

Last weeks of the course

- Lecture, Thursday Oct 9 – Multimedia (cont)
 - Student presentation
- Last lecture, Tuesday Oct 14
 - Test on fairness + multimedia
 - Time for project work
 - Consultation – if you have questions before the make-up test
- Home assignment
 - by Oct 14
- Project
 - Outline, max one page, paper structure, content of sections: Oct 6
 - Final report, max 5 pages: Oct 13
 - Presentation day: Oct 15
- Make-up test
 - Preferences? (Oct 16, 20, 21)?

Multimedia communication

Delay and loss control

- Lecture material:
 - E. de Souza e Silva, et al., "Performance Issues of Multimedia Applications," Performance 2002, tutorial
- Reading for next lecture:
 - N. Laoutaris, I. Stavrakakis, "Adaptive playout strategies for packet video receivers with finite buffer capacity," IEEE ICC 2001.
 - X. Yu, J.W. Modestino, X. Tian, "The accuracy of Gilbert models in predicting packet-loss statistics for a single multiplexer network model," IEEE Infocom, 2005

Multimedia communication

- Multimedia communication classes and requirements
- Delay control
 - Playout buffer / Student presentation
- Loss control
 - Forward Error Correction (FEC)

Multimedia transmission

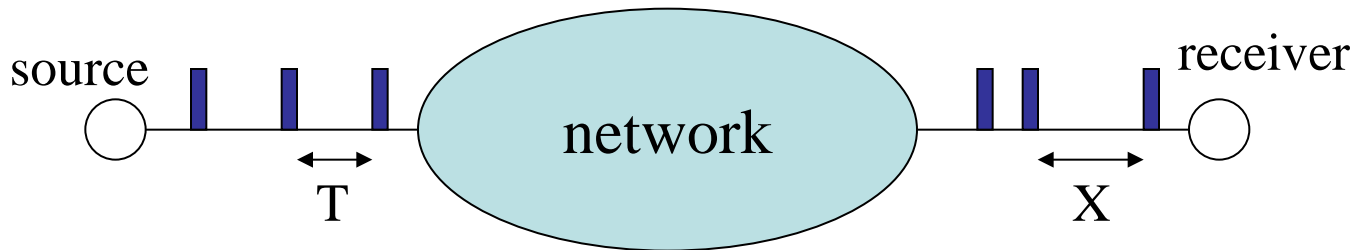
- Different cases of media distribution according to
 - availability of content
 - delay requirements
- **File download**
 - content is available at servers or at users for P2P applications
 - download then playback
 - no delay limitations
- **Video on Demand** – offline streaming
 - content is available at the servers or at the users
 - playback while downloading (offline streaming)
 - some delay limitation
- **Live streaming** – online streaming
 - content is generated during playback
 - playback delay have to be minimized

Streaming

- Delay-jitter control
 - to compensate delay variations (jitter) due to varying congestion at the network nodes
 - method: playout buffer at the receiver, playout delay control
- Loss control
 - to deal with packet losses, when retransmission is too slow
 - method: increase redundancy at the source, forward error correction
- All control methods have to consider the end-to-end delay limits
- In the case of live streaming the traffic characteristics is not known a-priori, this makes jitter and loss control challenging
- In general it is hard to find appropriate end-to-end delay and loss models.

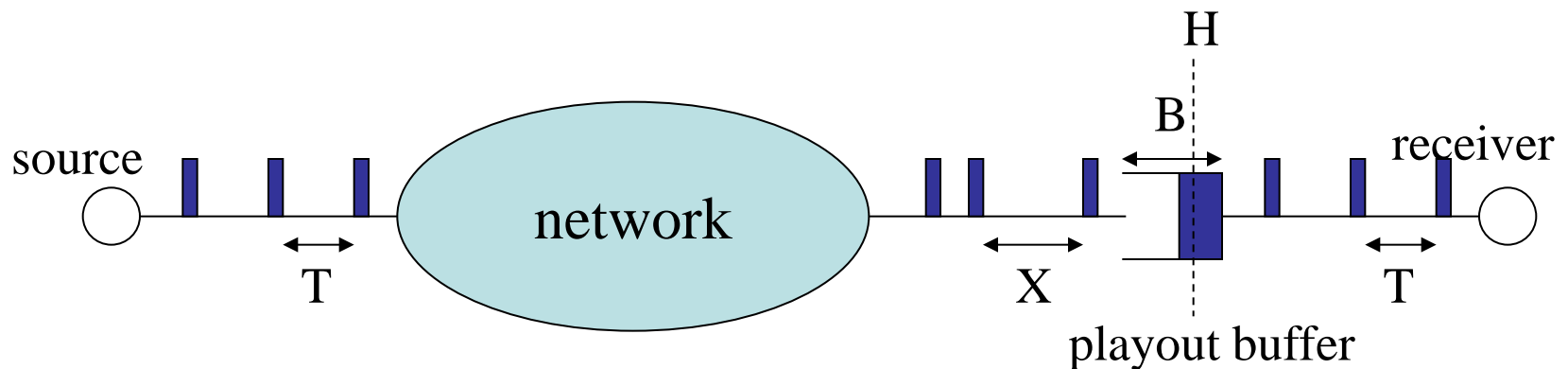
Delay-jitter control

- What is jitter:
 - T packet generation interval (constant or random v.)
 - X interarrival time at the receiver (random v.)
 - Jitter: $J = X - T$ (random variable)

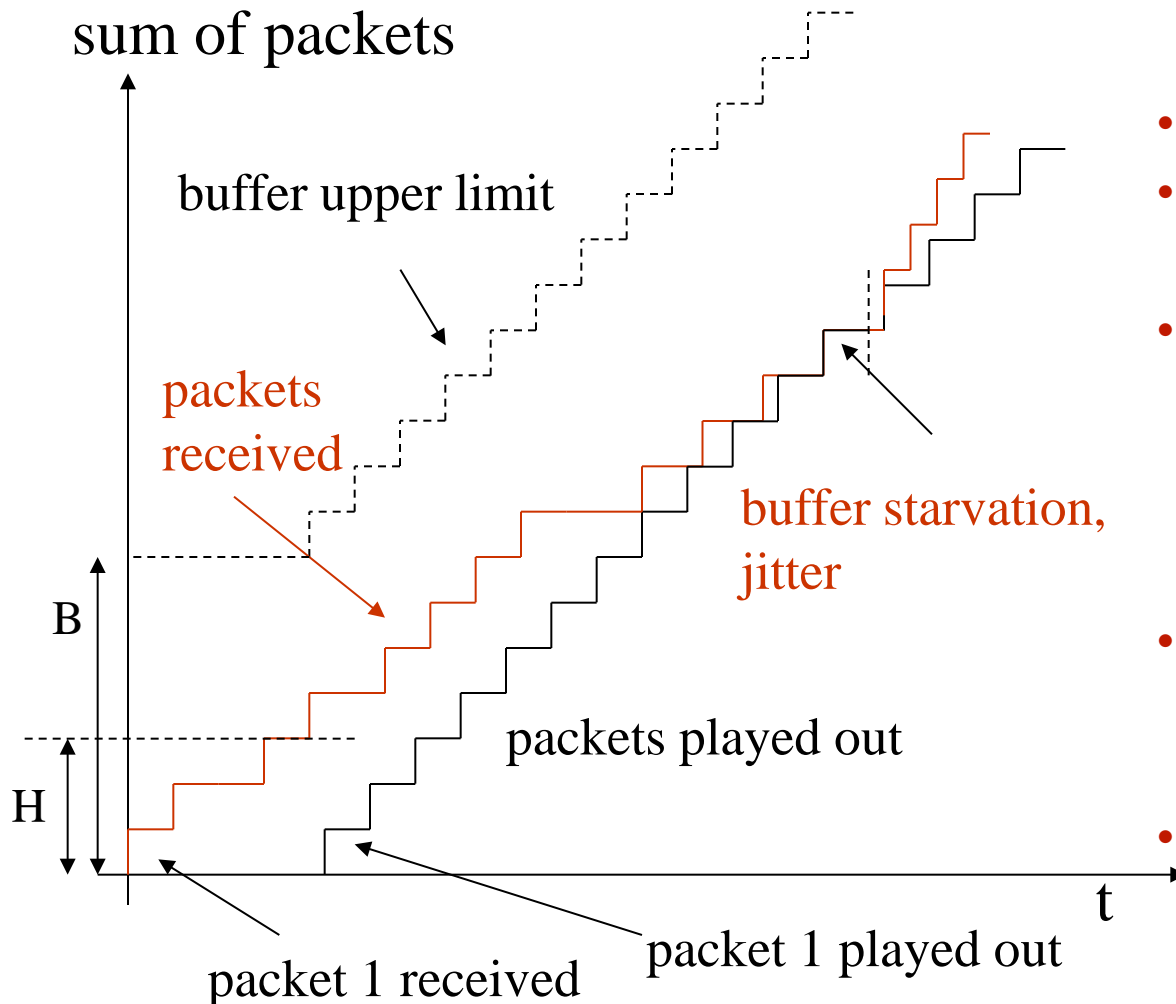


Playout buffer

- Playout buffer to compensate for jitter
 - at the receiver side
 - to store received, but not decoded packets
 - decoding rate determined by the coding scheme
 - buffer size B
 - threshold H , when decoding starts

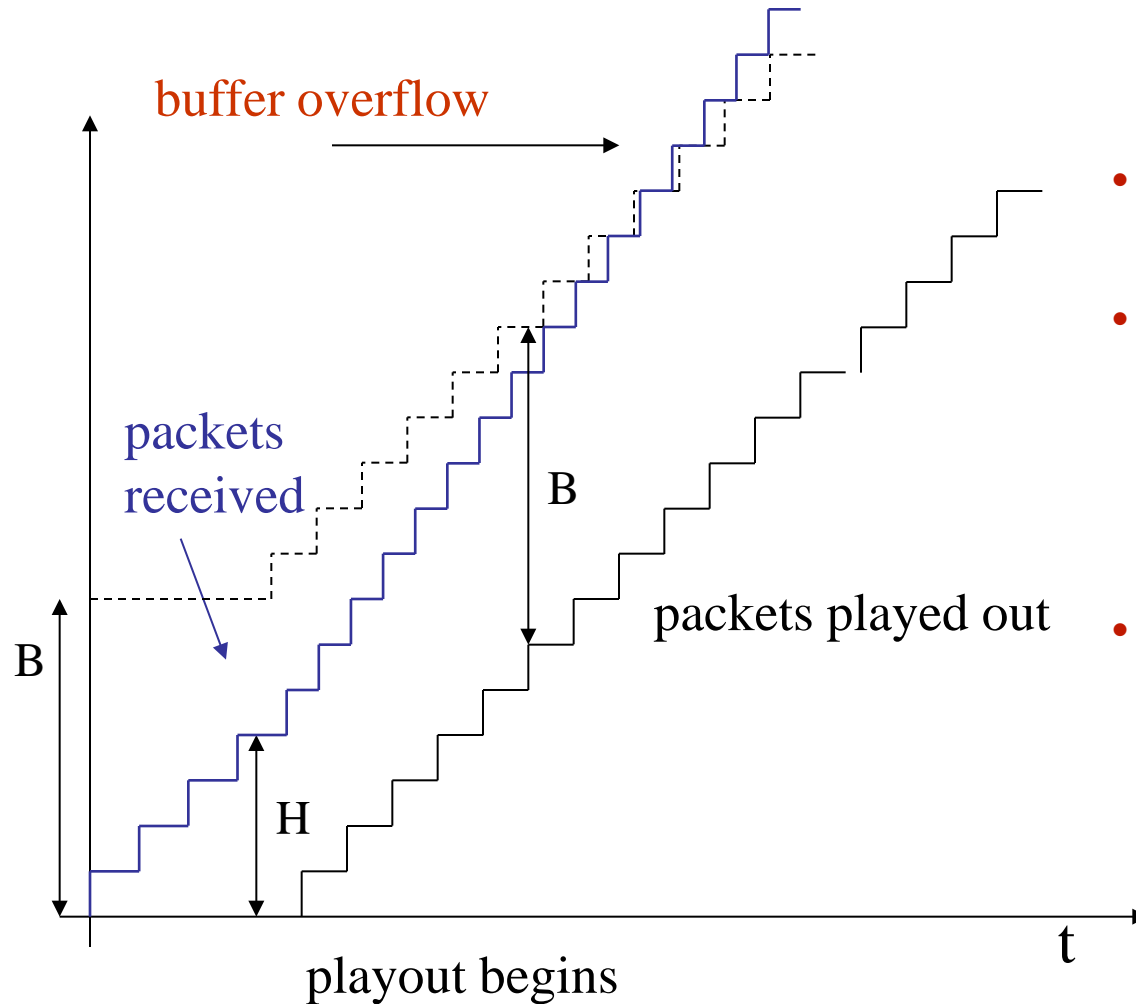


Playout buffer - starvation



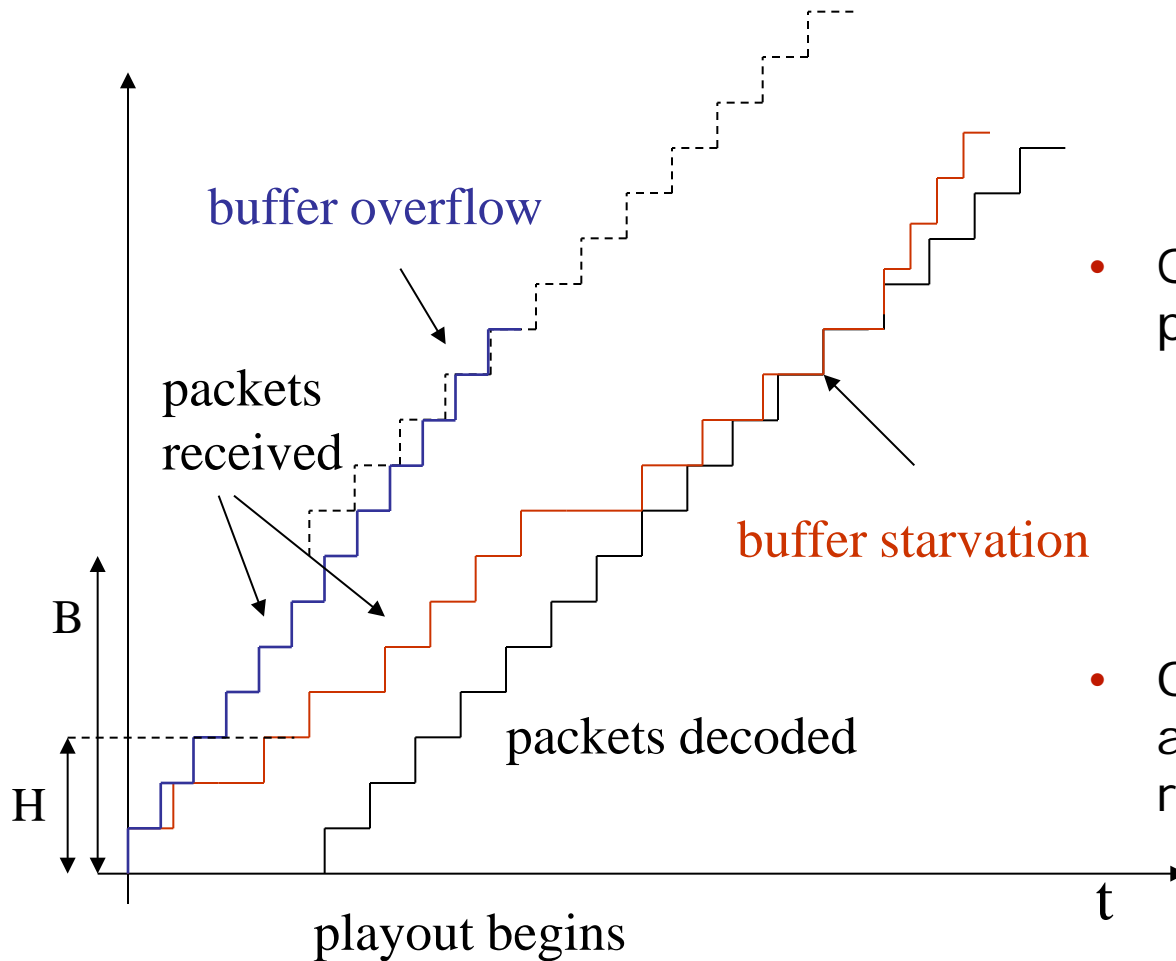
- B: playout buffer size
- H: playout threshold
- Buffer starvation: the buffer is empty, there is no available content to play out
- Jitter, increased delay between two samples
- H should be increased

Playout buffer



- For larger H :
- Buffer overflow: buffer can not store more packets, packet loss
- Large H also increases the end-to-end delay (network delay of the first packet + H)

Playout buffer



- Question: select H to provide acceptable:
 - end-to-end delay,
 - probability of starvation
 - probability of packet loss due to buffer overflow
- Other control solutions to achieve smooth reception?

Loss control for multimedia

- Loss control for elastic flows is often based on retransmission
 - This may not work for streaming applications due to the playout delay limit
 - At the same time some loss is tolerated by the application
- Solutions with lower delay – based on information redundancy
 - Application layer, utilizing the characteristics of the media content
 - Network layer – multimedia specific or not...

Loss control on application layer

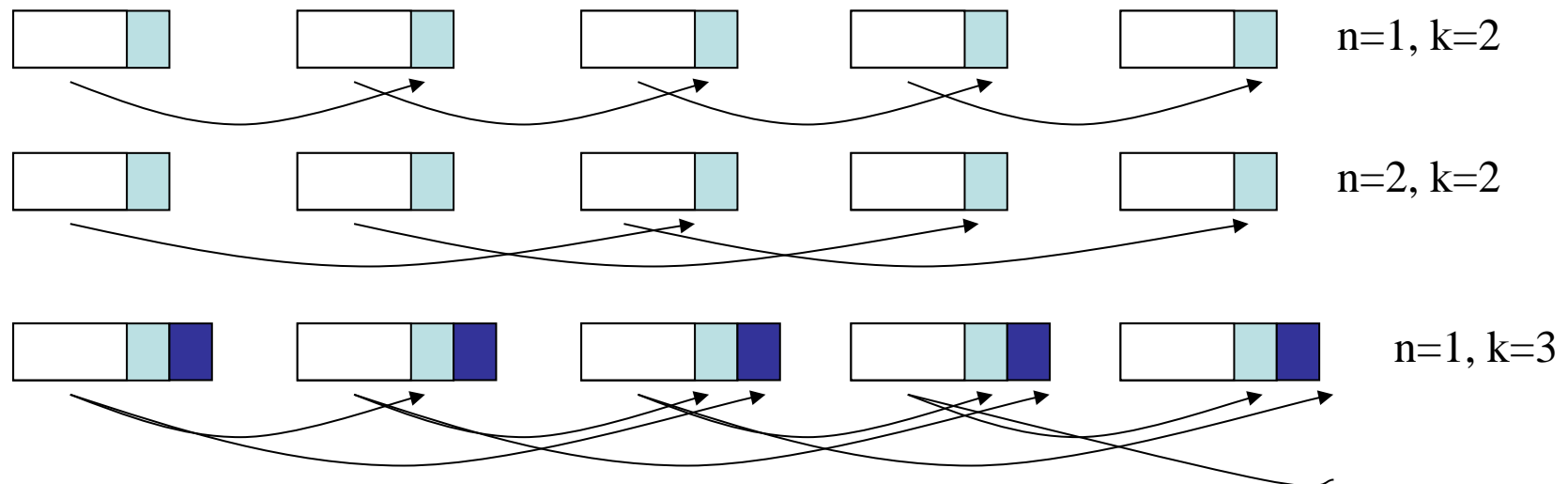
- Error concealment at the receiver
 - reconstruct the signal from available information
 - reuse last sample
 - interpolate from neighboring samples
- Error resilience with source coding
 - error propagation due to source coding (e.g., interframe coding in video coders)
 - limit error propagation (e.g., I frames in MPEG)
 - increases bandwidth
- Layered source coding
 - divide information into important and less important parts
 - transmit important information with high priority (requires network support!) or with higher redundancy
- Multiple description coding (this is source coding as well)
 - code the information in two redundant streams
 - receiving both streams gives good quality but receiving one stream only gives acceptable quality as well

Loss control for multimedia

- Loss control in the network layer (IP)
 - Interleaving of packets
 - loss can happen in bursts (buffer overflow)
 - single packet losses are easier to compensate for
 - scrambles the packets at the source
 - introduces latency, but does not consume extra bandwidth
 - Forward Error Correction (FEC)
 - adds redundancy to the stream of packets
 - uses this redundancy to reconstruct lost packets
 - introduces latency and uses extra bandwidth

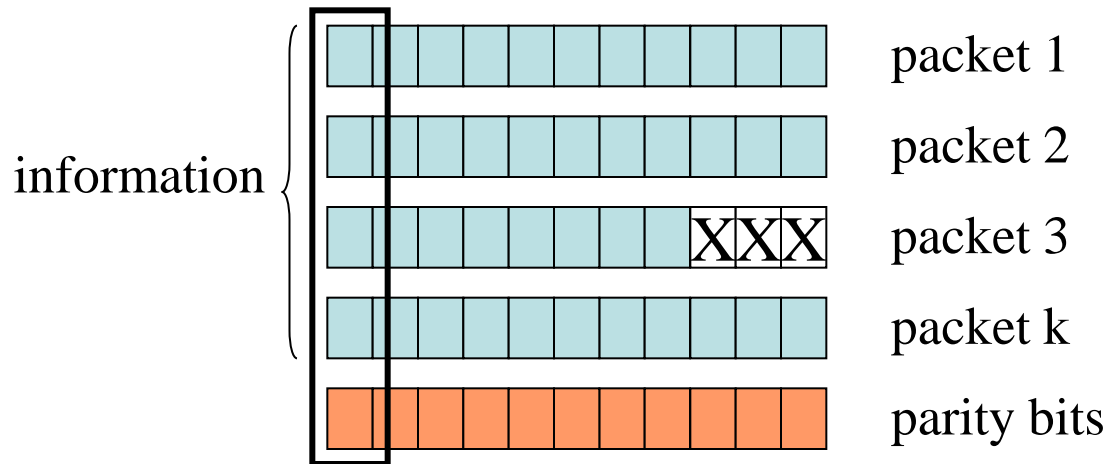
Forward Error Correction

- Media dependent FEC – parameters: n, k
 - add redundant copy to a consecutive packet
 - with low coding rate – media dependence
 - for low bitrate sources, with delay limitations
 - implemented in VoIP tools, IETF recommendation
 - for increased performance
 - increase the delay between the original and the copy ($n=1, 2, \dots$)
 - add multiple copies ($k=2, 3, 4, \dots$)



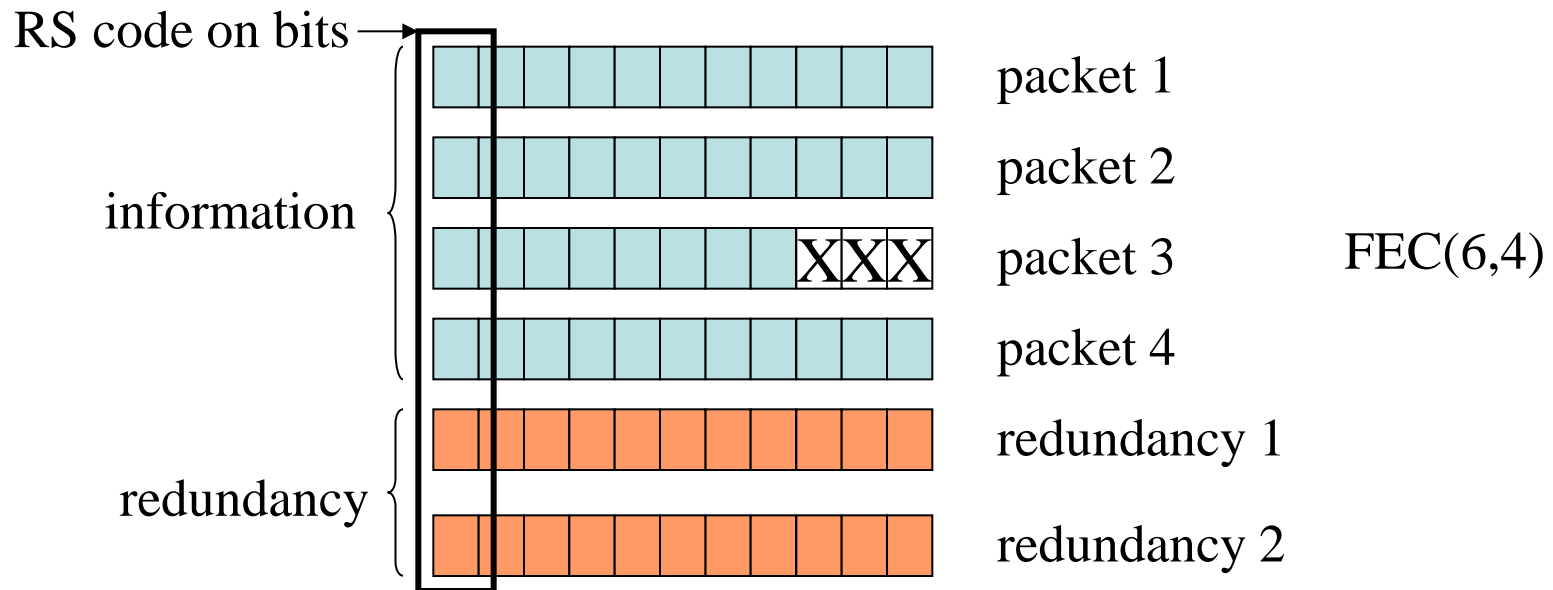
Forward Error Correction

- **Media independent FEC** (block codes)
 - apply error correction codes on a block of packets
 - e.g., parity bit – can regenerate one lost packet



Forward Error Correction

- Media independent FEC (block codes)
 - e.g., Reed-Solomon codes
 - FEC(n,k): k information packets, n-k redundant packets
 - all packets reconstructed if at least k received, otherwise no reconstruction
 - redundancy rate: $(n-k)/n$

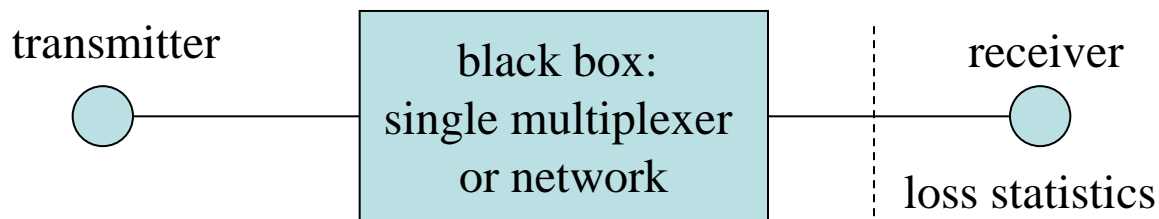


Forward Error Correction

- For both schemes:
 - performance depends both on the average packet loss probability and on the distribution of packet losses
 - consecutive packet losses (media dependent case)
 - number of lost packets in a block of n packets $P(j,n)$ (media independent case)
 - experience: packet losses in the Internet are correlated
- Modeling the loss process at the receiver
 - detailed queuing model
 - Bernoulli model – each packet gets lost with the same probability
 - Gilbert models (reading)
 - Requirement: calculate information loss probability for simple cases

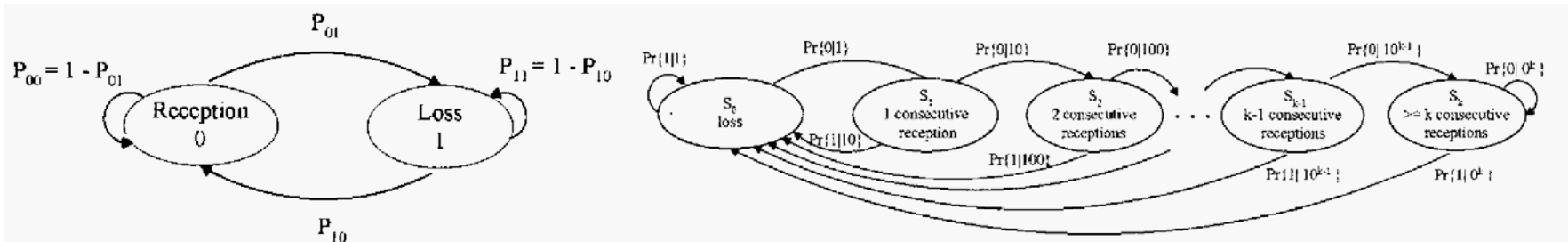
Modeling the loss process

- Queuing models
 - modeling the reasons of packet losses
 - e.g., congestion: queuing networks with finite queues, appropriate arrival and service processes
 - or at least one queue, at the bottleneck link of the connection
 - needs packet arrival process and packet length distribution
 - can be exact, but very complex
- Black-box models:
 - do not model the reason, just the outcome (discrete time process)
 - simple, but not that accurate
 - Bernoulli and Gilbert models



Modeling the loss process

- Bernoulli model
 - All packets get lost with the same probability, losses are independent
- Gilbert model – basic version
 - To model the burstiness of the loss process
 - Two state discrete time Markov chain
 - Parameters to calculate:
 - Steady state packet loss probability
 - Average loss burst length
 - When does the Gilbert model reduce to Bernoulli model?
- Extended Gilbert model (reception run lengths)



Modeling the loss process

The Bernoulli example – Group work

- Bernoulli model, loss probability p
- Media dependent FEC(n, k)
 - n : redundant copy in the n^{th} packet
 - k : altogether k copies (1 original, $k-1$ copies)Calculate $P(\text{information is lost})$ for
 - $n=1, k=2$
 - $n, k=2$
 - $n=1, k$
- Media independent FEC(n, k) – block code with k information packets
 - $P(j, n) = P(j \text{ packet lost out of } n)$
 - $P(\text{losses despite redundancy})$
 - $N = E[\text{lost packets in block after reconstruction}]$
 - $I = E[\text{lost information packets in a block after reconstruction}]$
 - $P(\text{information is lost})$

The accuracy of loss models

- Reading assignment: The accuracy of Gilbert models ...
- Packet loss process - Gilbert models for a single multiplexer
 - Single source
 - Multiple sources
- Performance of Forward Error Correction (media independent)
 - RS(n,k): k information packets, and additional n-k parity packets
 - Steps of modeling
 - $P(j,n)$ – Probability of losing j out of n packets
 - Number of lost packets given j
 - Average number of lost packets in a block
 - Average number of lost information packets in a block
 - Probability of losing an information packet

Summary

- Multimedia transmission
 - File download, off-line and on-line streaming
 - Delay and loss control
- Delay control: Playout buffer control
 - Playout buffer modeling with discrete time MC
- Loss control: FEC
 - Media-dependent and media independent
 - FEC modeling with Gilbert models