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
- Loss control
  - Forward Error Correction (FEC)

- Student presentation
  - FEC performance under bursty loss

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

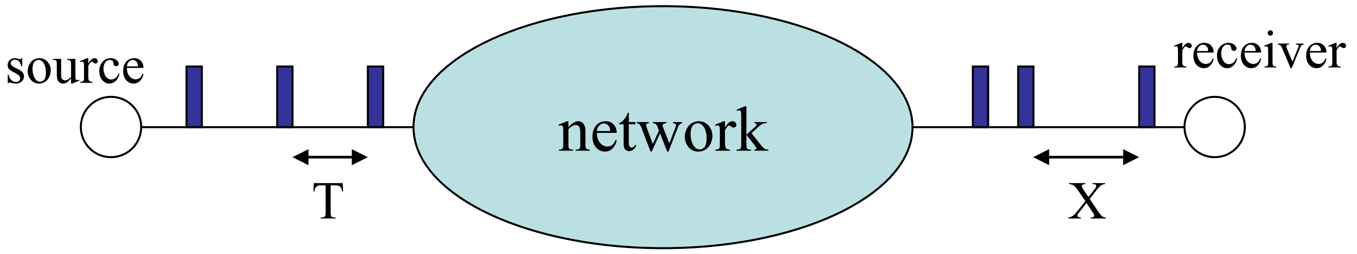
- Delay-jitter control
  - to compensate delay variations (jitter) due to varying congestion at the network nodes
  - method: playout buffer at the receiver, playout delay control
- Loss control
  - to deal with packet losses, when retransmission is too slow
  - method: increase redundancy at the source, 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.

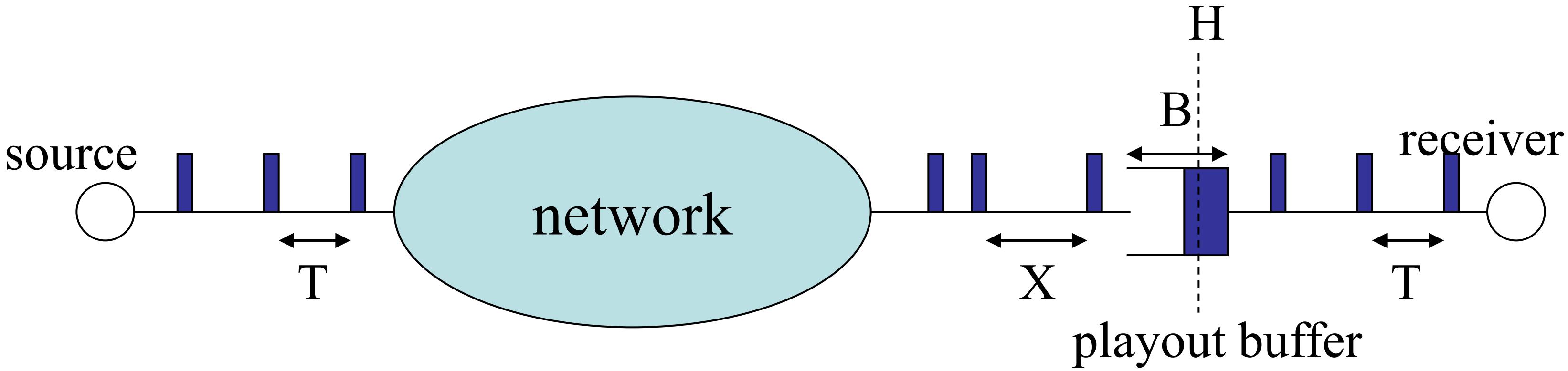
Delay-jitter control

- 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)

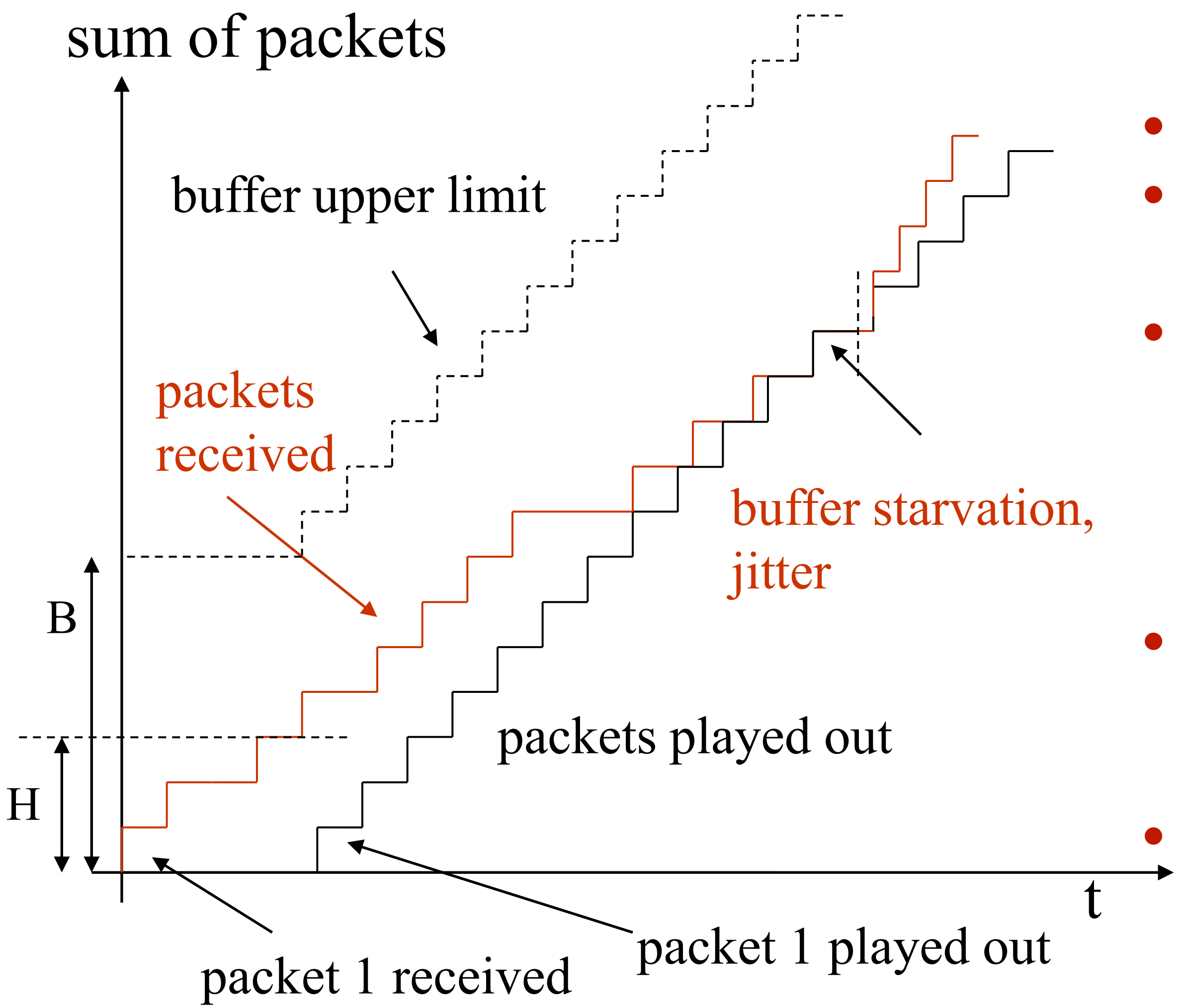


Playout buffer

- 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

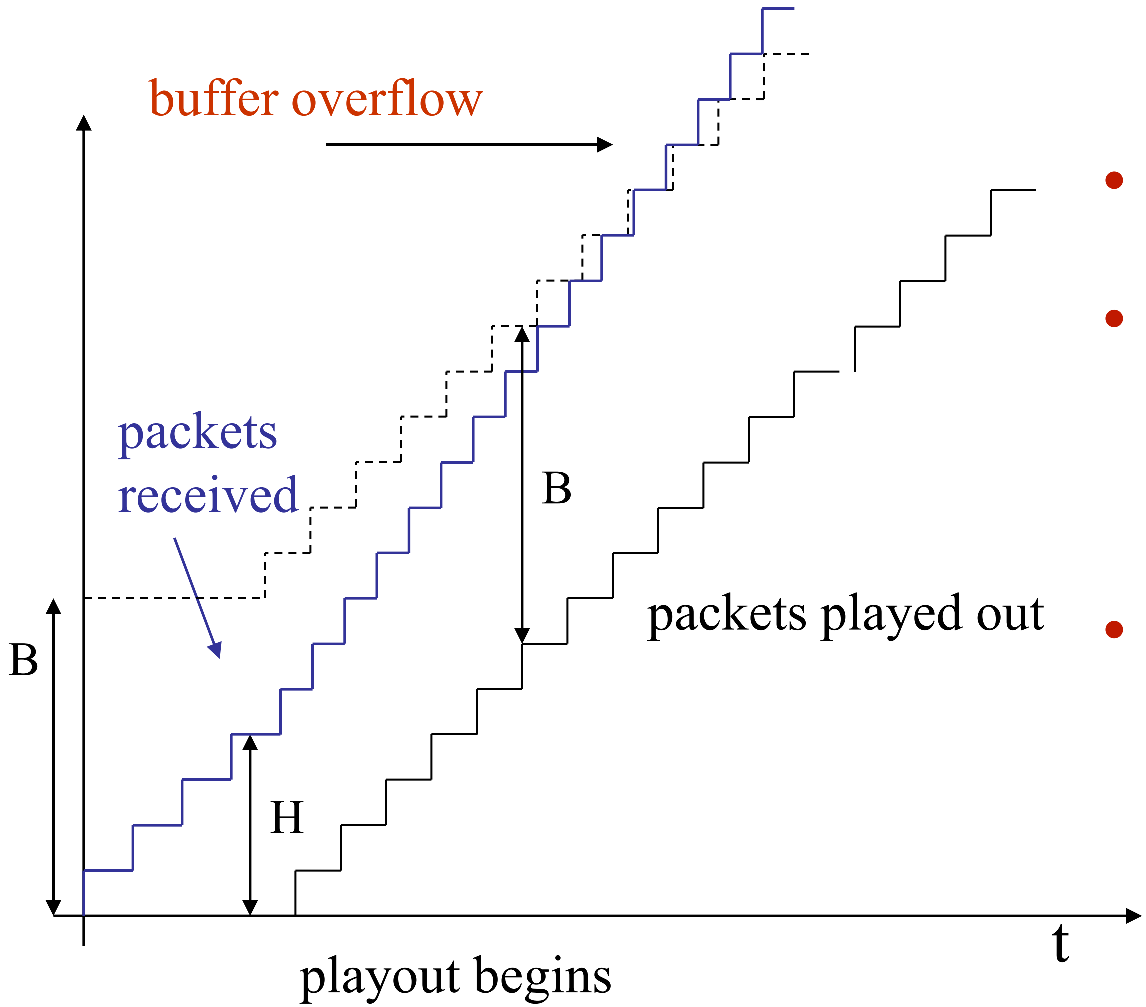


Playout buffer - starvation



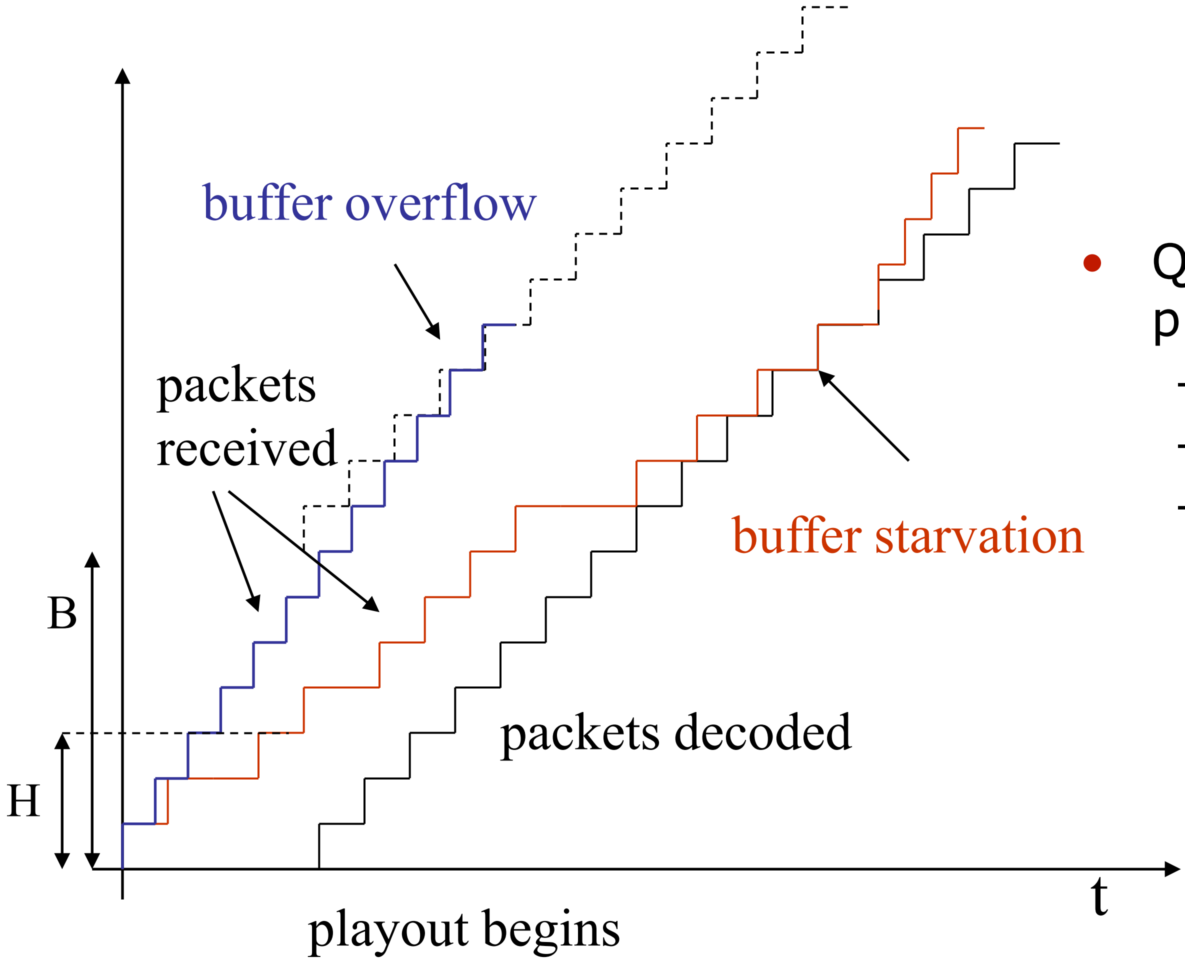
- B: playout buffer size
- H: playout threshold
- Buffer starvation: the buffer is empty, there is no available content to play out
- Jitter, increased delay between two samples
- H should be increased

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

Adaptive playout control ideas

- H should depend on the jitter level of the incoming stream
  - Zero jitter means H=0!
- For streams with silent periods (e.g., voice)
  - Reset H after each silent period
  - Based on previous experiences
  - The length of silent periods might change, but it is not that disturbing.
- For all flows:
  - reset H when starvation happens
  - starvation periods get longer, but their probability decreases
- For all flows:
  - decrease playout rate if queue length is below H
  - e.g., video: keep the frames on display for a longer time (frame freezes)

Playout control: an example

- N. Laoutaris, I. Stavrakakis, “Adaptive playout strategies for packet video receivers with finite buffer capacity,” IEEE ICC 2001.

- Here basic ideas, then home reading:
  - ideally: constant playout time
  - increase playout time if queue length is below TH

- Modeling:
  - system state: number of frames in buffer (i)
  - frame arrival: Poisson(λ) (note, this is the end to end delay jitter modeling)
  - service process (playout): deterministic, state dependent (B(i), μ(i))
  - Markov chain:
    - model the system at frame departure instances
    - discrete time Markov chain
  - Performance measures: considering both delayed playout and packet loss due to buffer overflow

Playout buffer – discrete time MC

$$\mu(i) \triangleq \begin{cases} min(\frac{\mu \cdot i}{\mathrm{TH}}, \mu) & \text{if} \quad 1 \leq i \leq N \\ \frac{\mu}{TH} & \text{if} \quad i = 0 \end{cases}$$

$$B(i) \triangleq \begin{cases} max(\frac{\mathrm{TH}}{\mu \cdot i}, \frac{1}{\mu}) & \text{if} \quad 1 \leq i \leq N \\ \frac{\mathrm{TH}}{\mu} & \text{if} \quad i = 0 \end{cases}$$

$$p_{i,j} = \begin{cases} P\{j, B(0)\} & i = 0, 0 \leq j < N \\ 1 - \sum_{k=0}^{N-1} P\{k, B(0)\} & i = 0, j = N \\ P\{j - i + 1, B(i)\} & 0 < i < N, i \leq j < N \\ 1 - \sum_{k=0}^{N-i} P\{k, B(i)\} & 0 < i \leq N, j = N \\ P\{0, B(i)\} & 0 < i \leq N, j = i - 1 \\ 0 & \text{elsewhere} \end{cases}$$

$P\{m, t\}$: Poisson(m arrivals in interval t)

Loss control for multimedia

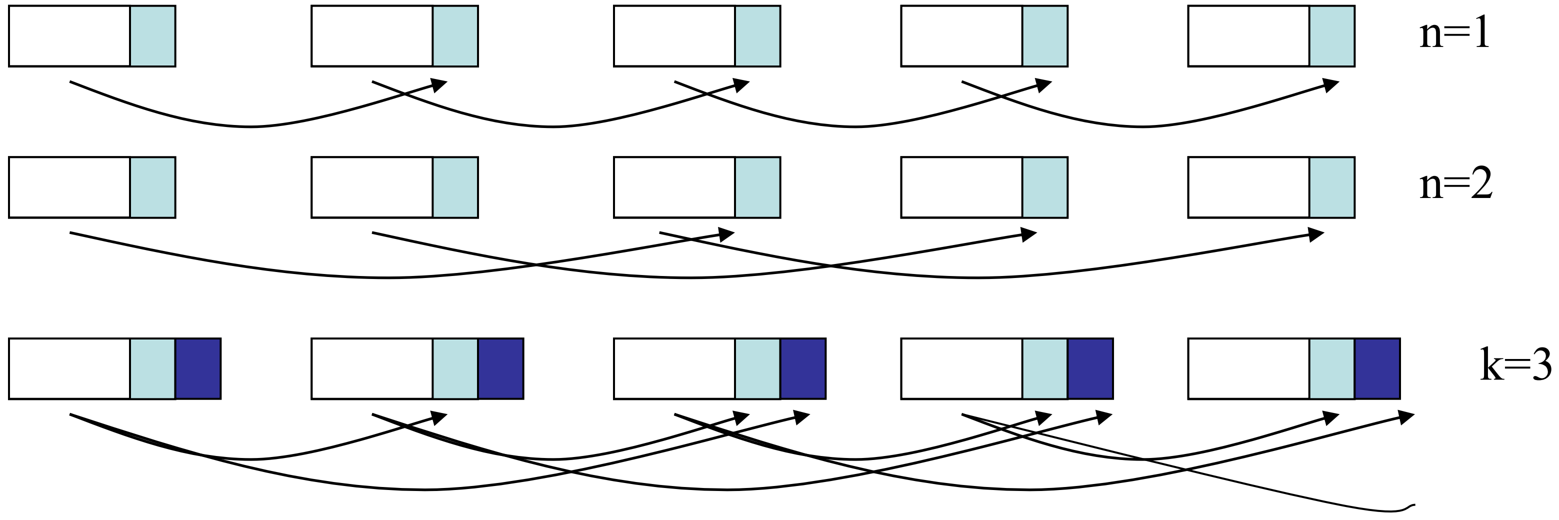
- Error concealment at the receiver
  - reconstruct the signal from available information
  - reuse last sample
  - interpolate from neighboring samples
- Error resilience with source coding
  - error propagation due to source coding (e.g., interframe coding in video coders)
  - limit error propagation (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

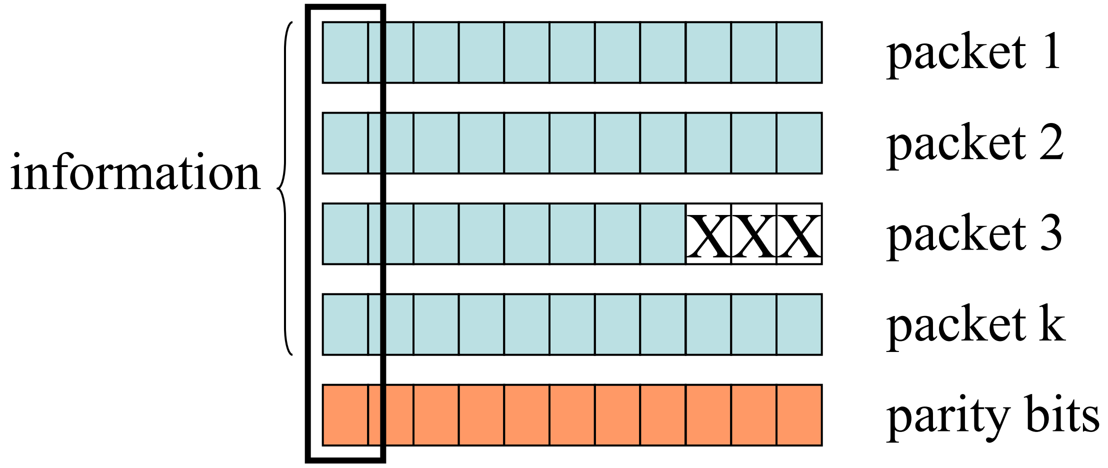
Forward Error Correction

- Media dependent FEC
  - add redundant copy to a consecutive packet (with low coding rate)
  - 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,…)
    - add multiple copies (k=2,3,4…)



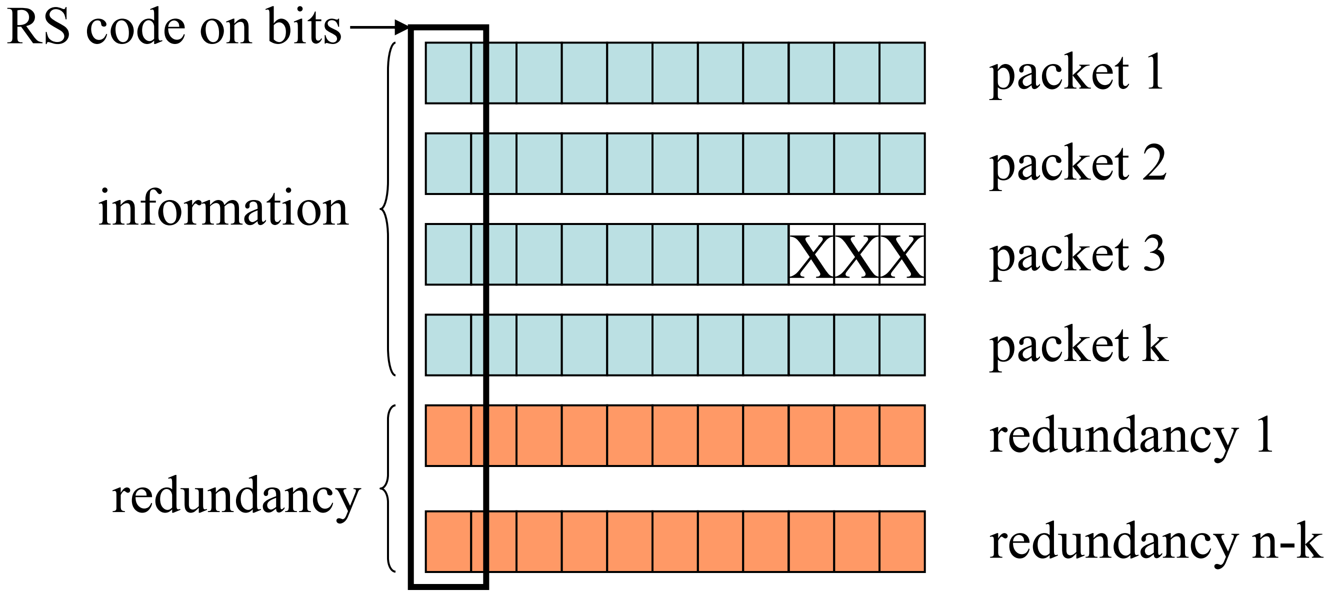
Forward Error Correction

- 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)
  - Reed-Solomon codes
  - RS(n,k): k information packets, n-k redundant packets
  - all packets reconstructed if at least k received
  - redundancy rate: (n-k)/n



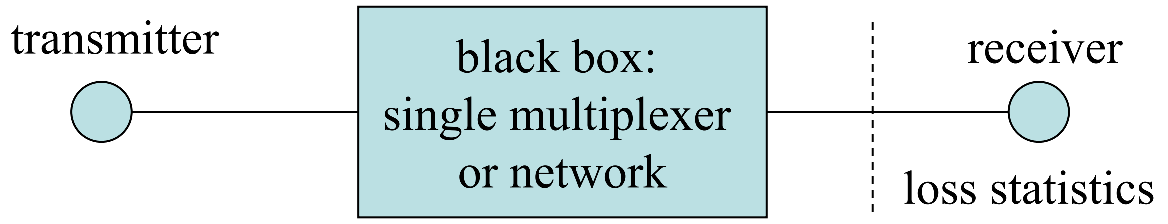
Forward Error Correction

- 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+student presentation)
  - Requirement: calculate information loss probability for simple cases

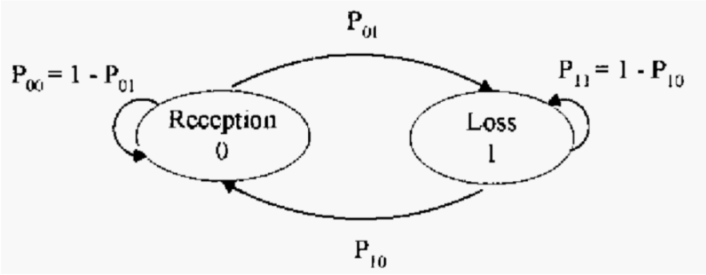
Modeling the loss process

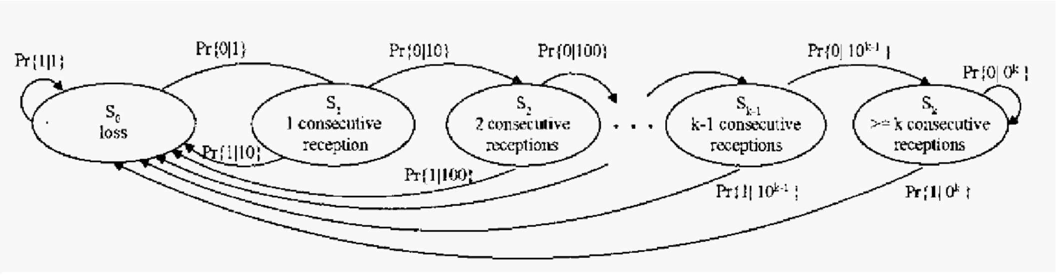
- Queuing models
  - modeling the reasons of packet losses
  - e.g., congestion: queuing networks with finite queues, appropriate arrival and service processes
  - or at least one queue, at the bottleneck link of the connection
  - Needs packet arrival process and packet length distribution
  - can be 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 rate
    - Average loss burst length
    - When does the Gilbert model reduce to Bernoulli model?
- Extended Gilbert model (reception run lengths)





The accuracy of loss models

- Next lecture:

- Packet loss process - Gilbert models for a single multiplexer
  - Single source
  - Multiple sources
- Performance of Forward Error Correction

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