A brief history of streaming media

1992
- MBone
- RTP version 1
- Audiocast of 23rd IETF mtg

1994
- Rolling Stones concert on MBone

1995
- ITU-T Recommendation H.263
- RealAudio launched

1996
- Vivo launches VivoActive
- Microsoft announces NetShow
- RTSP draft submitted to IETF

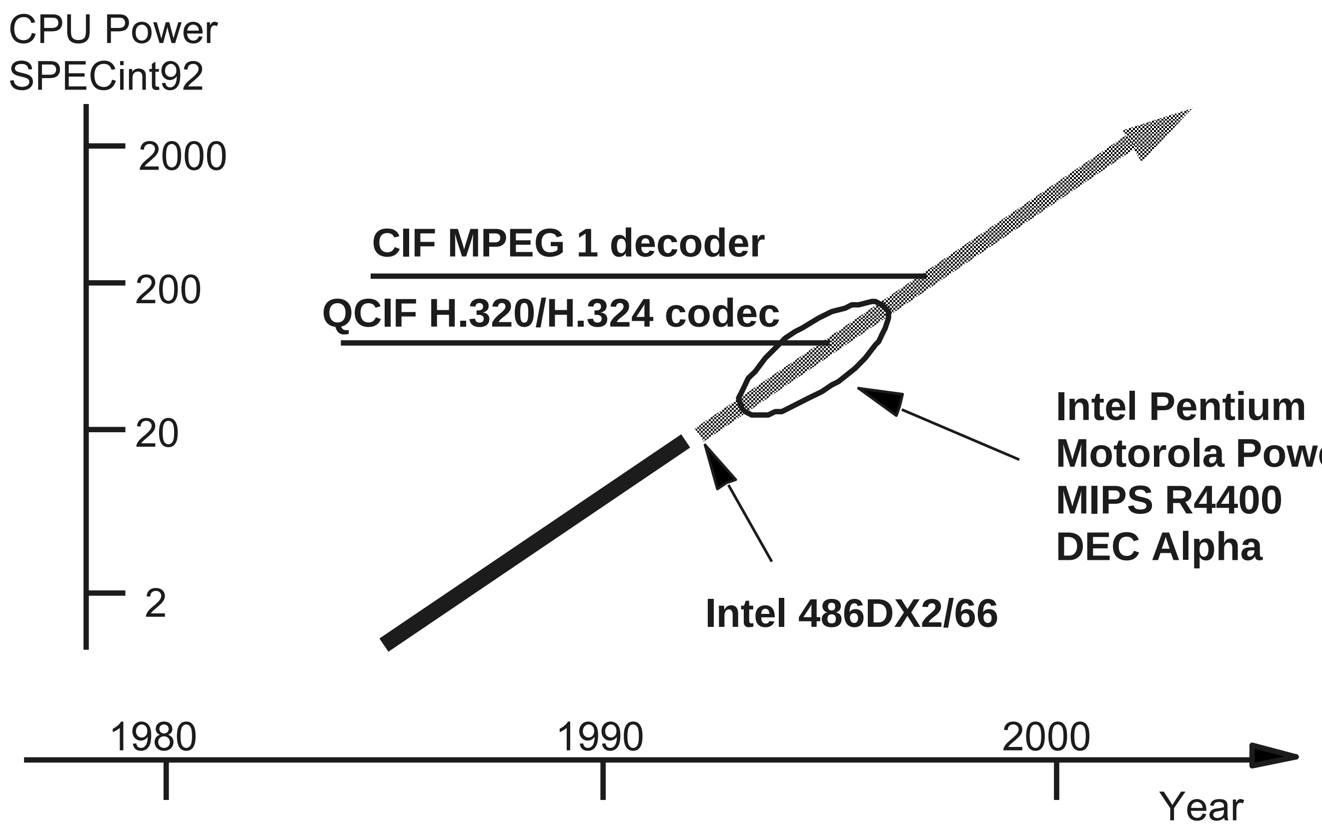
1997
- RealVideo launched
- Microsoft buys VXtreme
- Netshow 2.0 released
- RealSystem 5.0 released
- RealNetworks IPO

1998
- RealNetworks buys Vivo
- Apple announces QuickTime Streaming
- RealSystem G2 introduced

1999
- RealNetworks buys Xing
- Yahoo buys Broadcast.com for $ 5.7B
- Netshow becomes WindowsMedia

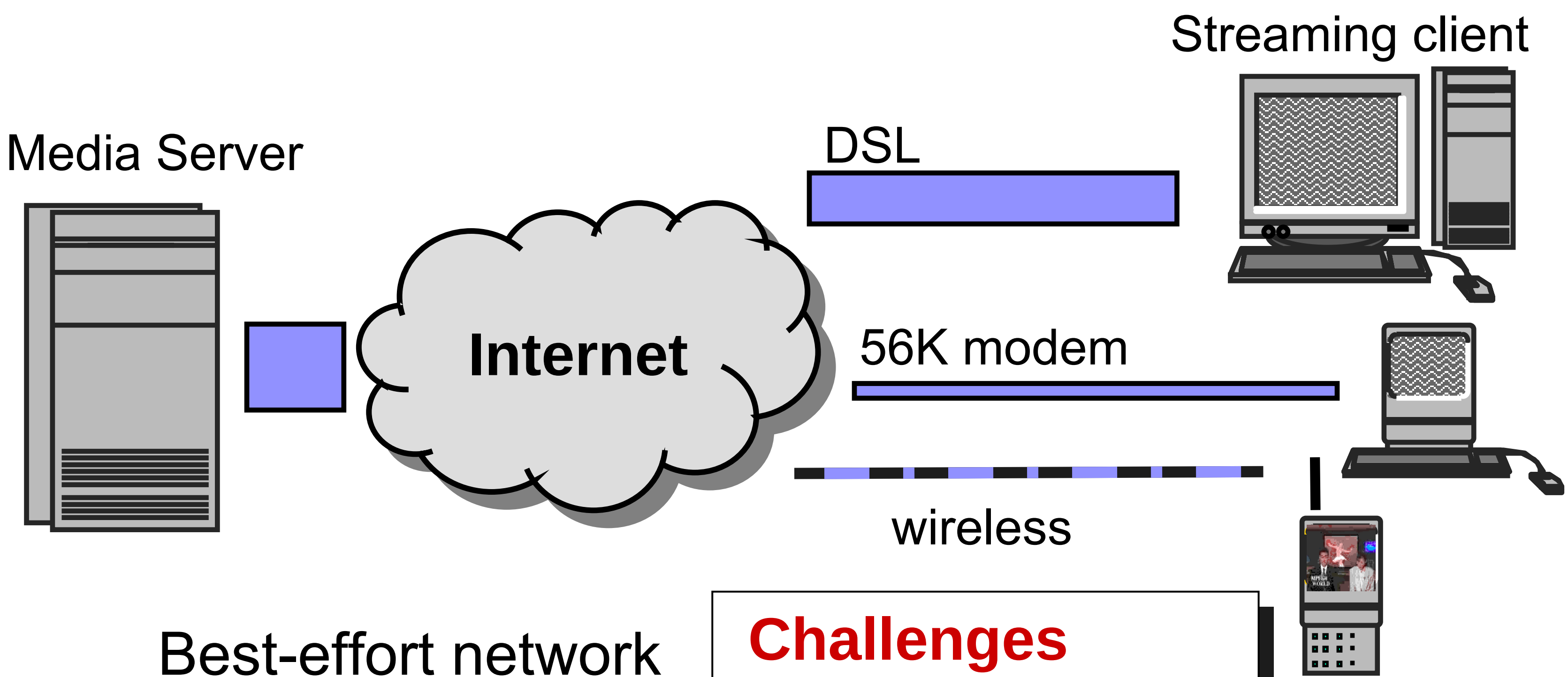
2000
- RealPlayer reaches 100 million users
- Akamai buys InterVu for $2.8B
- ***Internet stock market bubble bursts***
- WindowsMedia 7.0
- RealSystem 8.0

Desktop Computer CPU Power



[Girod 94]

Internet Media Streaming



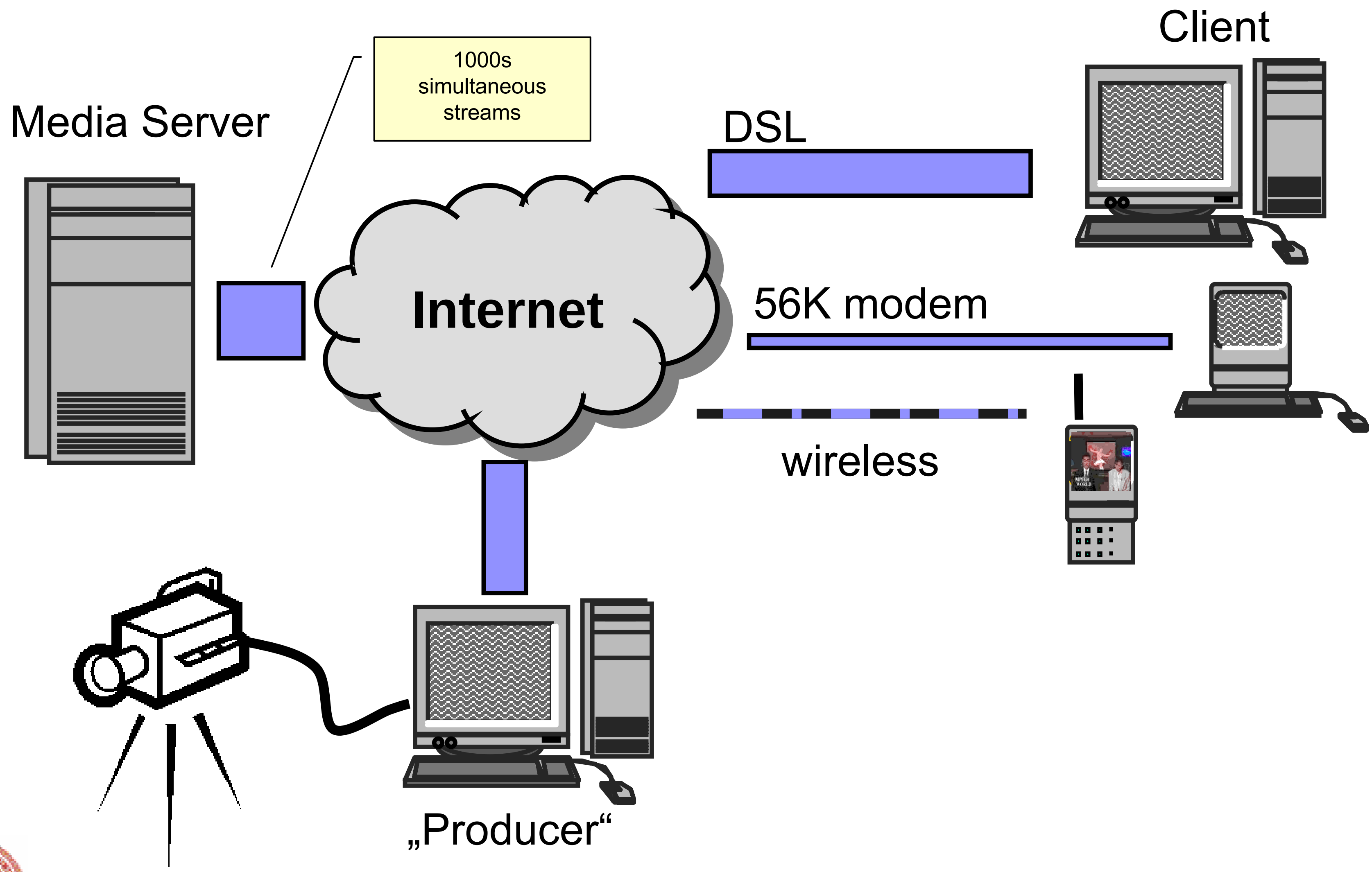
Best-effort network

- low bit-rate
- variable throughput
- variable loss
- variable delay

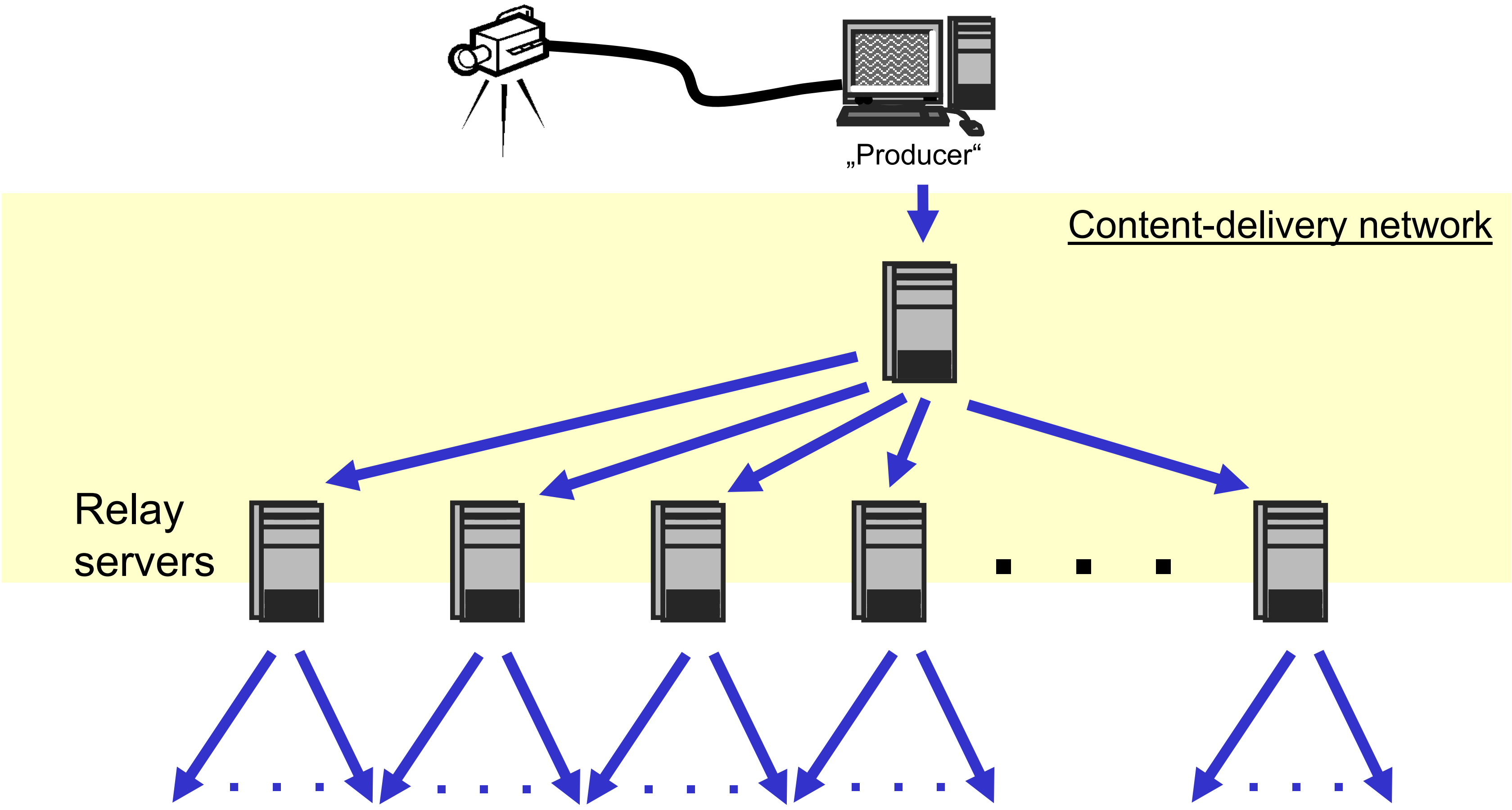
Challenges

- **compression**
- **rate scalability**
- **error resiliency**
- **low latency**

On-demand vs. live streaming

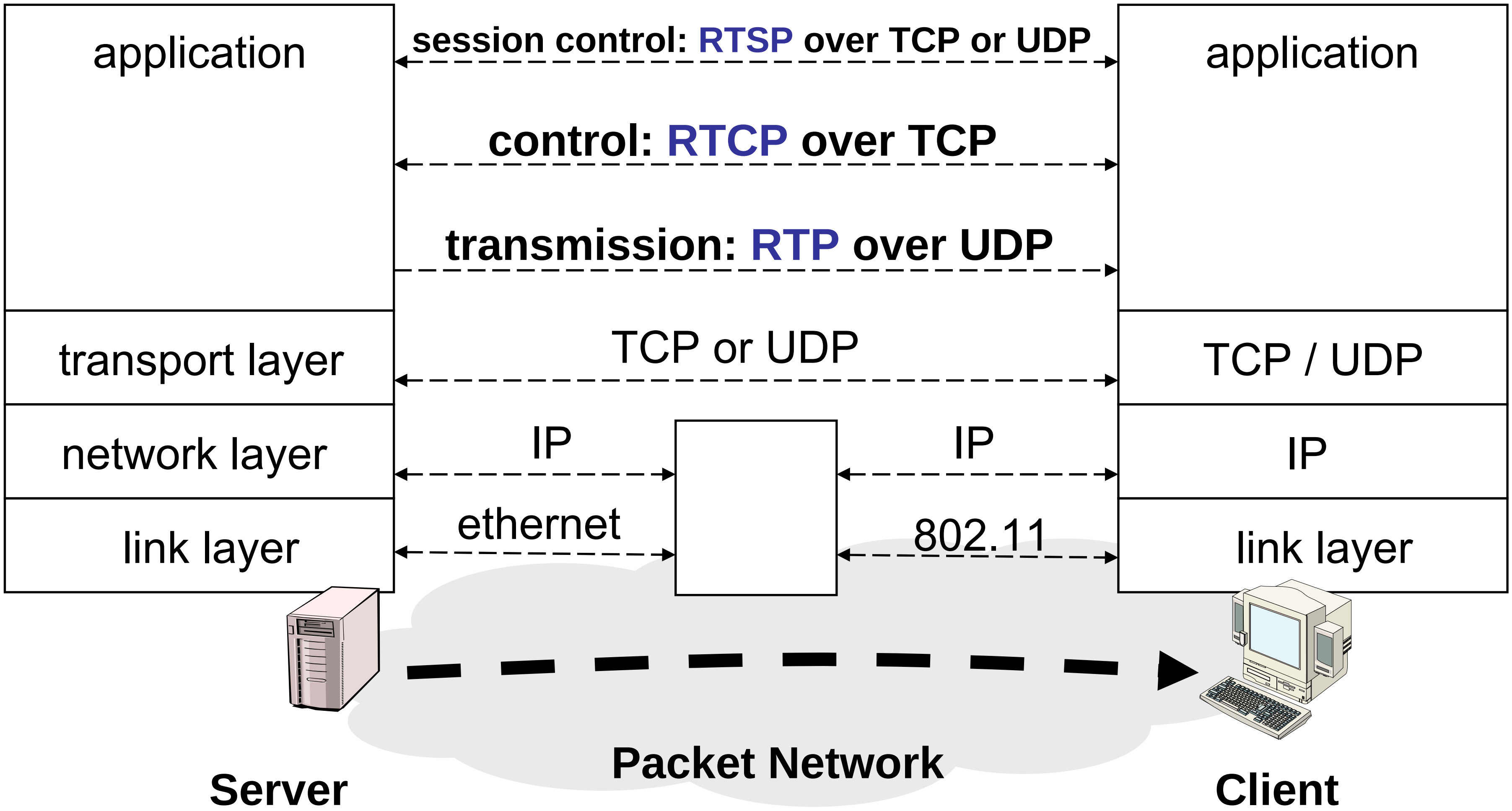


Live streaming to large audiences



“Pseudo-multicasting” by stream replication

Protocol Stack for Internet Streaming Media



RTP: A Transport Protocol for Real-Time Applications

- Defined by the IETF: RFC 1889
- Intended to provide a means of transporting real-time streams over Internet Protocol (IP) networks
- RTP packet

RTP header	Payload header	Payload

- RTP is session oriented (IP address and UDP port number)
- RTP provides data for the application to perform
  - Source identification
  - Packet loss detection and packet resequencing
  - Intra-media synchronization: playout with jitter buffer
  - Inter-media synchronization: e.g., lip-synch between audio and video
- IP/UDP/RTP header: 20+8+12=40 bytes

RTP Header Format

Sampling instant of the first data octet

- Multiple packets can have the same timestamp
- Not necessarily monotonic
- Used to synchronize different media streams

Payload type

Incremented by one for each RTP packet:

- Packet loss detection
- Restore packet sequence

```
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|V=2|P|X|  CC   |M|     PT      |       sequence number         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           timestamp                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           synchronization source (SSRC) identifier            |
+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
|            contributing source (CSRC) identifiers             |
|                             ....                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Identifies the synchronization source

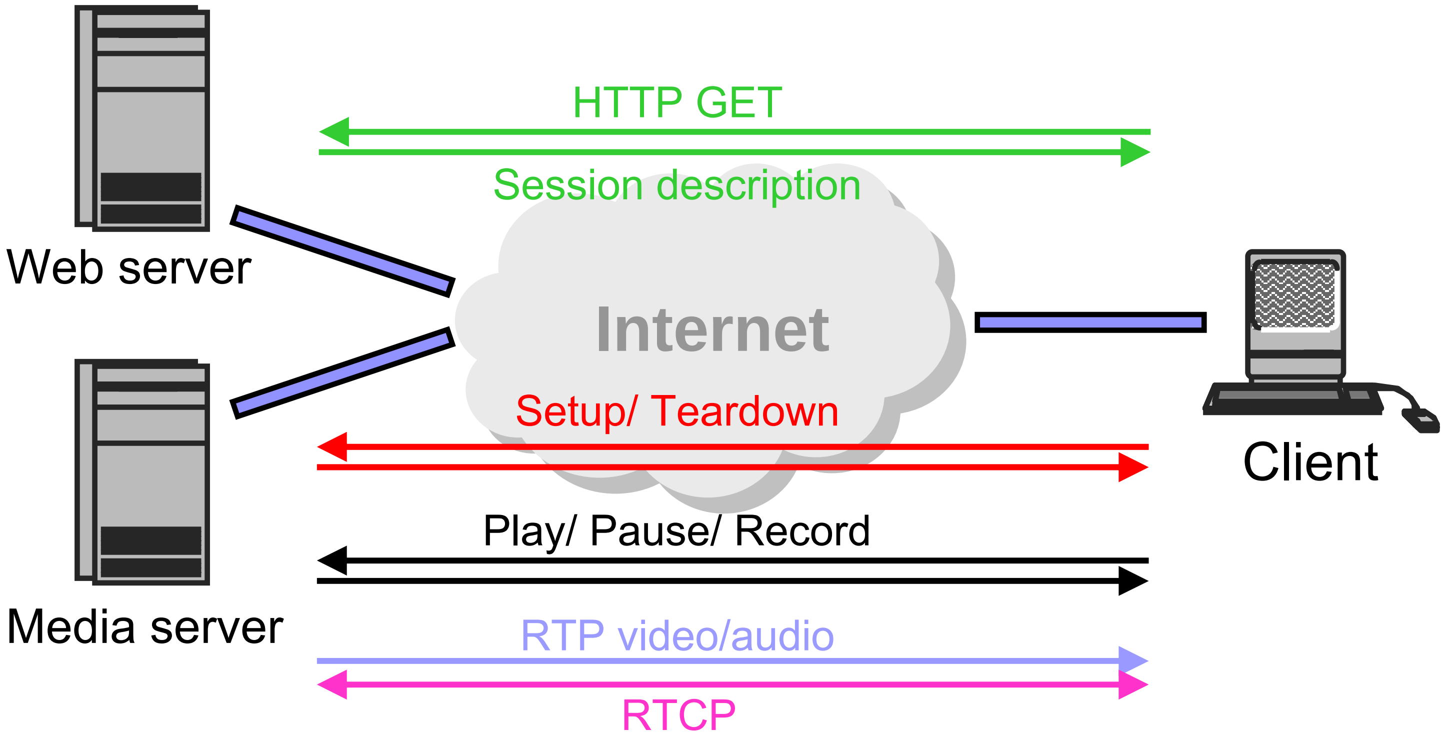
Identifies the contributing sources

RTCP (RTP Control Protocol)

- RTCP augments RTP by periodic transmission of control packets
- Feedback on the quality of data distribution
- Receiver reports (RR): statistics about the data received from a particular source
- Examples
  - Fraction of RTP data packets lost since the previous RR packet
  - Interarrival jitter: Estimate of the variance of the RTP data packet interarrival time distribution
  - RTP payload-specific feedback information, e.g.,
    - Intra-frame requests
    - Information about lost or damaged picture areas

Real Time Streaming Protocol

- Client-server multimedia presentation control protocol (RTSP: RFC 2326)
- Each presentation and media stream may be identified by a URL rtsp://
- RTSP also supports control of multicast events



Internet Congestion Control

- Network congestion causes burst loss and excessive delays
- All flow-control and error-control functions are left to the terminals
- Relies on voluntary fair sharing of network resources by sessions: TCP sets the standard
- For streaming media, it is required to dynamically adjust the streaming media bit-rate to match network conditions

TCP-friendly streaming

<u>Idea:</u> Explicitly estimate the rate that would be available to a TCP connection transferring data between the same source and destination TCP-friendly rate control

data rate

maximum transfer unit

$$r \approx \frac{1.22 \cdot MTU}{RTT \cdot \sqrt{p}}$$

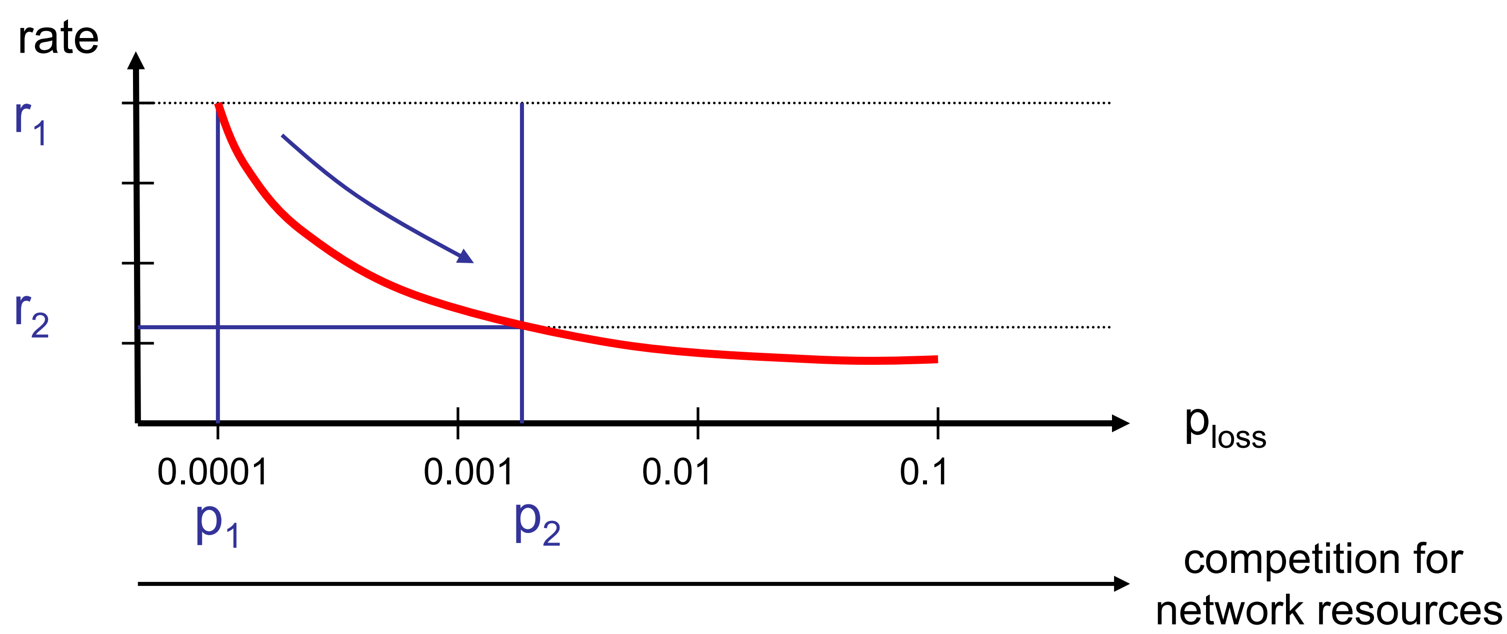
mean round trip time

mean packet loss rate

[Mahdavi, Floyd, 1997]

[Floyd, Handley, Padhye, Widmer, 2000]

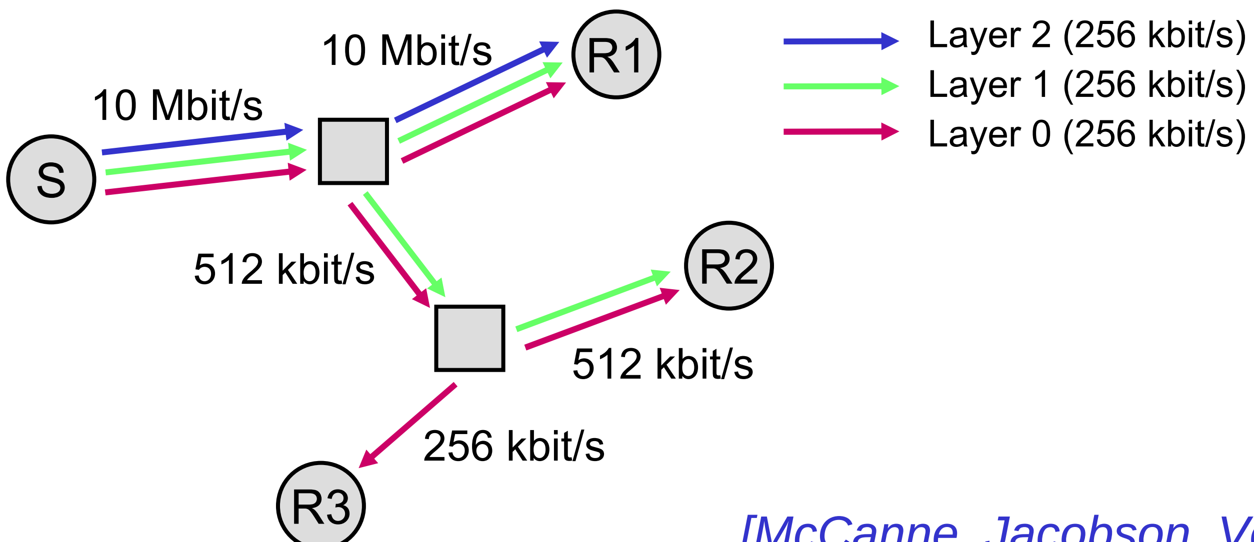
TCP-friendly streaming (cont.)



- Maximum packet size (MTU) known by source (e.g., 1500 Bytes for Ethernet)
- Mean round trip time from RTP timestamps
- Mean packet loss rate from RTCP receiver reports
- Constrain maximum data rate accordingly

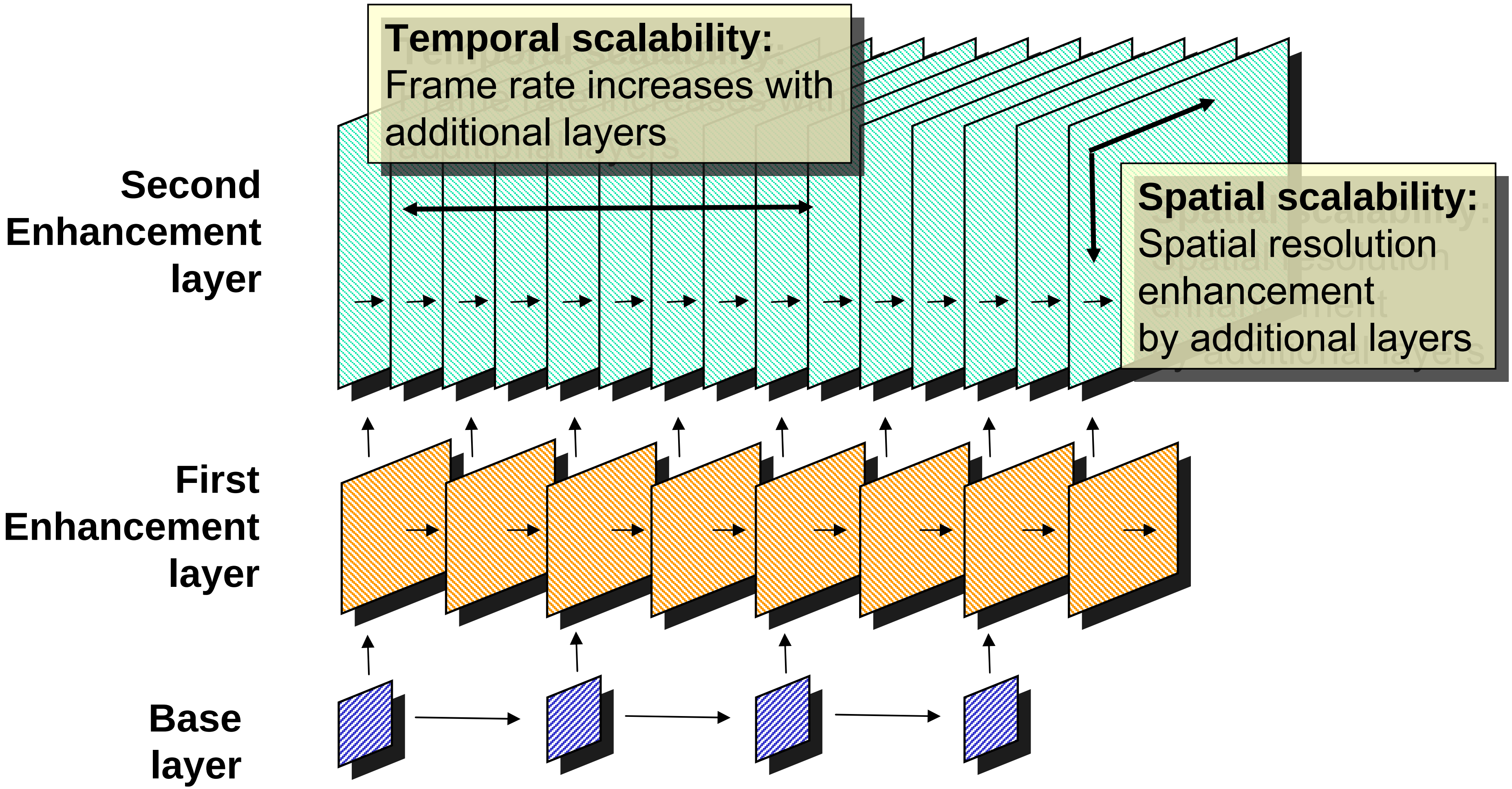
Receiver-Driven Layered Multicast

- Video and audio are encoded using layered, scalable scheme
- Different layers are transmitted on different multicast groups
- Each receiver subscribes to the base layer and depending on the available data rate to one or more enhancement layers
- Adaptation is carried out by joining or leaving groups

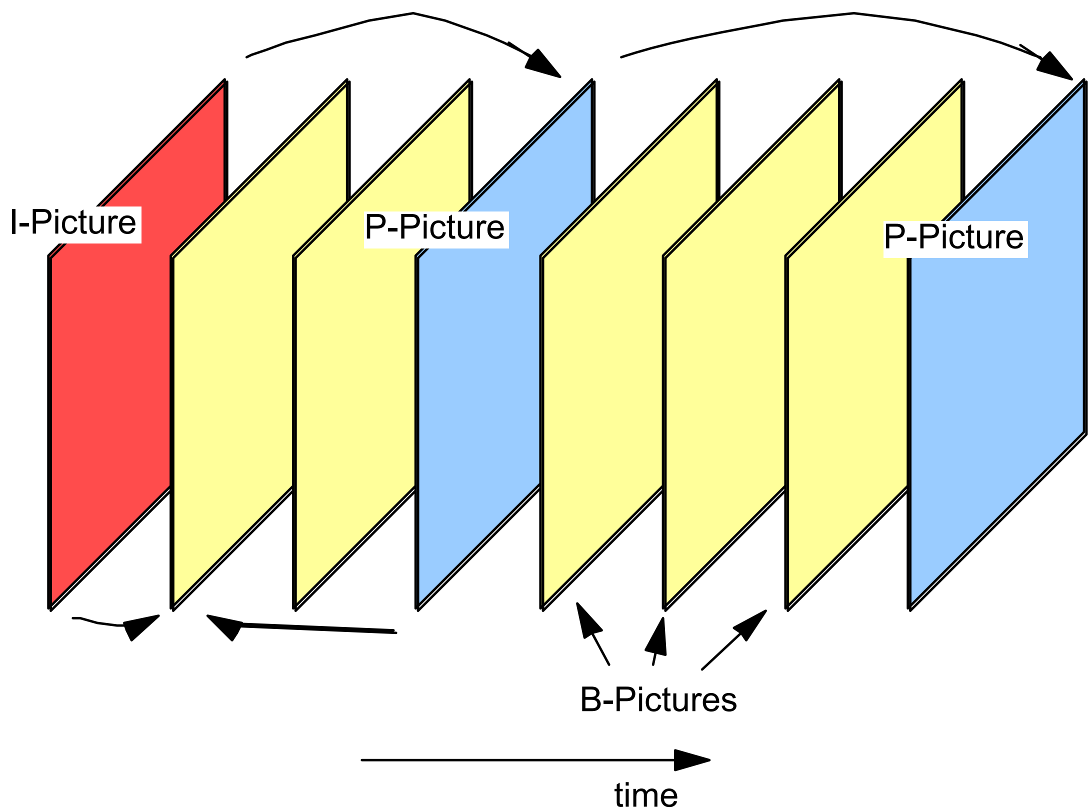


[McCanne, Jacobson, Vetterli, 96]

Layered Video Coding



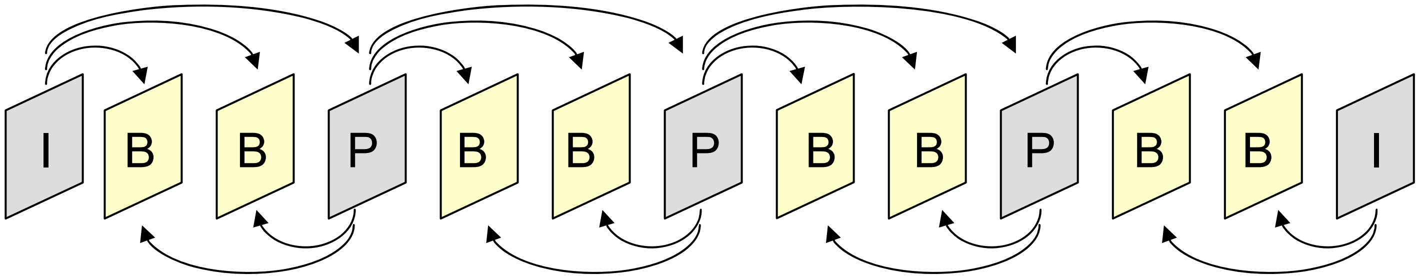
Hierarchical frame dependencies (MPEG, H.263)



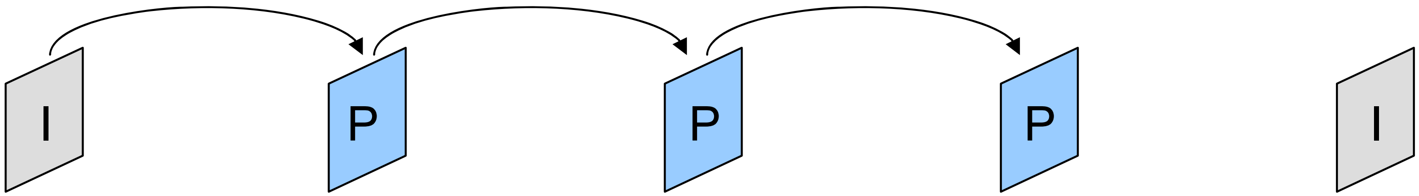
- Each I-picture starts a "Group of Pictures (GOP)" that can be decoded independently.
- Encoder can flexibly choose I-picture, P-pictures and B-pictures.
- B-pictures are not reference pictures for other pictures and hence can be dropped for temporal scalability.

Example layers with MPEG frame structure

- Base layer + first + second enhancement layer

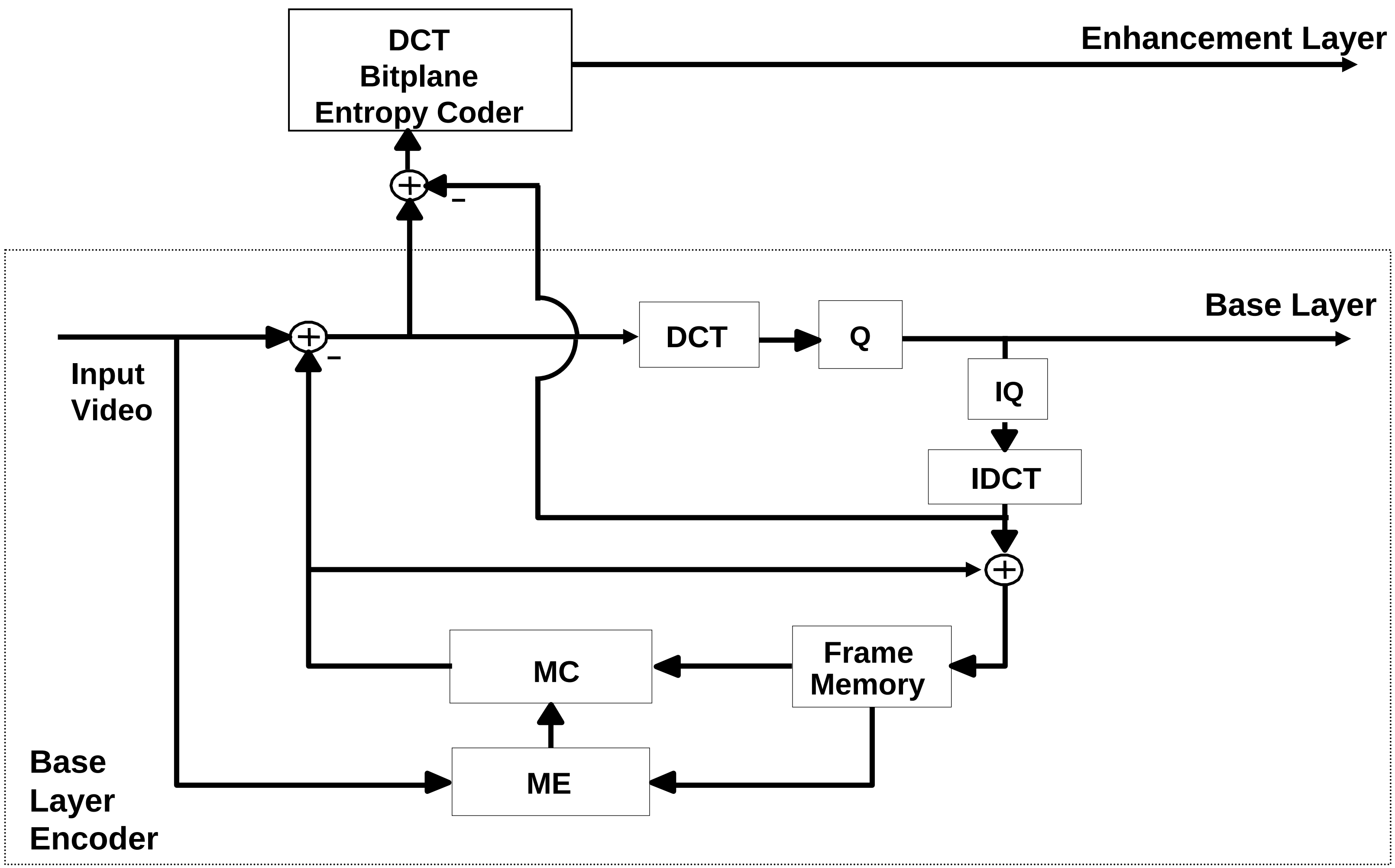


- Base layer + first enhancement layer

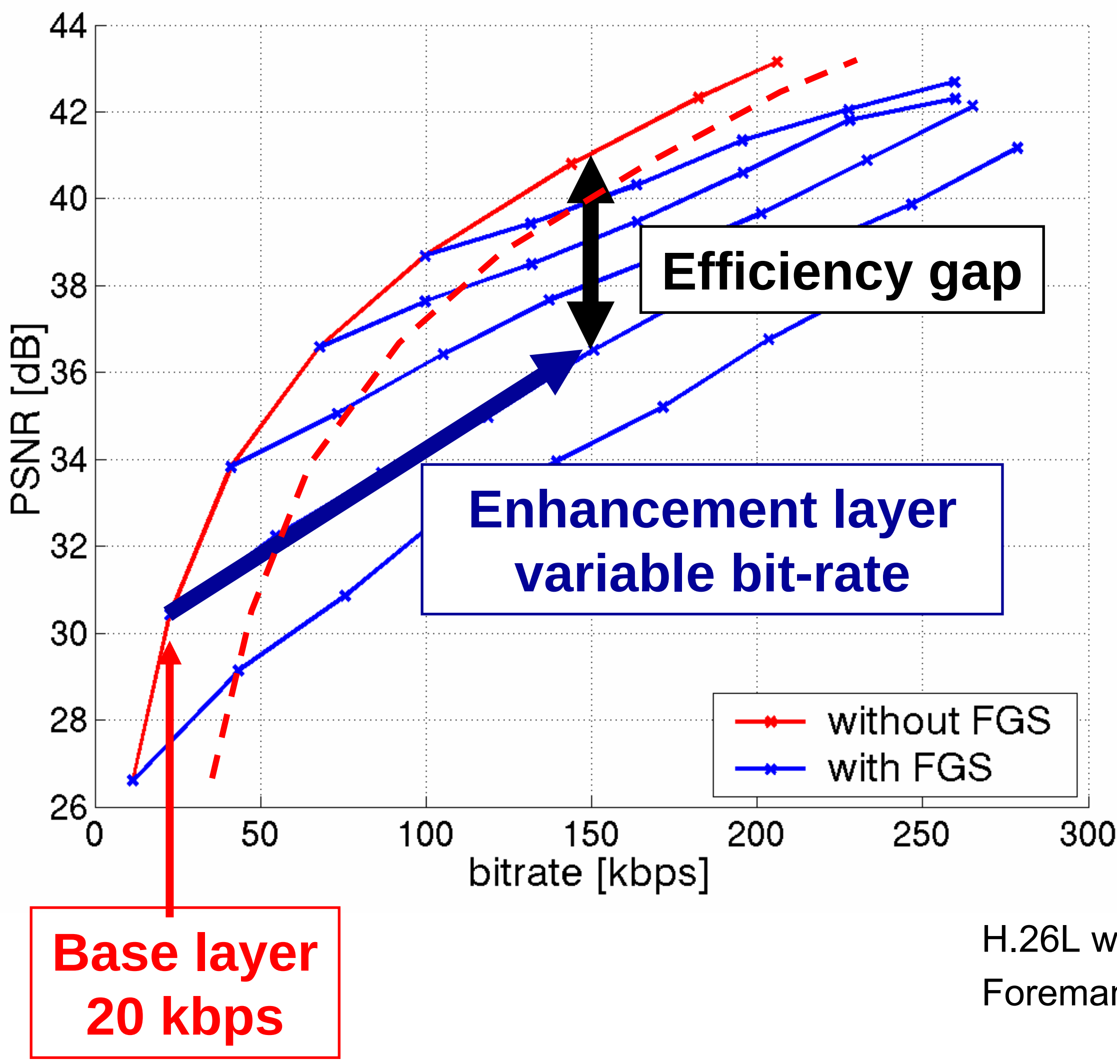


- Base layer

SNR Scalability: Fine Granular Scalability (FGS) for MPEG-4 Video

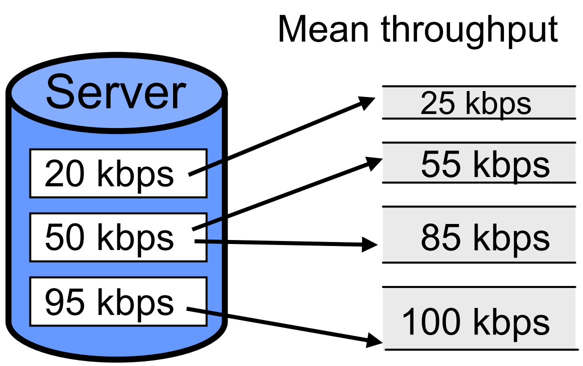


FGS is inefficient for low bit-rates

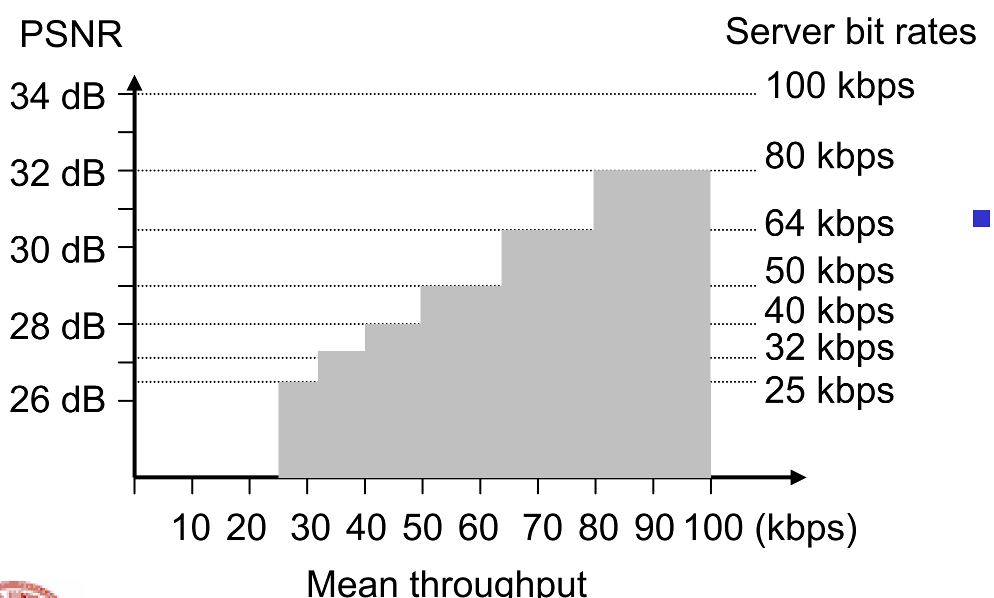


H.26L with/without FGS option
Foreman sequence (5fps)

Dynamic Stream Switching: SureStreams



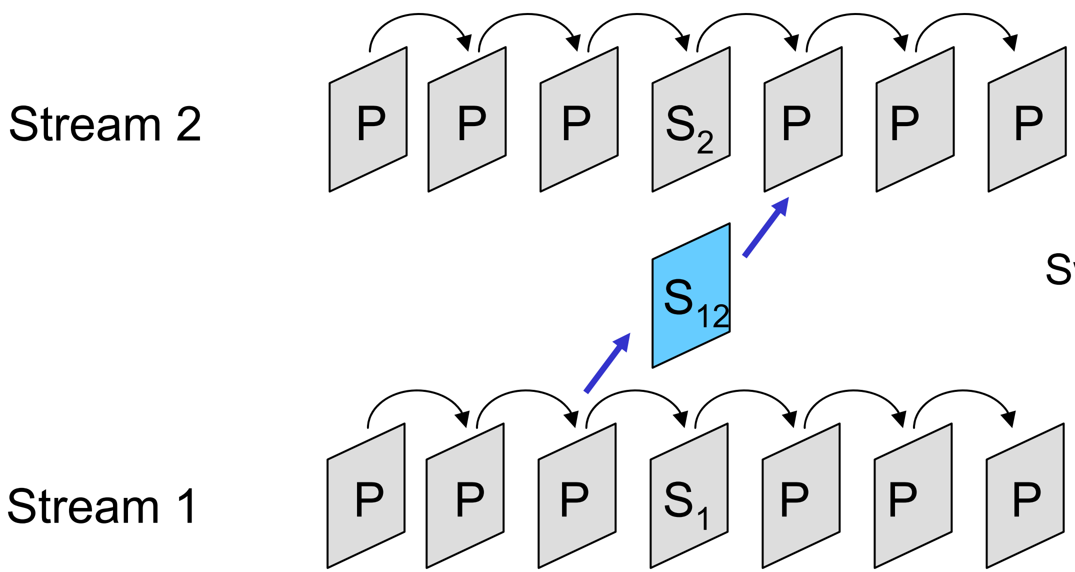
- SureStream Technology by RealNetworks *[Lippmann 99] [Conklin, Greenbaum, Lillevold, Lippman, Reznick, 2001]*
- Single-layer encoding at multiple target bitrates



- Illustration of operational area for 20% stream-to-stream rate difference

Dynamic Stream Switching: SP-frames

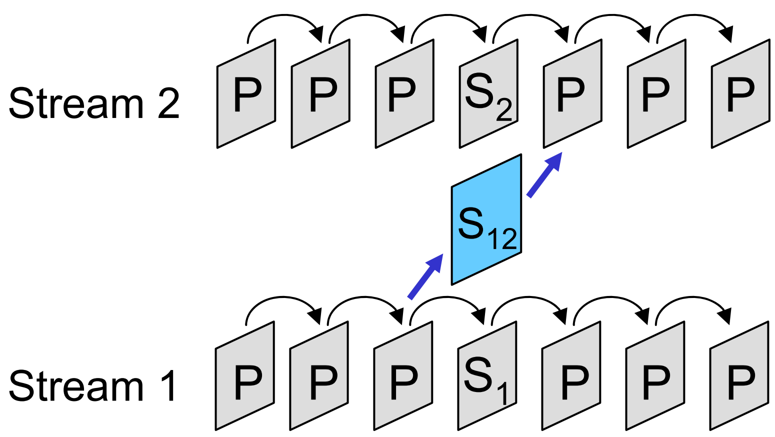
- SureStreams can only switch at the next I-frame
- S-frames *[Färber, Girod 97]*
- H.26L: SP-frames *[Karczewisz, Kurceren 01]*
  - SP-frames require fewer bits than I-frames
  - Identical SP-frames can be obtained even when different reference frames are used



Switching between bitstream 1 and 2 using SP-pictures

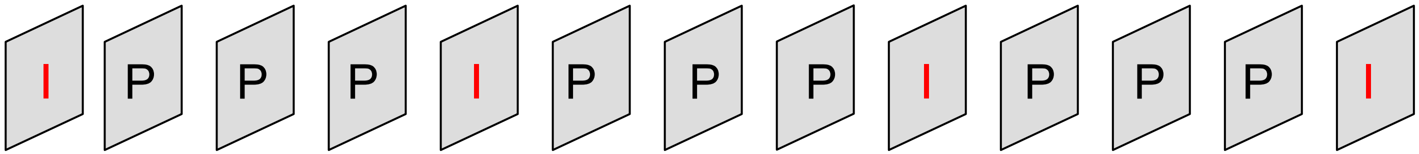
Dynamic Stream Switching: SP-frames (cont.)

- SP-frames are placed wherever one wants to enable switching from one stream to another
- When switching from Stream 1 to Stream 2, S_{12} is transmitted
- Although S_2 and S_{12} use different previously reconstructed frames as a reference, their reconstructed values are identical
- No error introduced
- SP-frames have lower coding efficiency than P-frames but significantly higher coding efficiency than I-frames

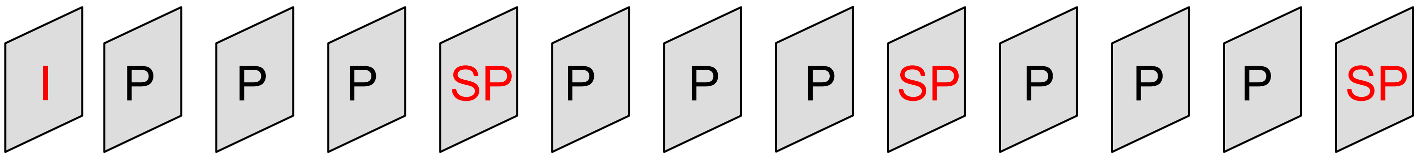


SP-frames: performance gain

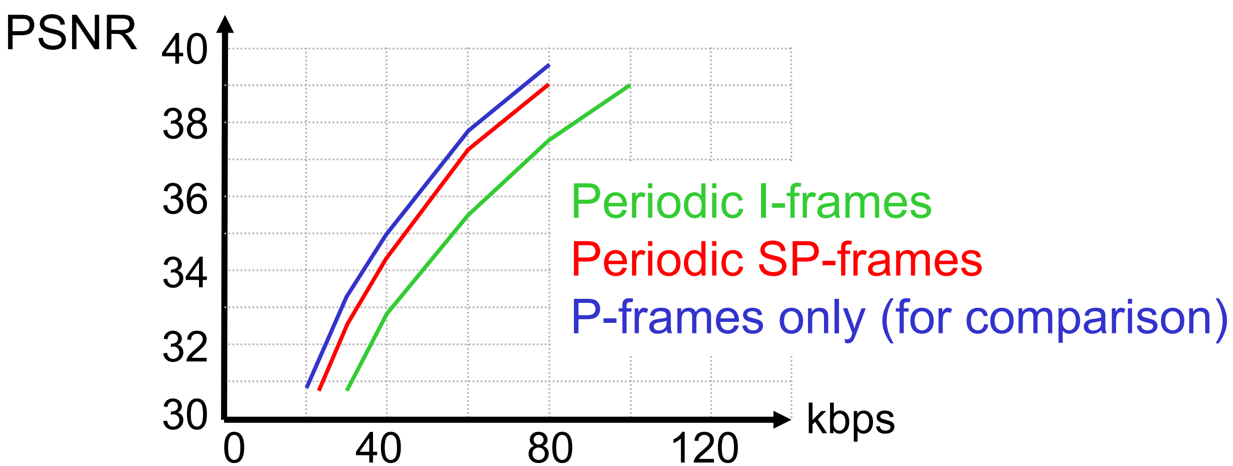
- Periodic insertion of I-frames



- Periodic insertion of SP-frames



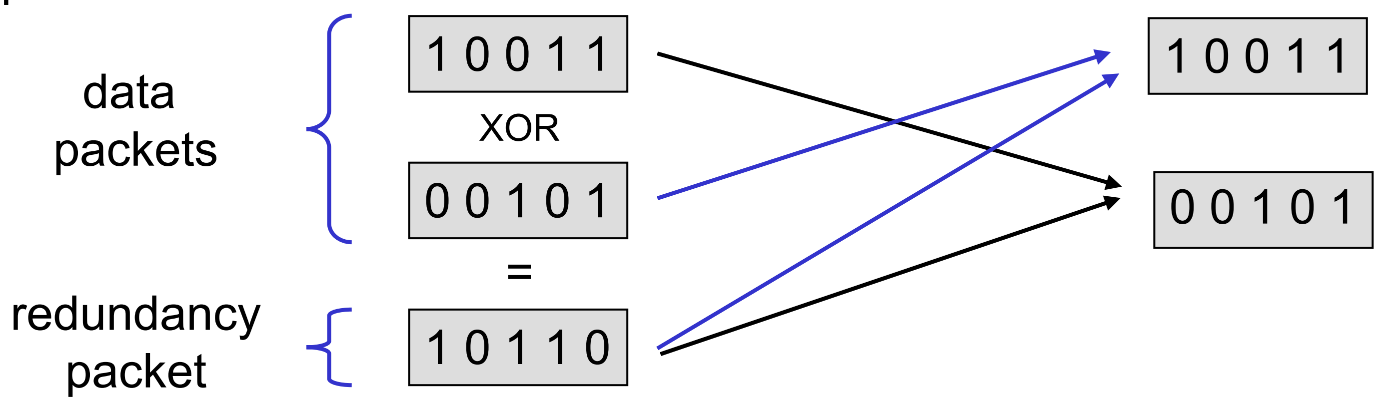
- I-frames or SP-frames every second for test sequence „News“



From: [Karczewisz, Kurceren 01]

Forward Error Correction

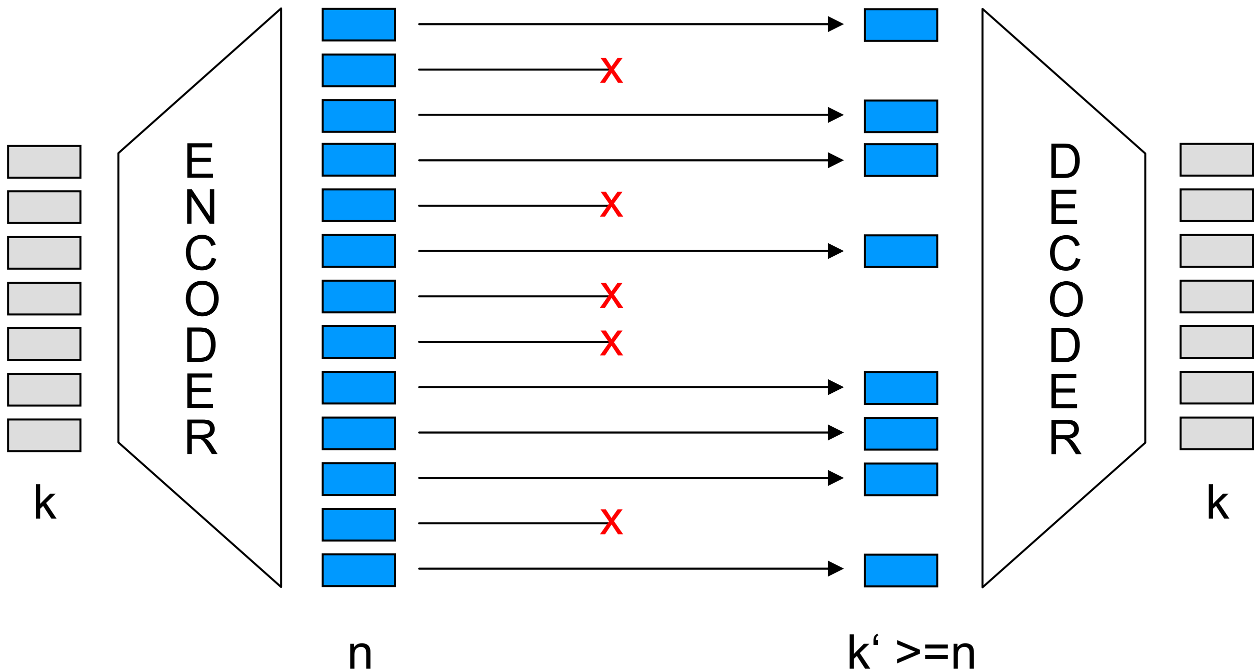
- For packet-based transmission, FEC can be employed across packets (erasure decoding)
- Erasures → the exact position of missing data is known
- Transmission of redundant data for recovery of lost packets at the receiver (redundancy packets)
- Exclusive OR (XOR) allows to compute one parity packet for a set of original packets



- RFC 2733: An RTP Payload Format for Generic Forward Error Correction
  - Media independent
  - XOR-based

Erasure Codes

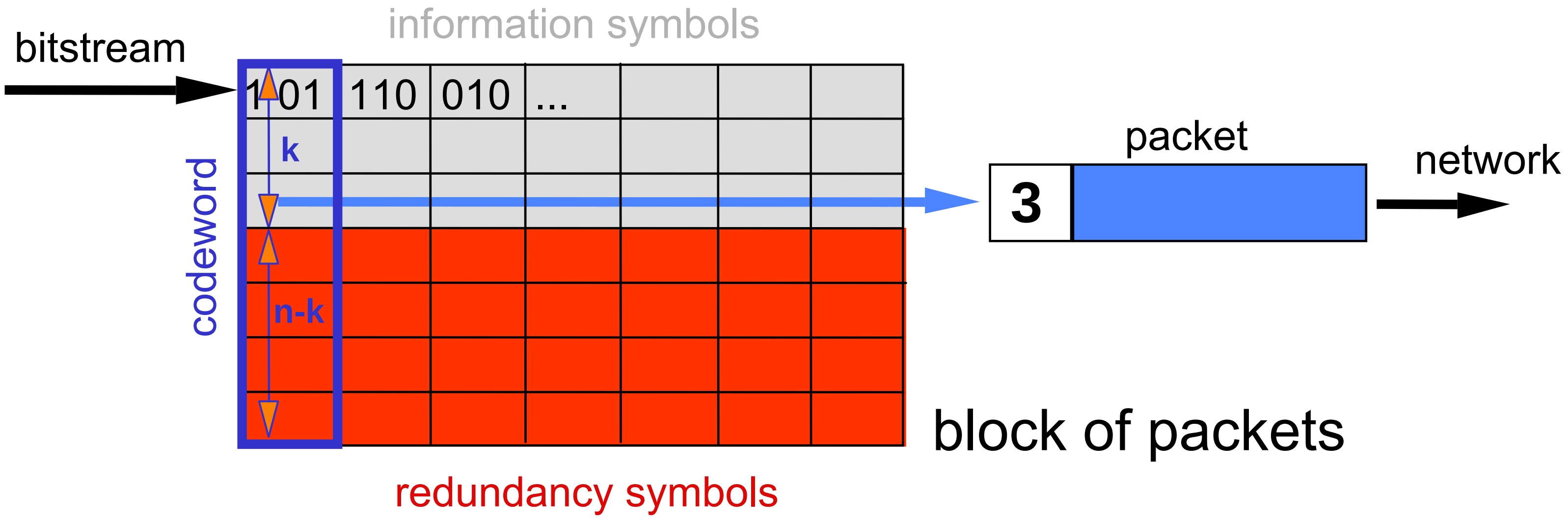
- Idea: k blocks of source data are encoded at the sender to produce n blocks of encoded data in such a way that any subset of k received blocks suffices to reconstruct the source data



from [Rizzo 97], for more info [Blahut 84],[Lin, Costello 83]

Erasure Codes: Packet Loss Protection

- k information packets, n-k redundancy packets
- Resulting n packets are called block of packets (BOP)
- Packets are the rows of the BOP
- Codewords are calculated across the columns, e.g., Reed-Solomon codes over GF(2^8)
- No additional delay at the sender (information packets can be sent immediately)

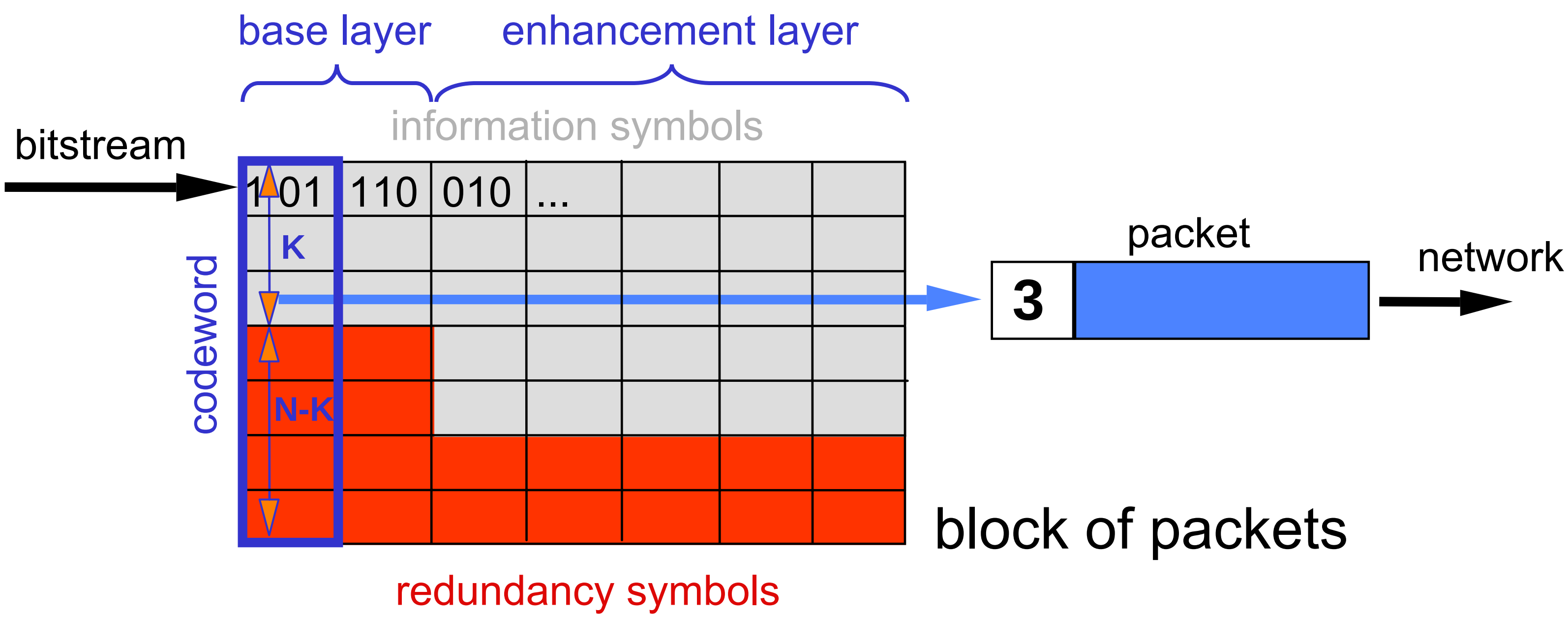


FEC performance

- FEC is the preferred error-control scheme for multicast or low-latency streaming applications
- The reconstruction delay at the receiver increases with k
- Parity packets are particularly efficient for multicast since a single parity packet can repair the loss of different data packets seen by different receivers
- Relationship between FEC and congestion control (CC)
  - CC reduces network load for high error rates
  - FEC increases redundancy for high error rates
  - Contradicting approaches
  - Solution: FEC in combination with rate control

Priority Encoding Transmission

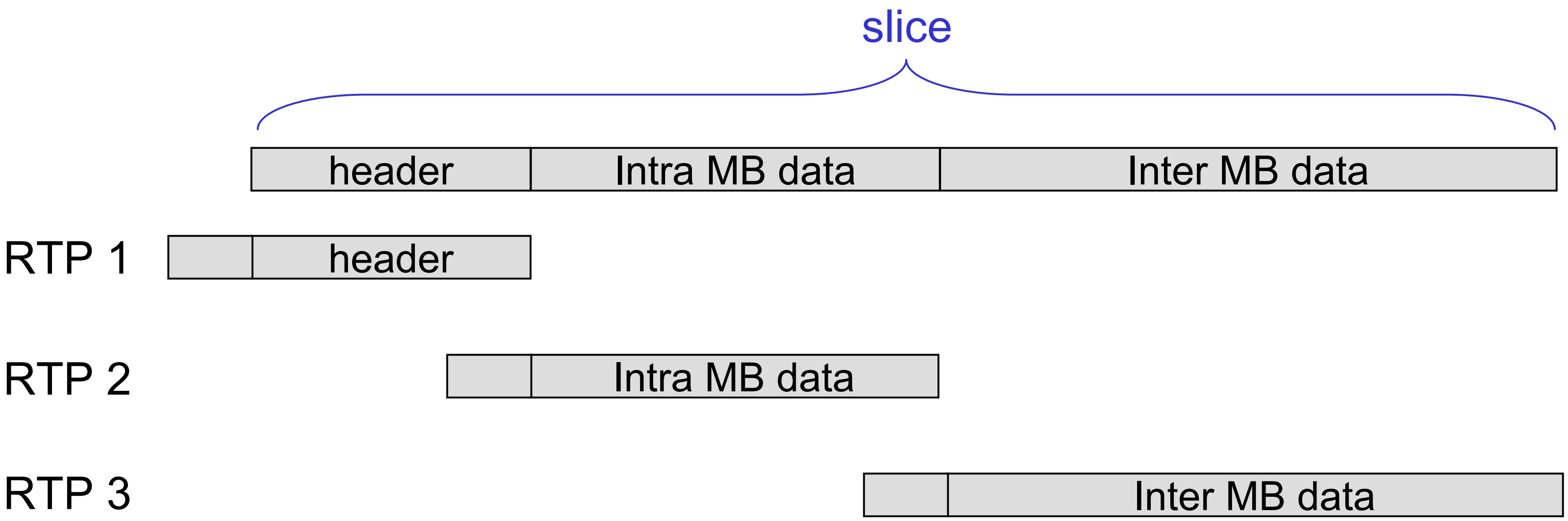
- Specify different priorities for different data segments
- According to the assigned priority, PET generates different amount of redundancy
- Example: Protect I frames more than P frames more than B frames (100%, 33%, 5%)
- Example: PET in combination with scalable coding *[Horn, Girod 99]*



[Albanese, Blömer, Edmonds, Luby, Sudan 96]

Data partitioning

- Without data partitioning: RTP packet contains full slice as payload
- With data partitioning



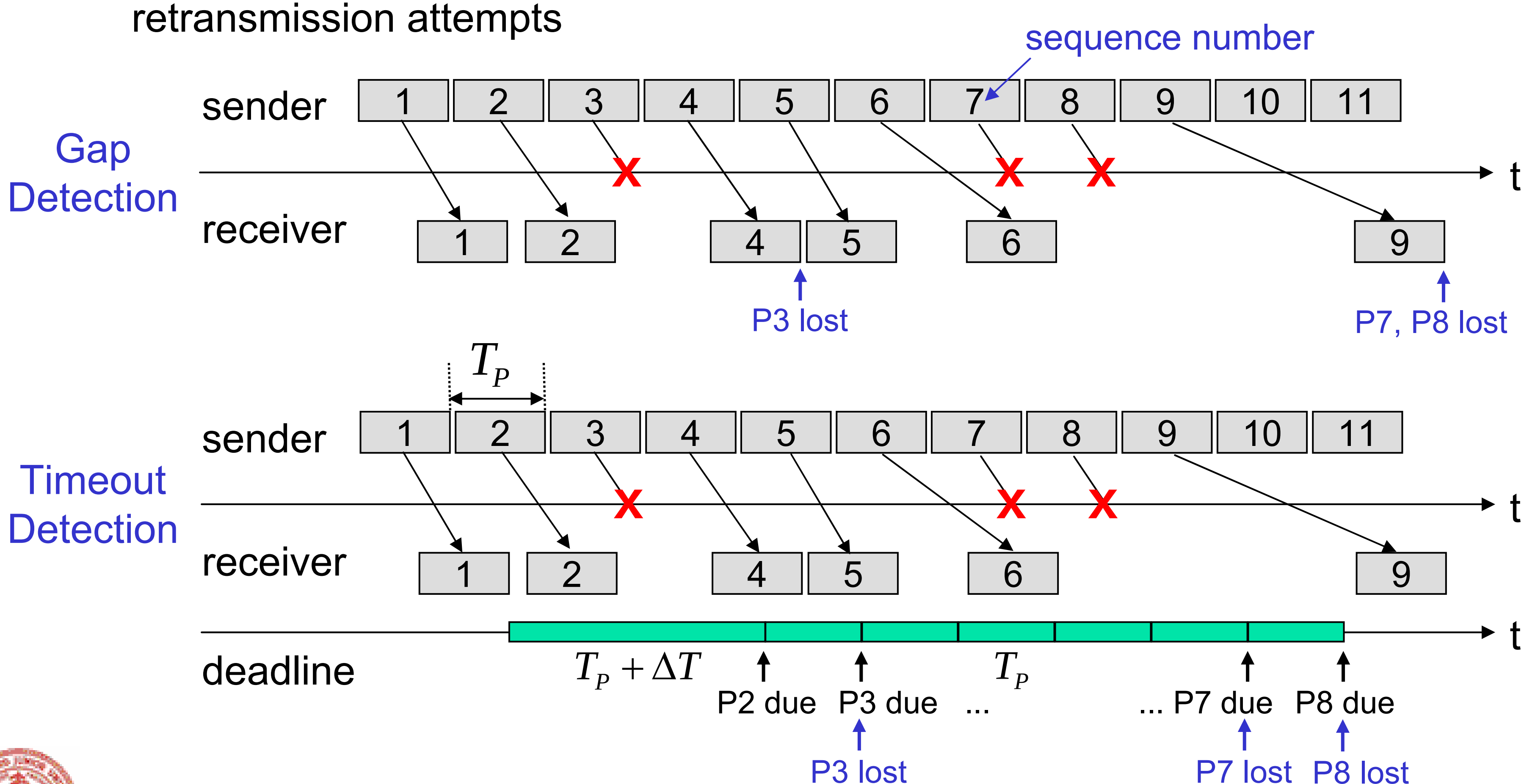
- Prioritization or FEC for more important packets

Automatic Repeat reQuest (ARQ)

- Missing packets are retransmitted upon timeouts or explicit requests from the receiver
- ARQ-based schemes consist of three parts
  - Packet loss detection
  - Acknowledgment strategy
    - Indicate which data have been received (positive ACKs)
    - Indicate which data are missing (negative ACKs or NACKs)
  - Retransmission strategy
    - Go-Back N
    - Selective Retransmission
    - Trade-off simplicity of the receiver implementation and transmission efficiency

Packet Loss Detection

- Retransmitted packets must arrive at the receiver before playout deadline
- Early detection of packet loss is the key to maximize the number of retransmission attempts

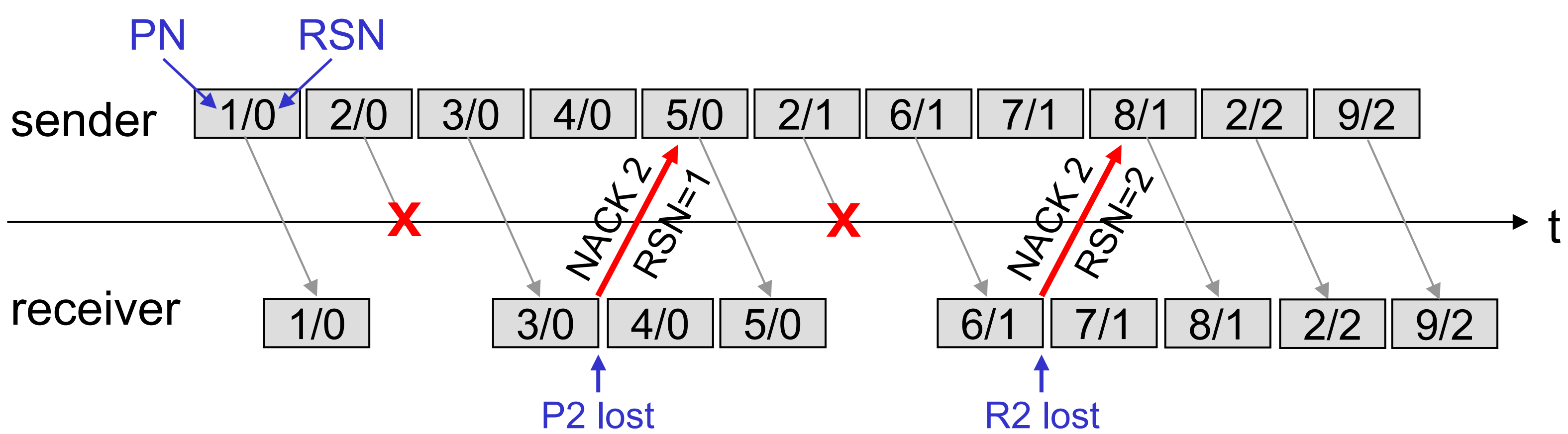


Packet Loss Detection

- Gap detection
  - Detection delay depends on the inter-packet time
  - Packet loss often occurs in bursts → larger gaps
- Timeout detection
  - Limited applicability for large delay jitter
- Combination
  - NACK is sent when either scheme declares packet to be lost
- Nice extension in *[Sze, Liew, Lee 01]*
  - Gap detection even for retransmitted packets

Gap Detection for Retransmitted Packets

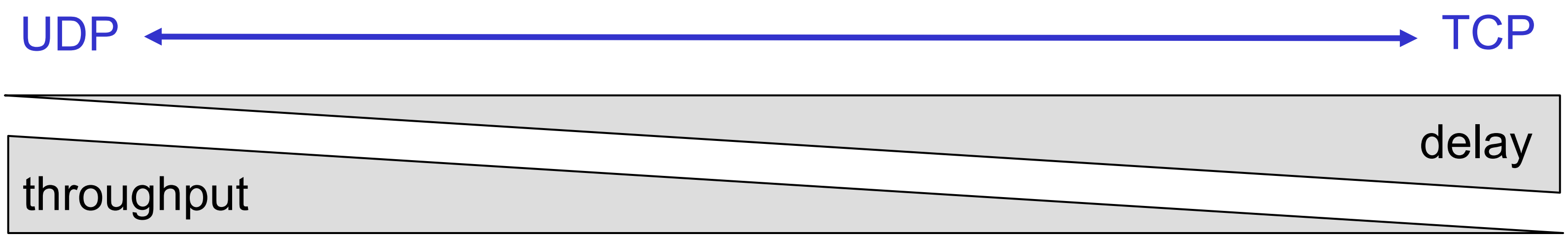
- Retransmission sequence number (RSN) in all packets
- The retransmitted packet and all subsequent ordinary packets will be marked with the RSN until the next NACK arrives
- The retransmitted packet corresponding to the NACK should be the first packet to arrive at the receiver with the new RSN



[Sze, Liew, Lee 01]

Partially reliable transport

- Instead of trying retransmission indefinitely to recover missing packets, the number of retransmissions can be limited *[Marasli, Amer, Conrad 96]*
  - Limit on maximum number of retransmissions
  - Limit on maximum delay



No reliability

Partial reliability respects the loss tolerance of the application

1) Detect lost packet
2) Decide whether or not to recover it

Full reliability at the cost of increased delay and reduced throughput

Delay-constrained retransmission

- Receiver-based: request to retransmit packet N if

current time　　safety interval

$$T_c + RTT + \Delta T < T_d(N)$$

round trip time estimate　　playout deadline for packet N

- Sender-based: retransmit packet N if

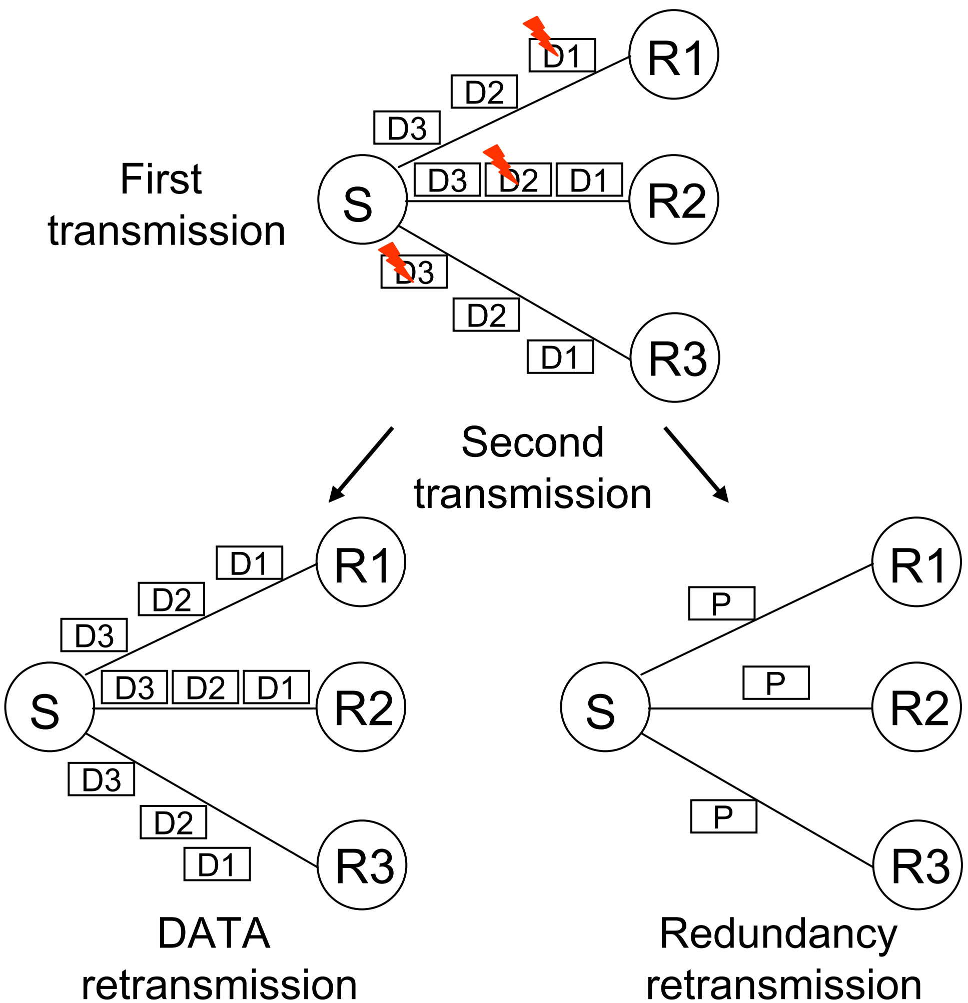
$$T_c + \frac{RTT}{2} + \Delta T < T_d'(N)$$

estimate of playout deadline

FEC versus ARQ

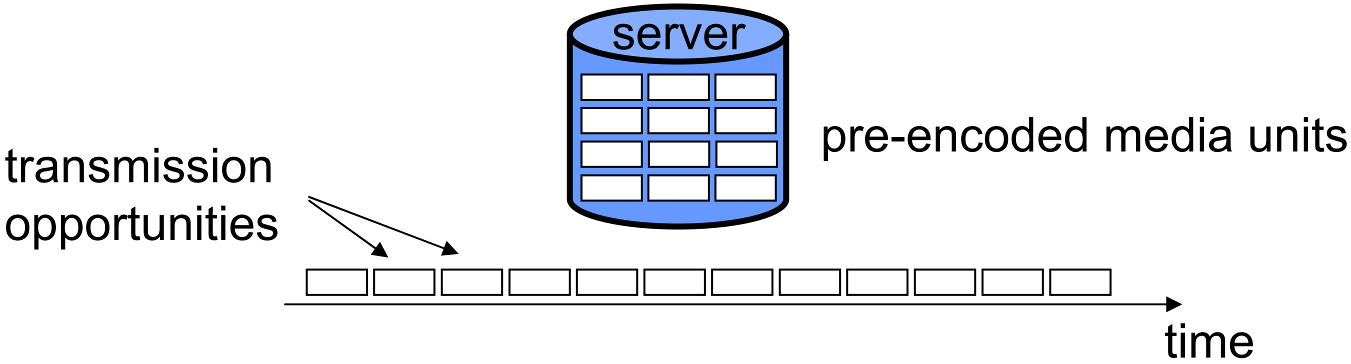
- Open-loop error control with FEC
  - No feedback required
  - Suitable for large groups, large RTTs, stringent delay requirements
  - Individual loss dominates: Transmission of redundant packets can be used to allow the receivers to recover from independent packet losses
  - Redundancy determined by maximum loss probability
- Retransmission-based error control
  - Suitable for unicast or small groups
  - Feedback explosion for large groups
  - Error recovery delay depends on RTT
  - Non-interactive application, relaxed delay requirements
  - Automatic adaptation to varying packet loss rates

Hybrid Error Control (ARQ/FEC)



- A major difficulty when using FEC is to choose the right amount of redundancy
- Hybrid ARQ type II *[Wicker 95, Nonnenmacher, Biersack, Towsley, 97]*
  - No redundancy with the first transmission
  - Send parity packets after request for retransmission
- Efficient for reliable multicast to a large number of receivers

Rate-Distortion Optimized Streaming (RaDiO)

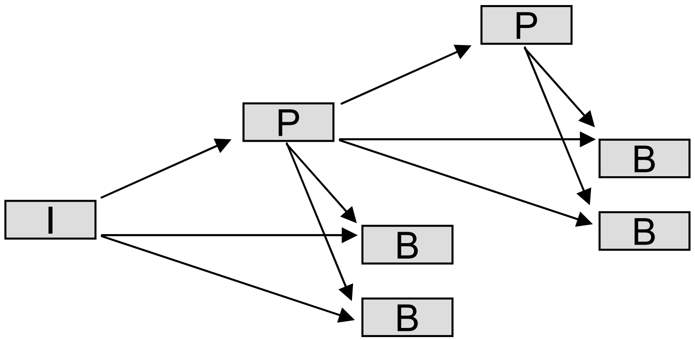


- Media unit is put into packet for transmission
- Packet may be retransmitted or sent multiple times
- Requirements
  - Meet target rate
  - Maximize reconstruction quality
- Packet scheduling problem: which packets should be selected for transmission and when?
- Rate-distortion framework proposed, e.g., in *[Podolsky, McCanne, Vetterli 2000] [Miao, Ortega 2000] [Chou, Miao 2001]*

RaDiO: Rate-Distortion characterization

- Example packet dependencies
  - I P P P P P P P P

  - I B B P B B P B B

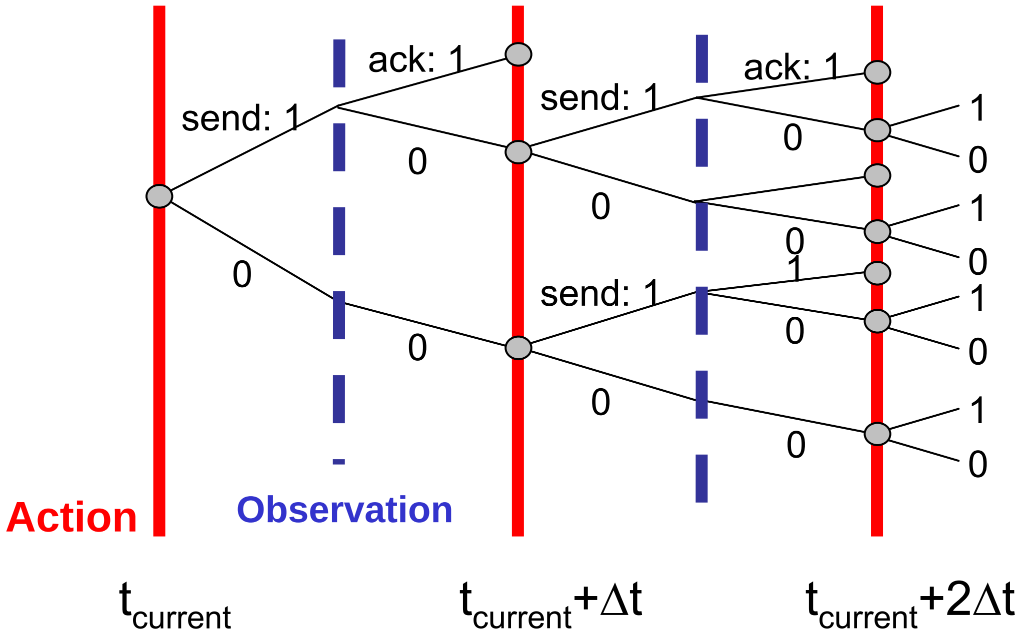


- Describe packet *n*
  - Size in bytes B_n
  - Distortion reduction Δd_n
  - Delivery deadline t_n

[Chou, Miao 2001]

RaDiO: Decision Tree with Finite Time Horizon

- Markov decision tree for one packet



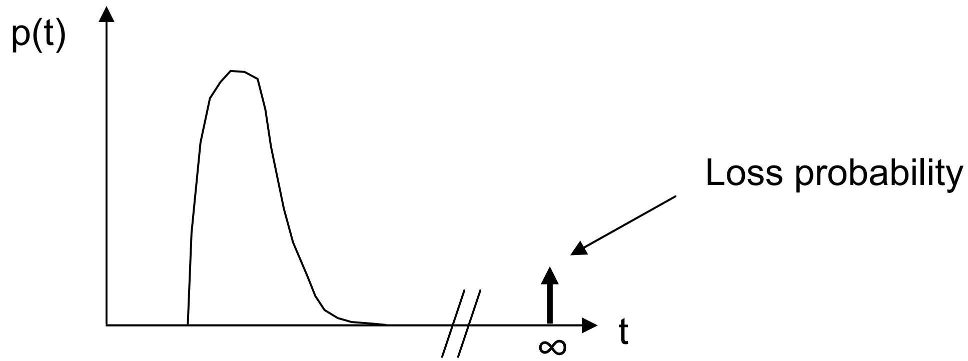
... N transmission opportunities before deadline

- Construct combined tree for all packets
  - Limit the number of packets sent per transmission opportunity
  - Omit inefficient subtrees (not on convex hull in RD plane)

[Chou, Miao 2001]

RaDiO: Observation Probability Model

- Assign observation state transition probabilities using packet delay and loss model



- Typical assumptions:
  - Identical, independent delay/loss pdfs for each transmission opportunity
  - Delay/loss pdf independent from RaDiO actions (no self-congestion)

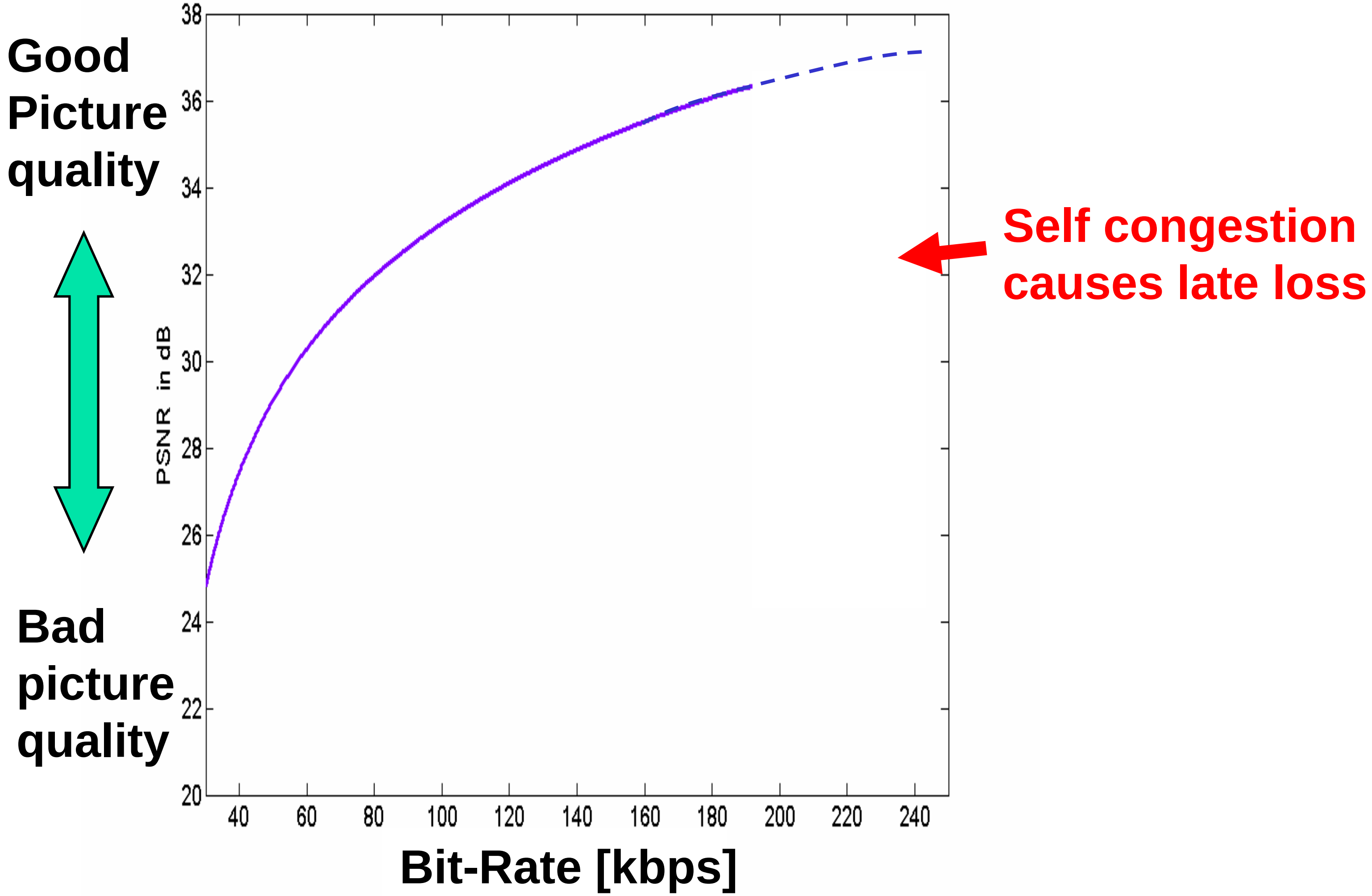
[Chou, Miao 2001]

RaDiO: Determine Optimum Decision

- Consider entire sequence of actions between now and time horizon
- Minimize Lagrangian cost function $J = D + \lambda R$, e.g., iteratively by considering one action at a time
- Calculate expected distortion D for each sequence of actions, considering packet dependencies, delay distribution, and acknowledgment probabilities
- Calculate expected rate R for each sequence of actions, considering delay distribution and acknowledgment probabilities
- Repeat for each transmission opportunity

[Chou, Miao 2001]

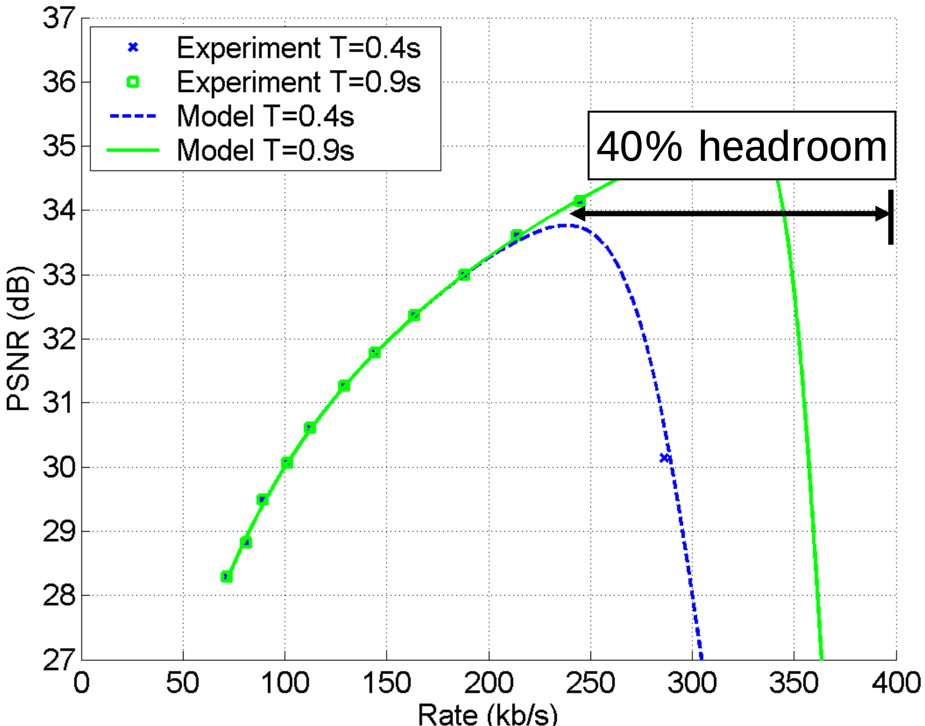
Video Distortion with Self Congestion



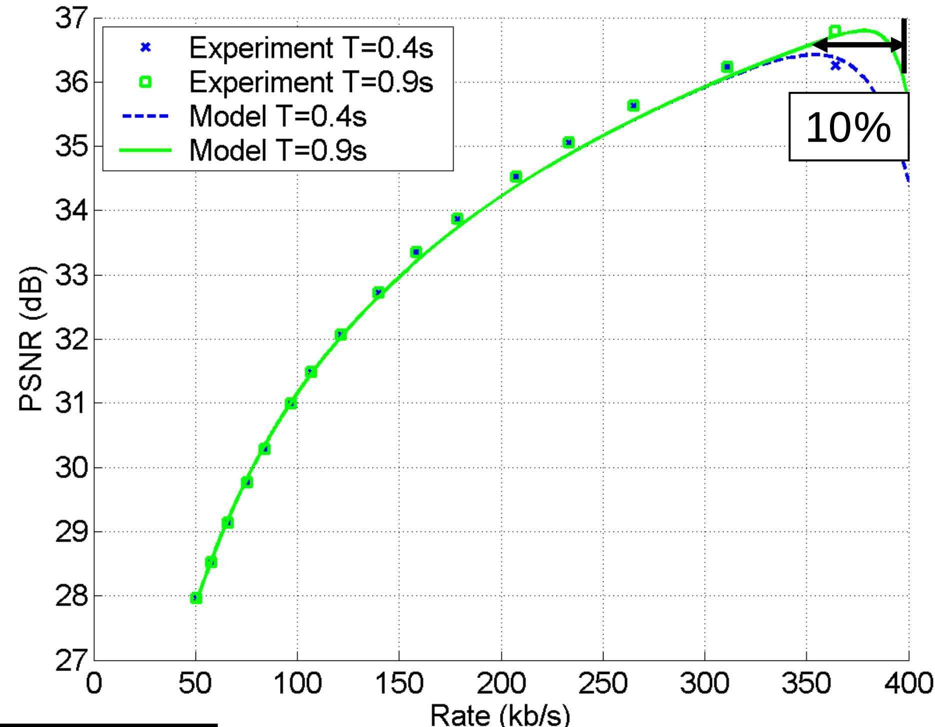
Effect of Playout Delay and Loss Sensitivity

$$D = D_0 + \frac{\theta}{(R - R_0)} + \kappa e^{-(C-R)T/L}$$

Foreman



Salesman

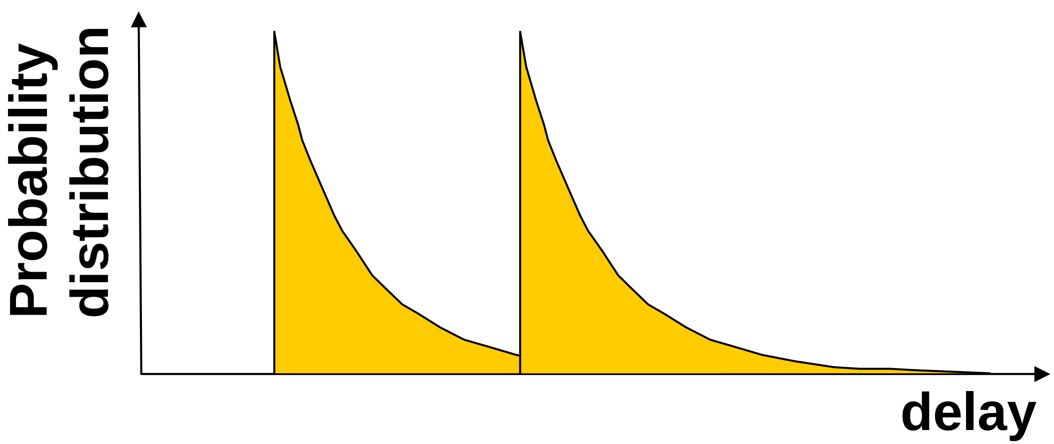


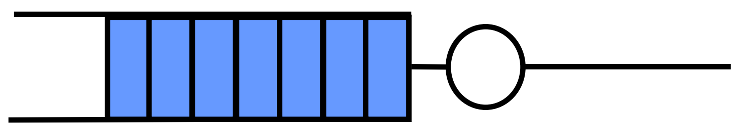
Simulations over ns-2

Link capacity 400 kb/s

Modeling Self-Congestion for Packet Scheduling

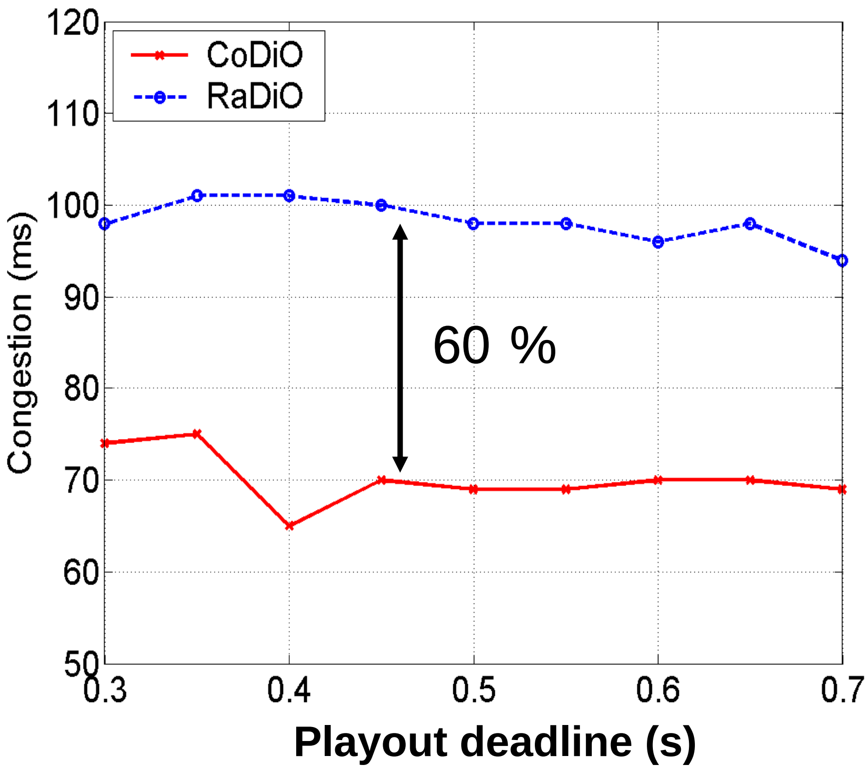
- Rate-distortion optimized packet scheduling (RaDiO) typically assumes independent delay pdfs for successive packet transmissions *[Chou, Miao, 2001]*
- Model delay pdf by exponential with varying shift





[Setton, Girod, 2004]

CoDiO vs. RaDiO



Sequence: Foreman

Packet loss rate 2%

Link capacity 400 kb/s

Propagation delay: 50ms