff between jitter buffer size and delay!



Humans experiencing delay

- A text chat is not sensitive to one-way delays of 400 ms
- A speech/video real-time conversation cannot tolerate large delays

  < 150 ms one-way is good,

  150-400 ms is acceptable,

  > 400 ms is bad.

  International Telecommunication Union (ITU) considers network delay for voice applications in Recommendation G.114.

Some application requirements

- Real-time speech/video conversation (“*hard real-time*”)
  - Tolerate distortion but not delay > 400 ms
- Streaming video (*real-time*)
  - Tolerate minor distortion, tolerate some delay but not jitter
- Text applications (*non real-time*)
  - Tolerate delay but no distortion

None of these services tolerates that packets are delivered in wrong sequence order

“Transforming” the best-effort channel

- Can we “transform” the best-effort channel to the following types:
  1. A guaranteed low-delay channel (speech/video conversation)?
  2. A guaranteed fixed-delay, jitter-free channel (streaming)
  3. An error-free channel (for text/files transmissions)

  - Including guaranteed sequence order in all cases?
- Yes, to a certain extent, but not completely!

Basic “transforming” tools

1. Guaranteed low-delay channel
   - Use **sequence number** to keep track of sequence order of packets and lost packets
   - Use **small receive buffer** to allow reordering of received packets
   - **Throw away packets** that are not received in time
   - Drawback: Distortion is generated from lost and delayed packets

2. Guaranteed delay, jitter-free channel
   - Use **sequence number** to keep track of sequence order of packets and lost packets
   - Use **large receive buffer** to even out the jitter and allow reordering of packets
   - **Throw away packets** that are not received in time
   - Drawback: Distortion is generated from lost and delayed packets

3. Error-free channel
   - Use **sequence number** to keep track of sequence order of packets and lost packets
   - Use **receive buffer** to allow reordering of packets
   - Use **transmit buffer and ARQ** to retransmit lost packets
   - Drawback: Arbitrarily long delays may occur

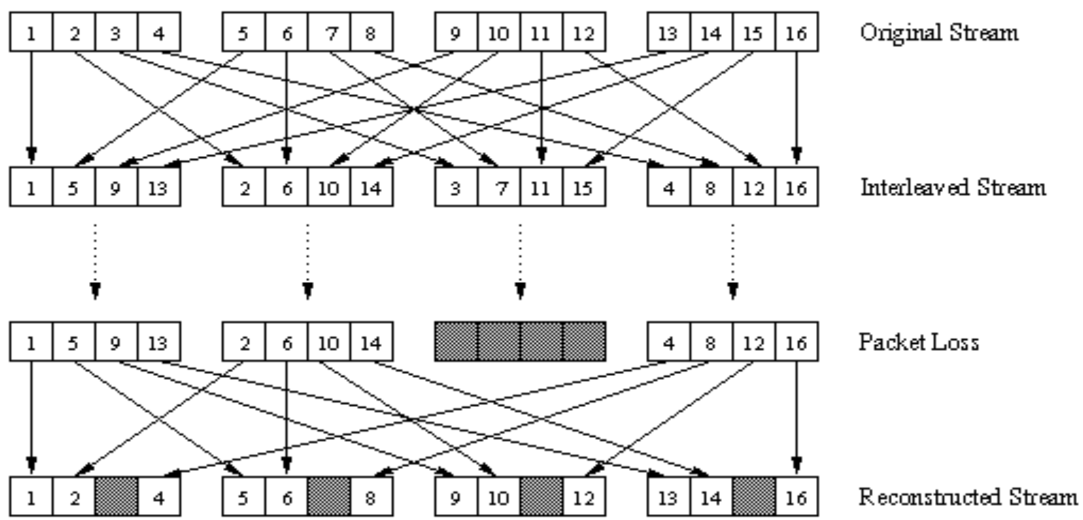
Additional tools

- Time stamp => we know when to play out a packet <u>even with variable length coding if previous packet is lost</u>
- For video with variable length coding, we need extra synchronization info to understand at the receiver where to play out a packet in the image/video frame <u>even if previous packet(s) lost</u>.
- If the distortion is too high, forward error correction (FEC) and interleaving can be added.
- Alternatively use Error Concealment (for audio and video)
- Multiple-description coding / Layered coding for audio/video
- Transmit over multiple paths in the network to increase data rate and decrease risk of lost packets
- Use packet classification – Diffserv, IntServ
- Use distributed file/playout servers
- Use "over-provisioning": Build the network to be able to carry much more traffic than is actually used
- Use "managed network": Set up your own (closed) TCP/IP network to enable full control (resource reservation).

Some application protocols which incorporate the above tools have been standardized.

Interleaving

- Spread out message over several packets:



- A lost packet gives rise to isolated errors in several packet
- These errors can be corrected using FEC or interpolation

Slide adapted from *Computer Networking: A Top Down Approach*
Jim Kurose, Keith Ross, Addison-Wesley

Error concealment

- An alternative to FEC and ARQ
- Use dependencies (correlation) in the data to interpolate or predict lost data at the receiver side
- Audio errors can be concealed based on a AR signal model
- Video errors are concealed through interpolation in - and between images.

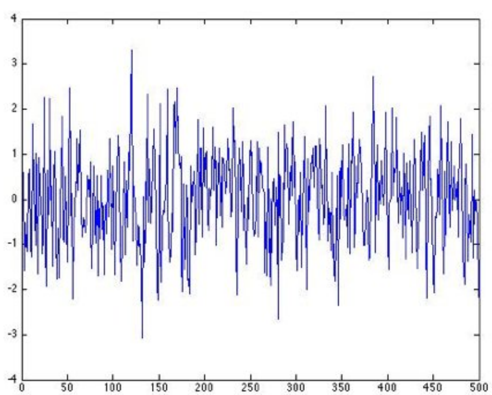
INFORMATION CODING
Linköping University

The AR model of a signal

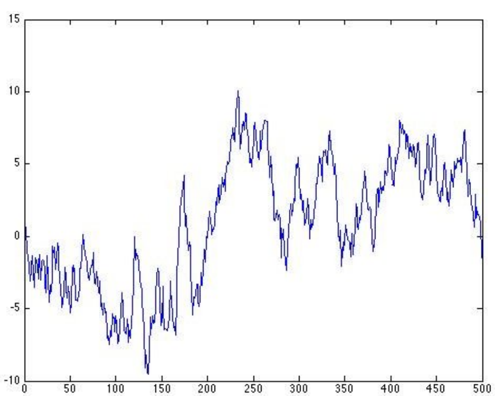
- A very simple mathematical model of natural signals (like audio) is the 1:st order auto-regressive (AR) model:
  - A sample has an amplitude that is a linear combination of the previous sample amplitude and a random value:

    $$x(n) = a\,x(n-1) + u(n)$$

  - The random value u is drawn from a Normal (Gaussian) distribution with mean value 0 and variance 1.



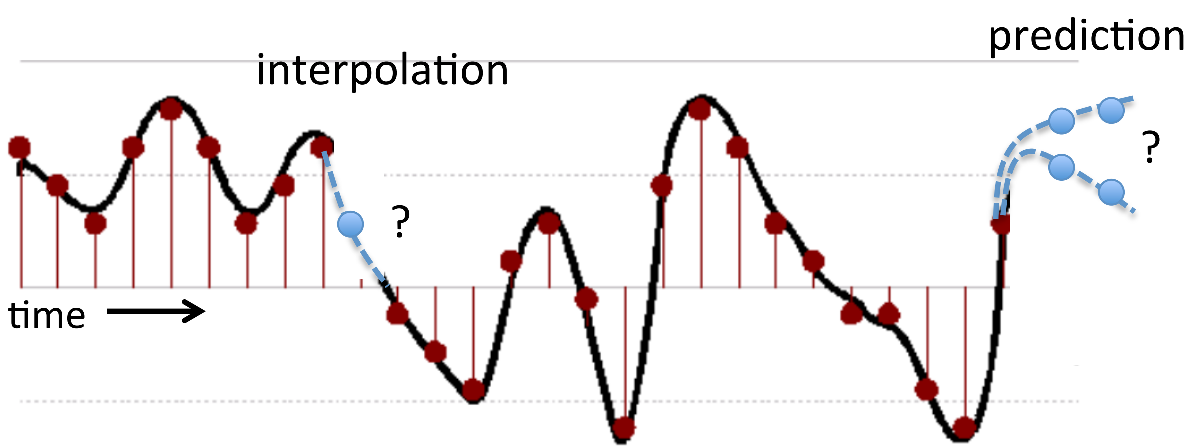
$a = 0.2$



$a = 0.98$

Interpolation and prediction

- **Interpolation**: estimate one, or several missing samples in-between other samples
- **Prediction**: estimate one, or several future samples



- What is the best estimate?

AR model – interpolation and prediction

- The average signal power: $\sigma^2 = \dfrac{1}{1-a^2}$
- The best interpolation between two neighboring samples:

$$\tilde{x}(n) = \frac{a}{1+a^2}\left[x(n-1) + x(n+1)\right]$$

- the average interpolation error: $MSE_{inter} = 1/(1+a^2)$

- The best prediction of a future sample: $\tilde{x}(n+k) = a^k x(n)$

- the average prediction error: $MSE_{pred} = \dfrac{1-a^{2k}}{1-a^2}$

Note: predicting long into the future gives the predicted value 0 and an error that is the same as the power (σ^2) of the signal. Similarly for a lost part of the signal e.g. due to packet loss.

Video error concealment example*

Packet loss: 1%, 5%, 10%

*http://www.youtube.com/watch?v=-FLXEOxPUSk

Multiple description coding/Layered coding

- Multiple description coding: Send several packets that can be decoded separately if one packet is lost, or together to get better quality, if all packets are received (error control).
- Layered coding: Users with poor channels only get the basic layer/packet, but some users with better networks receive extra layers/packets improving the service quality (congestion control).

Multiple description coding - Examples

- Example 1: Sending even samples in one packet and odd samples in a second packet.
  - If both packets make it => full quality
  - If only one packet makes it => reduced quality (interpolation)
  - If neither packet makes it => Distortion = σ^2
- Example 2: Use two quantizers with M/2 levels, shifted half of the quantization interval.
  - If both packets make it => full quality (corresponding to M levels)
  - If only one packet makes it => reduced quality (by 6 dB, M/2 levels)
  - If neither packet makes it => Distortion = σ^2

Layered coding - Examples

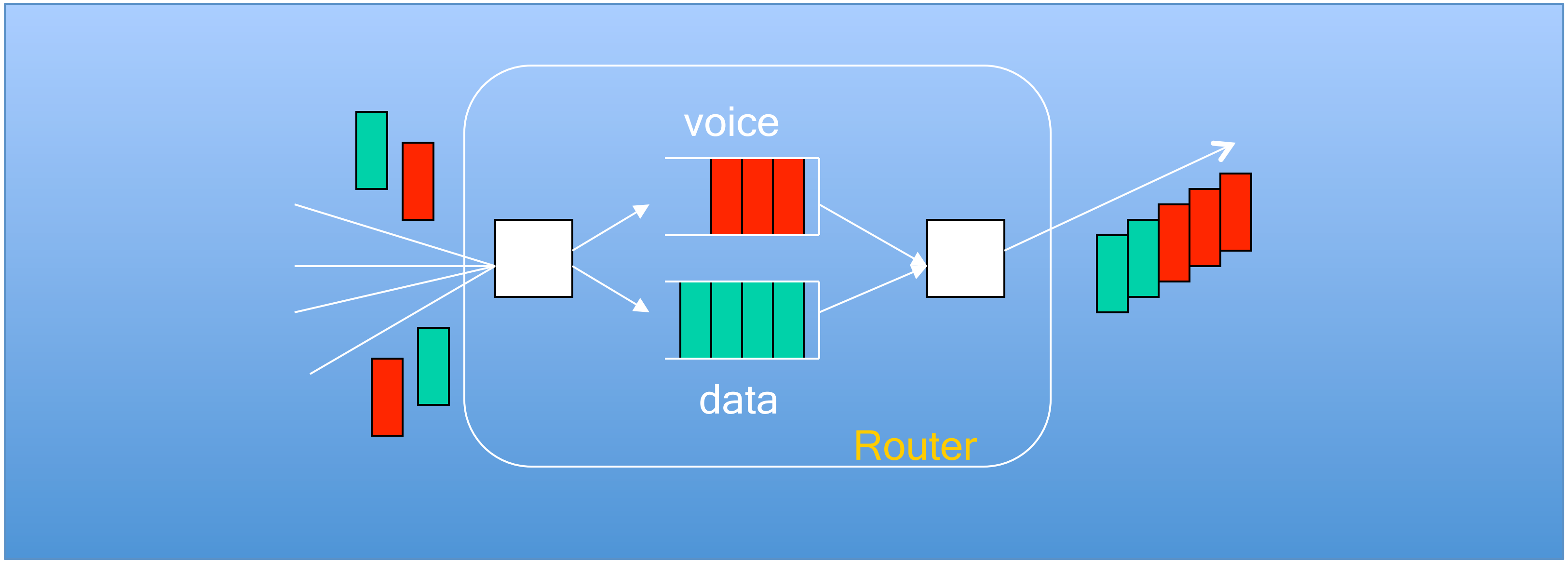
- Example 1: Quantize samples coarsely (e.g 16 levels). Put these samples in one packet (4 bit/sample). Within each level, continue quantize with 16 finer levels. Put in second packet.
  - Users who get both packets => full quality, $D \approx 1/12\ (A/256)^2$
  - Users who only get first packet => $D \approx 1/12\ (A/16)^2$

- Example 2: Split the signal in low frequencies and high frequencies. Send in different packets
  - Users who get both packets => full quality
  - Users who only get first packet => reduced quality (low-pass signal)

INFORMATION CODING
Linköping University

DiffServ – Differentiated services

Packets are labelled to belong to a traffic class and enter different queues in the routers. High priority packets are transmitted before low priority packets.



DiffServ

- RFC 2474
- Main traffic classes are
  - *Default* PHB (Per hop behavior)—which is typically best-effort traffic
  - *Expedited Forwarding* (EF) PHB—dedicated to low-loss, low-latency traffic
  - *Assured Forwarding* (AF) PHB—gives assurance of delivery under prescribed conditions
- Has not been successful due to problems to handle traffic classes across different operator networks. Also it has turned out that the discipline of the users is not high enough for the method to work (who would like to label their packets as “less important”?)

IntServ - Virtual circuits

- IntServ – "Integrated services"
- Before sending the data packets, a "virtual connection" is established. For each connection, we accept only a certain number of packets per second ("resource allocation").
- When the router cannot handle more connections, it announces this as soon as a host tries to establish a new connection.
- In theory one can guarantee throughput, no packet loss and low delay, i.e. quality of service (QoS). However, this requires full control of all traffic that passes all the routers along the path ("Managed network"). In practice this is difficult to implement on a large scale due to the many involved ISPs.

Protocols supporting multimedia

- TCP has sequence numbers and time stamps. We know that TCP ARQ gives delay, and is therefore not suitable for audio/video conversations.
- UDP does not use retransmissions, and is therefore suitable for video conversations. It does not have time stamps and sequence numbers. How can we complement it?
- One solution is the Real-time Transport Protocol (RTP). Each RTP packet is encapsulated in a UDP packet. The RTP header includes time stamp and packet sequence numbers.
- Real-Time Transport Control Protocol (RTCP): statistics for RTP sessions, answering, e.g., how many packets were lost. It is up to the application developer what to do with this info.
- H.323 (ITU) / Session initiation protocol (SIP) (IETF): Standards for computer to computer/PSTN communicaton. Both work with video and speech
- H.323: Umbrella standard for both control and data transfer, uses RTP
- SIP: Only control, works with RTP.

End of lecture

Next lecture: More on streaming media