

# Networking Issues for Multimedia Delivery



Trac D. Tran

ECE Department

The Johns Hopkins University

Baltimore, MD 21218



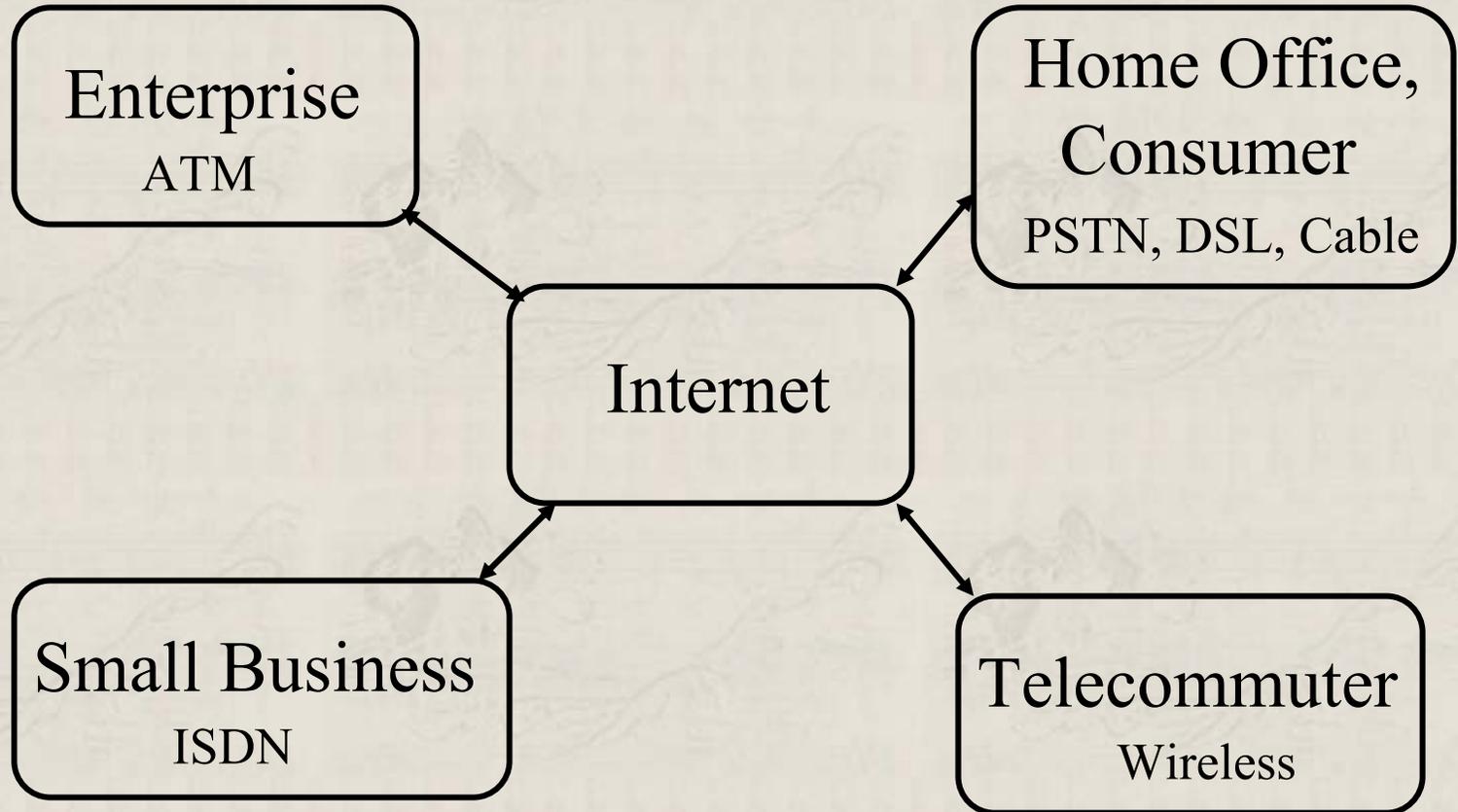
# Outline

---

- ◆ Considerations in networked multimedia
- ◆ Network types and examples
- ◆ IP networks: TCP, UDP, RTP/RTCP...
- ◆ Source versus channel coding
- ◆ Error resilience,
  - error recovery
  - error concealment
  - Layered coding, multiple description coding
  - Pre/post-processing techniques

# Network Characteristics

---



# Considerations for Multimedia

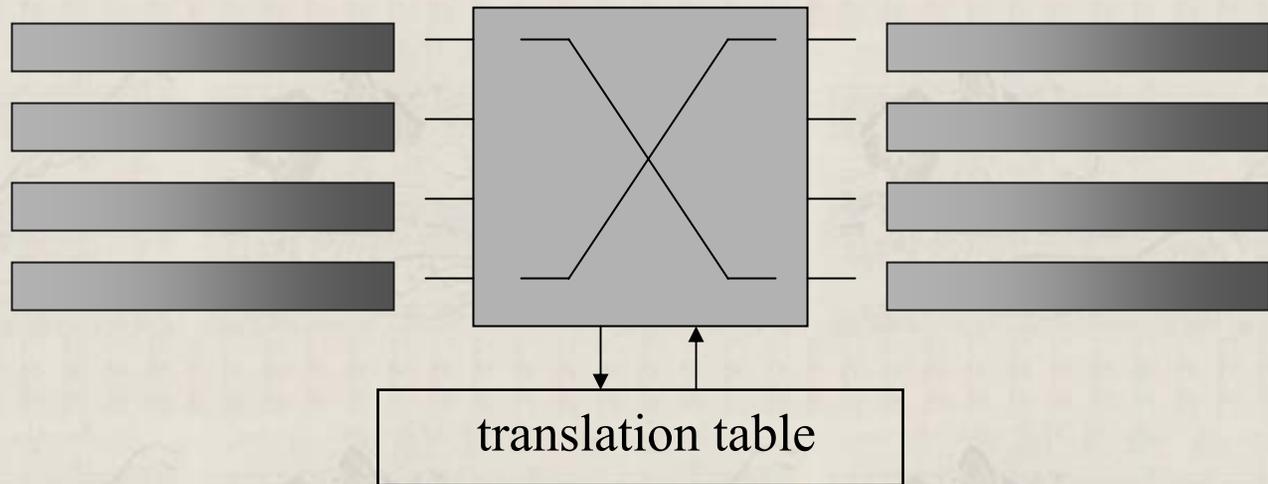
---

- ◆ Error resilience
- ◆ Bandwidth requirements
  - Constant bit-rate (CBR) vs. variable bit-rate (VBR)
  - Symmetrical vs. asymmetrical
- ◆ Quality-of-service (QoS)
  - Delay, delay jitter
  - Packet loss
  - Bit-error rate
  - Burst-error rate, burst error length...
- ◆ Real-time constraints
- ◆ Synchronization: video, audio, data, applications
- ◆ Cost

# Circuit-Switched Network

## ◆ Principle

- Several connections are time-multiplexed over one link
- A dedicated circuit is established during the complete duration of the connection



# Circuit-Switched Network

---

## ◆ Features

- Constant bit-rate, e.g. 64 kbps PCM channel
- Short transmission delay
- Small delay jitters

## ◆ Examples

- Public Switched Telephone Network (PSTN)
  - Plain Old Telephone Service (POTS)
- Integrated Service Digital Network (ISDN)
  - Narrowband-ISDN (N-ISDN)

# Circuit-Switched Network

---

- ◆ Suitable for real-time applications that requires constant bandwidth
  - Audio streaming
  - CBR compressed video (conferencing)
- ◆ Not efficient for bursty applications
  - Data: file transfer, fax, email, telnet, web browsing...
  - VBR compressed video

# Packet-Switched Network

## ◆ Principles

- Communication links are shared by multiple users
- Information encapsulated in *packets*
- Data packet

- Header



- Packet length, packet number
    - Source and destination routing information (IP addresses)
    - Synchronization, transmission protocol

- Payload

- Packet body containing data to be transmitted

- Trailer or footer

- cyclic redundancy check: parity checking on the payload

- Connectionless

- Can have re-transmission request

# Packet-Switched Network

---

## ◆ Features

- Variable length packets
- Large transmission delay
- Large delay jitters

## ◆ Examples

- Local Area Network (LAN)
  - Ethernet: IEEE 802.3
  - Token Ring: IEEE 802.5 (from IBM)
- Wide Area Network (WAN)

# Packet-Switched Network

---

- ◆ Suitable for applications which require dynamic bandwidth
  - Data
  - VBR compressed video
- ◆ Problem with delay-sensitive applications
  - Real-time video and audio communication (video conferencing)

# Circuit- versus Packet-Switching

	<i>Circuit-Switched</i>	<i>Packet-Switched</i>
<b>Dedicated Connection</b>	Yes	No
<b>Call Set-up</b>	Yes	No
<b>Bandwidth</b>	Fixed	Dynamic
<b>Fixed Route</b>	Yes	No
<b>Network Congestion</b>	Set-up time	Anytime
<b>Utilization Charge</b>	Time-based	Packet-based

# Network Examples

---

- ◆ PSTN: up to 56 kbps, ubiquitous, low cost
- ◆ N-ISDN: 128 kbps, widely available, low cost
- ◆ ATM (Asynchronous Transfer Mode):  
broadband cell-switched network, guaranteed QoS, variable bit-rate, priority, not widely available yet
- ◆ Ethernet: packet-switched network, non guaranteed QoS, delay, packet loss, congestion, widely available, low cost
- ◆ Mobile: low bit rates, bit errors, fading...
- ◆ Others: DSL, cable, satellite...

# TCP and UDP

---

- ◆ Transmission Control Protocol (TCP)
  - Acknowledgement is required for every packet
  - Offers reliable in-sequence delivery
  - Long latency
  - Connection-oriented protocol
- ◆ User Datagram Protocol (UDP)
  - No acknowledgement is needed
  - Offers best-effort delivery
  - Simple protocol, connectionless

# RTP

---

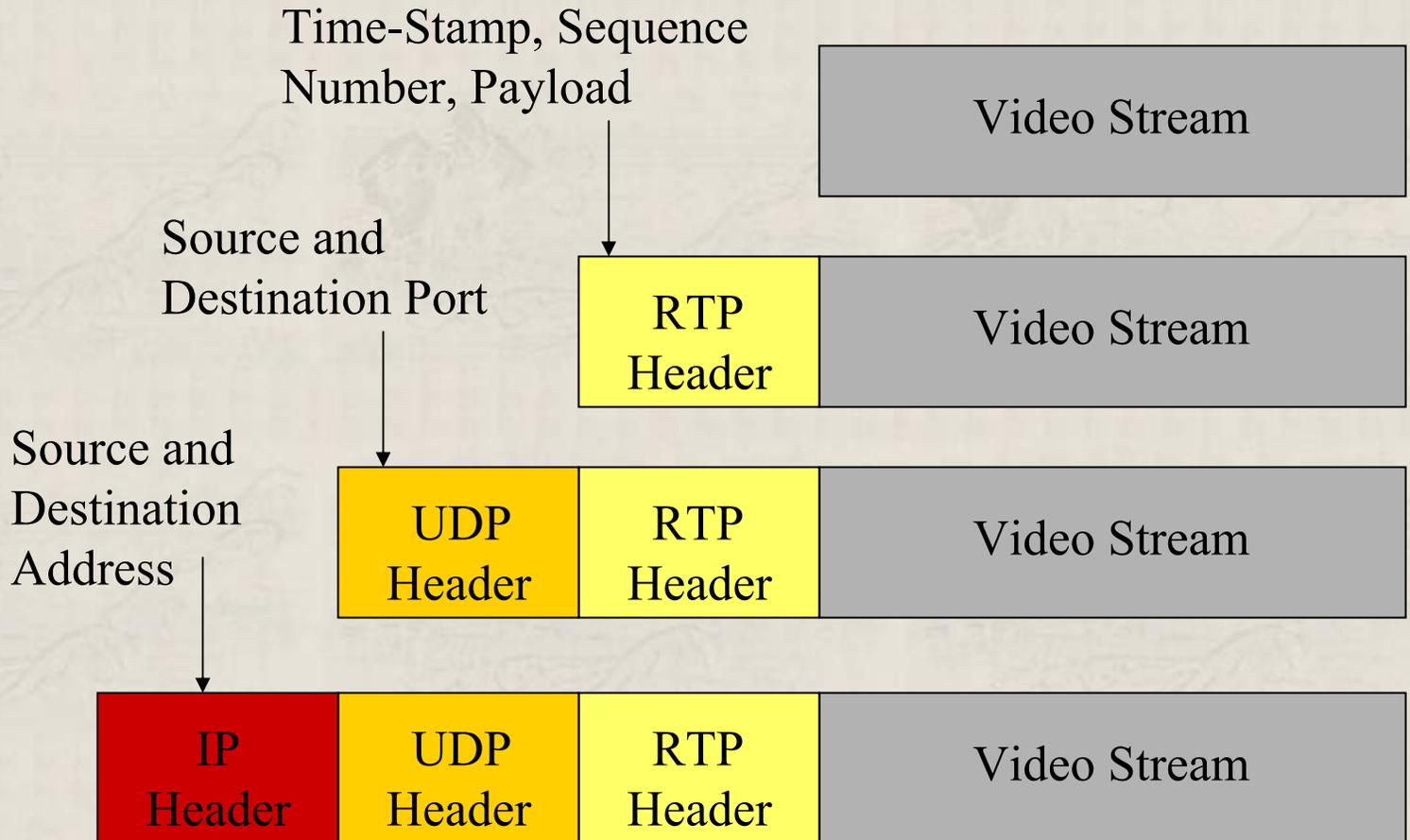
- ◆ Real-time Transport Protocol
  - Provides Time-Stamp to resolve delay jitters
  - Provides sequence number for in-sequence ordering of received packets
  - Provides payload type information
    - H.261, H.263, M-JPEG, MPEG1, MPEG2 video...
    - Payload format adds redundant information to the header to eliminate data dependency between packets

# RTCP

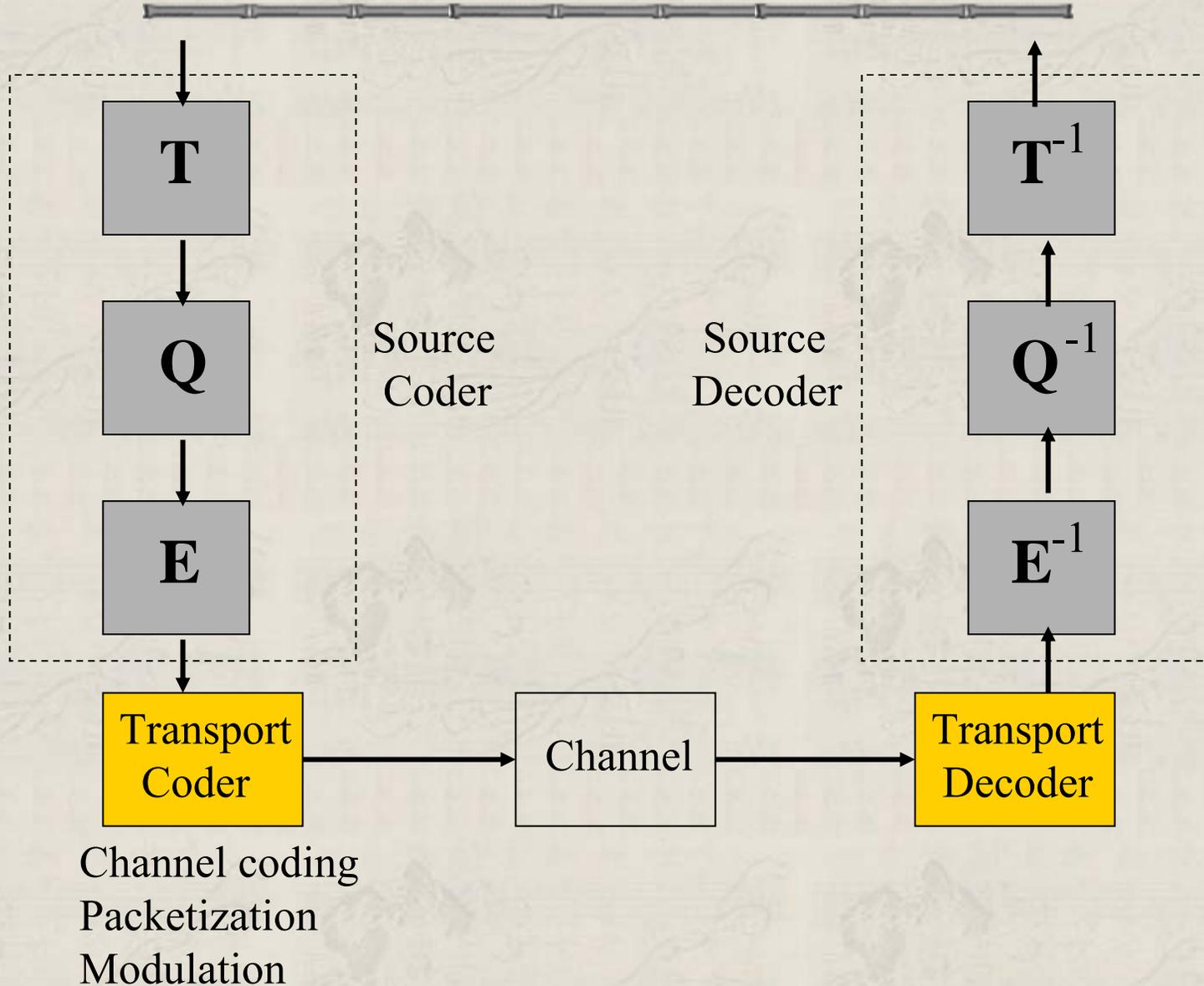
---

- ◆ Real-Time Control Protocol (RTCP)
  - The companion control protocol to RTP
  - Used to monitor the Quality of Service (QoS) and convey information such as name or e-mail to conference participants
  - Sender report and receiver report are used to monitor reception quality, e.g. round-trip delay, packet-loss rate, and inter-arrival jitters.

# Data Encapsulation Example



# Multimedia Communication



# Source vs. Channel Coding

---

- ◆ Source coding
  - Remove redundancy based on source statistics
  - To achieve compression and to reduce bandwidth
- ◆ Channel coding
  - Add redundancy based on channel characteristics
  - Classic example: Hamming error detection and error correction codes
  - To help error detection, recovery, and concealment
- ◆ Joint source and channel coding
  - Active research field

# Error Resilience

---

- ◆ Multimedia over unreliable channels
  - Wireless
  - Internet
- ◆ Transmission errors
  - Random bit error
    - Bit inversion, bit insertion, bit deletion
  - Bursty error
    - Packet loss: packet collision on shared LAN, late arrival (too many hops), buffer overflow in routers, noise in transmission links
    - Bit errors can result in bursty error because of VLC

# Error Control Goal & Techniques

---

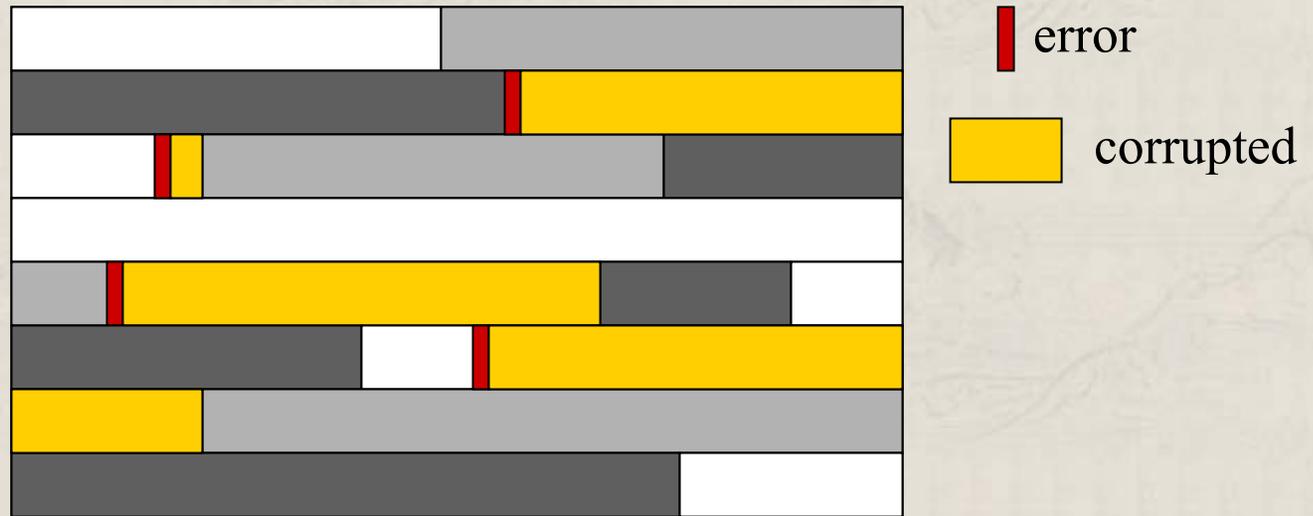
- ◆ Goal:
  - To overcome the effect of errors such as packet loss on a packet network or bit or burst errors on a wireless link.
- ◆ Error control techniques
  - Retransmission
  - Forward Error Correction (FEC)
  - Error concealment
  - Error-resilient coding

# Error Recovery

---

- ◆ Perfect recovery
  - Bit level error detection and correction
  - Forward error correction (FEC), automatic retransmission request (ARQ)
- ◆ Lossy recovery
  - Approximation to the original statistics
  - Post-processing to make error less perceptible to the human visual system (HVS)

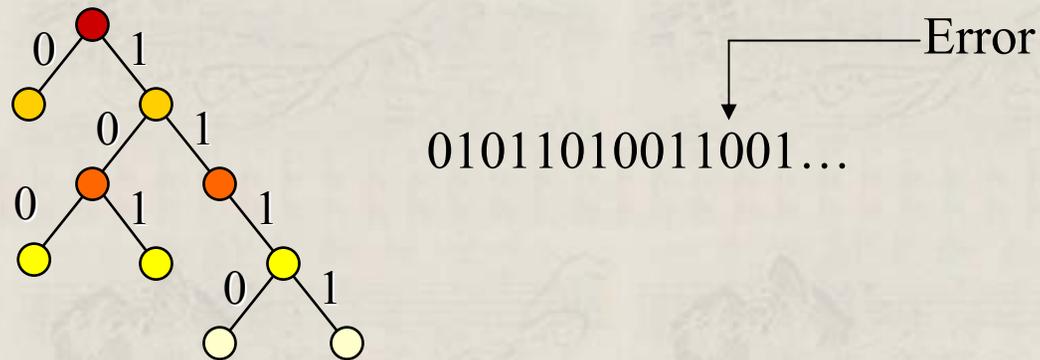
# Error Example



- ◆ GOB/Slice structure
  - Start code (synchronize word) at each slice
  - One error makes the rest of the slice useless
  - Errors do not cross slice boundary
- ◆ Error propagation onto other frames

# Error Detection Methods

- ◆ At the transport codec level
  - Header information: packet sequence number
  - FEC: e.g. in H.261, 18-bit FEC for 493 bits of video
- ◆ At the source codec level
  - Detecting difference of adjacent lines or blocks
  - Syntax mismatch: more than 64 DCT coefficients
  - Non-existing VLC entry



# Retransmission

---

- ◆ Requirement
  - **Feedback-channel** between receiver and sender
- ◆ Approach
  - Receiver tells sender which packets were received/lost and sender resends lost packets
- ◆ Advantages
  - Only resends lost packets, efficiently uses bandwidth
  - Easily adapts to changing channel conditions
- ◆ Disadvantages
  - Extra latency (roughly equal to the round-trip-time (RTT))
  - Not applicable when feedback channel not available (e.g. broadcast, multicast)
  - Effectiveness decreases with increasing RTT

# Retransmission II

---

- ◆ Variations
  - Delay-constrained retransmission
    - Only retransmit packets that can arrive in time
  - Priority-based retransmission
    - Retransmit more important packets before less important ones
    - Different frame types
      - I-frame: Most important
      - P-frame: Medium importance
      - B-frame: Minimum importance (can be discarded)
    - Different layers in a scalable coder
      - Base layer: Most important
      - Enhancement layer 1: Medium importance
      - Enhancement layer 2: Minimum importance

# Forward Error Correction

- ◆ Goal of FEC or channel coding:

- Add specialized redundancy that can be used to recover from errors

K symbols

N-K

- ◆ **Reed-Solomon Code:** RS(N, K) code with s-bit symbols

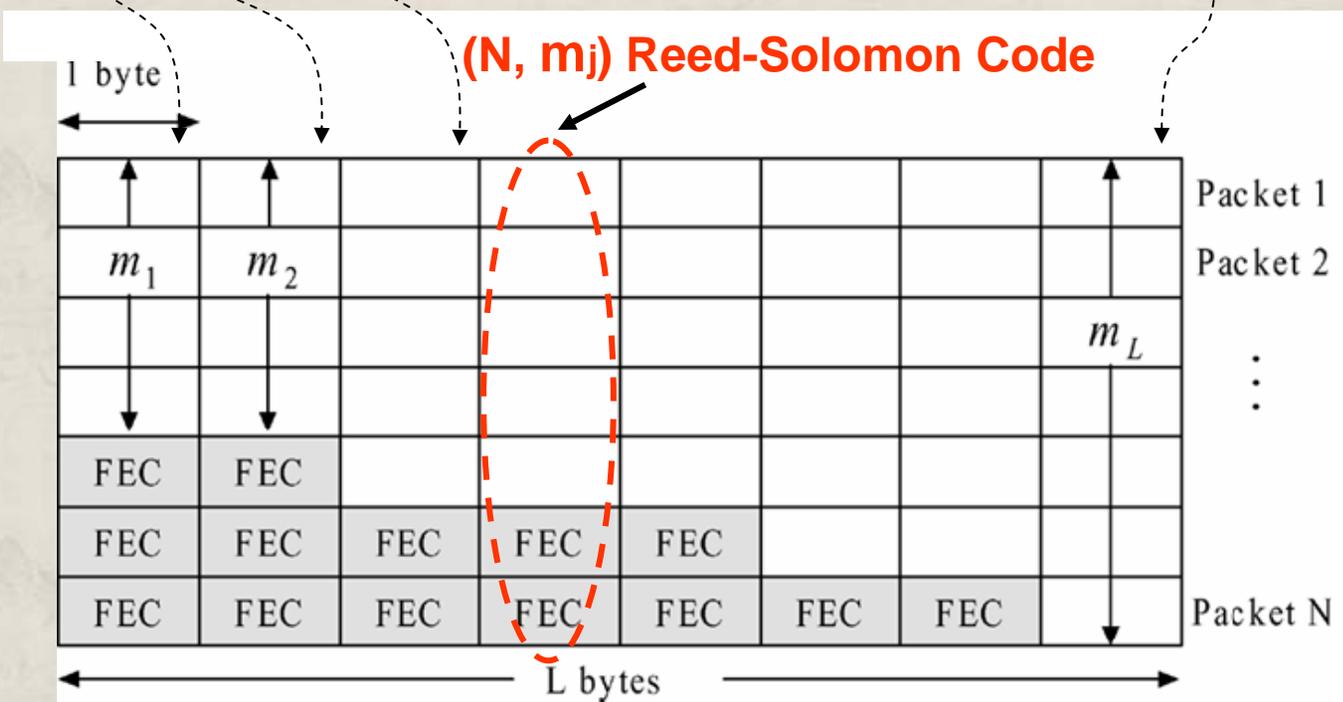
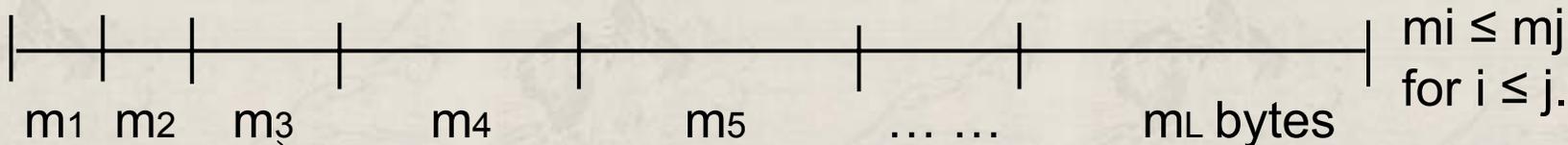
- Invented in 1960 at MIT Lincoln Lab
- Input: K s-bit symbols
- Output: N s-bit symbols (i.e., N-K s-bit parity symbols)
- Error correction capability
  - If error locations are unknown: Up to  $(N-K) / 2$  symbol errors
  - If errors locations are known (**erasure**): Up to (N-K) symbol errors.
  - **One symbol error:** One or more bits of a symbol have errors
- Very suitable for bursty errors: storage (CD, DVD), satellite com.
- Example: RS(255, 233) with 8-bit symbols
  - $N = 255, K = 233$
  - $N - K = 32$
  - Correction capability: up to 16 symbol errors or 32 erasure errors

# Forward Error Correction II

Unequal error protection (UEP):

More (Less) protections for more (less) important data.

Partition of **embedded** bit-stream:



Internet error:  
**Erasure** error!

If any  $m_j$  packets are received, then the first  $j$  parts of the bit-stream can be decoded.

# Forward Error Correction III

---

- ◆ Optimal unequal error protection:
  - Find the optimal bit allocation,  $\{m_j\}$ , such that the expected distortion is minimized.
- ◆ Problems of FEC:
  - Overhead: Loss of compression efficiency
  - Delay:
    - All data have to be available to prepare the N packets
    - The first packet can only be sent after all packets are constructed.

# Error Concealment

---

- ◆ Multimedia communications does not need perfect reception of all data:
  - Different from data communications like ftp.
- ◆ Human visual/audio systems are not sensitive to small amount of errors
- ◆ **Error concealment**
  - Estimate the lost data so as to conceal the fact that an error has occurred.
  - Performed at the decoder: no loss of efficiency
  - General approach: Exploiting the strong spatial/temporal correlation within the data.

# Spatial Error Concealment

- ◆ Spatial interpolation:
  - Estimating the missing pixels by using data of the same frame.
  - Edge-adaptive interpolation



Received image with error

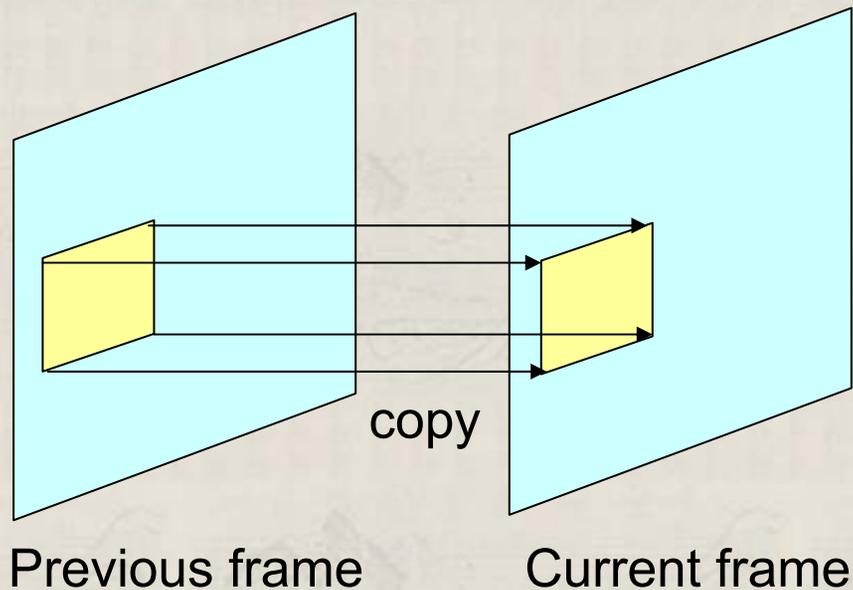


Concealed Image (35.8dB)

# Temporal Error Concealment

## ◆ Temporal interpolation:

- Copy the pixels at the same spatial location in the previous frame (freeze frame)
- Effective when there is no motion, artifacts created when there is motion

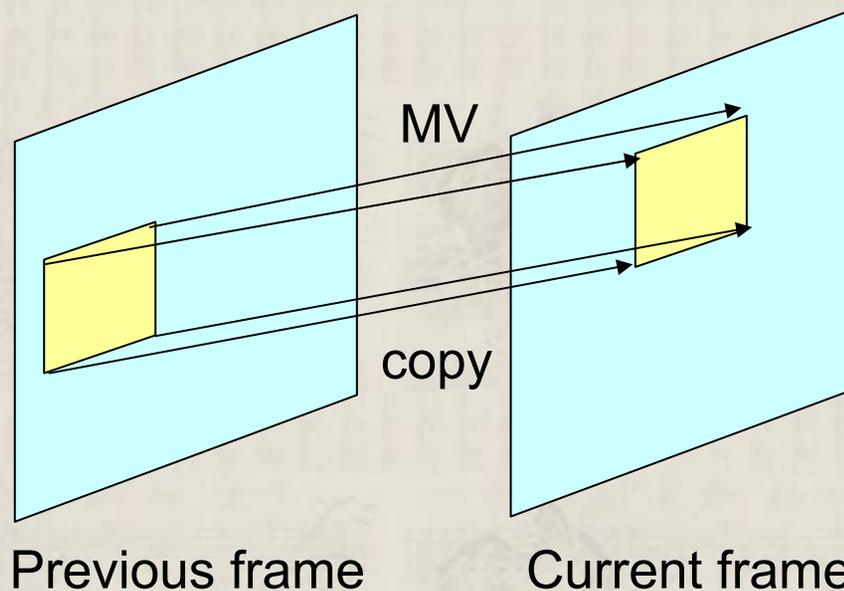


# Temporal Error Concealment II

- ◆ Motion-compensated temporal interpolation:
  - Use motion vector to estimate missing block as motion-compensated block from prior frame
  - Can use coded motion vector, neighboring motion vector, or compute new motion vector

MV1	MV2
MV3	Lost

Estimated MV:  
Median(MV1, MV2, MV3)



# Post-Processing Techniques

---

- ◆ Motion compensated temporal prediction
  - Given motion vector, replace the corrupted MB with the motion compensated block
- ◆ Maximally smooth recovery
  - Exploit block spatial and temporal correlation
  - Does not work well for object boundary
- ◆ Projection onto convex set (POCS)
  - Iterations of two projections: smoothing and replacement
- ◆ Frequency domain interpolation
  - Interpolate DCT coefficients from neighbors
  - Require block interleaving to be effective
- ◆ Recovery of coding modes and motion vectors
  - Interpolate from adjacent blocks, usually from above and below

# Error-Resilient Coding

---

- ◆ Goal
  - Design compression algorithms and compressed bit-streams so that they are resilient to errors
- ◆ Compressed video is highly vulnerable to errors
  - VLC
  - Prediction

# Error-Resilient Video Coding

---

- ◆ Two basic classes of problems:
  - **Loss of synchronization**: Decoder does not know what bits correspond to what parameters
    - e.g. error in Huffman codeword
    - Solutions
      - Isolate corrupted data
      - Enable fast re-synchronization
  - **Error propagation**: Decoder's state is different from encoder's, leading to incorrect predictions and error propagation
    - e.g. error in MC-prediction or DC-coefficient prediction

# Error-Resilient Coding

## ◆ Loss of synchronization:

- Any error in the bit-stream will cause loss of sync.

01100101001000111010101000101011011010100111010101

Discarded

## ◆ Solutions:

### ■ **Insert Resync marker (start code)**

- Marker are distinct from all codewords
- Place resync markers at **strategic locations** in bitstream, e.g. beginning of frame, slice, etc.
- Include information after marker to restart decoding

0110010100100011101010100010101101000000111010101

Discarded

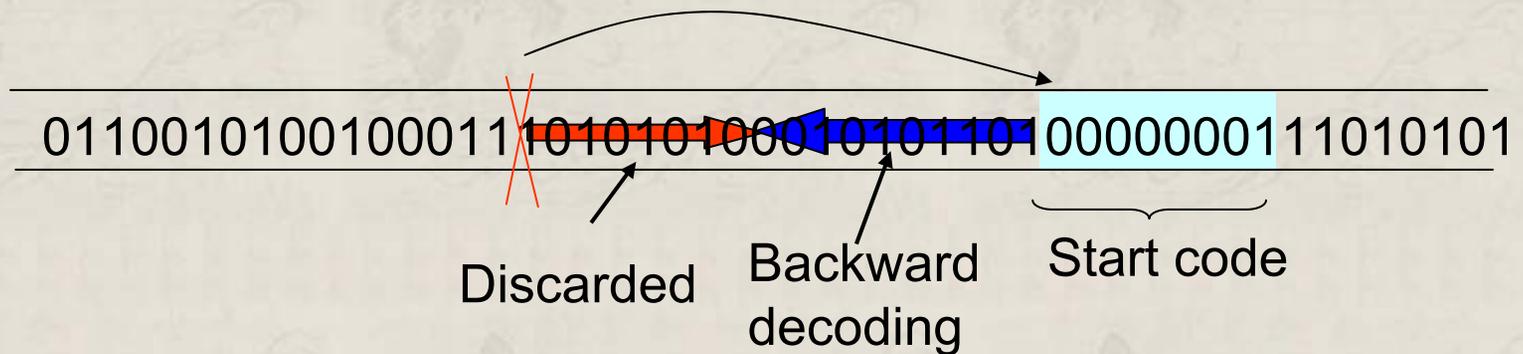
Start code



# Error-Resilient Coding

## ◆ Reversible Variable Length Codes (RVLC)

- Conventional VLC's are uniquely decodable only in forward direction
- RVLC's can also be uniquely decoded in the **backward direction**
- Use: If an error is detected, jump to the next resync marker and start decoding backwards, enabling partial recovery of data (otherwise would be discarded)
- Used in MPEG-4 and AAC audio coding



# Reversible VLC

Example:

	Non-reversible Golomb-Rice Code (m=2)	Reversible Golomb-Rice Code	Reversed Codeword
0	00	00	00
1	01	01	10
2	100	110	011
3	101	111	111
4	1100	1010	0101
5	1101	1011	1101
6	11100	10010	01001
7	11101	10011	11001

↑  
Still uniquely decodable

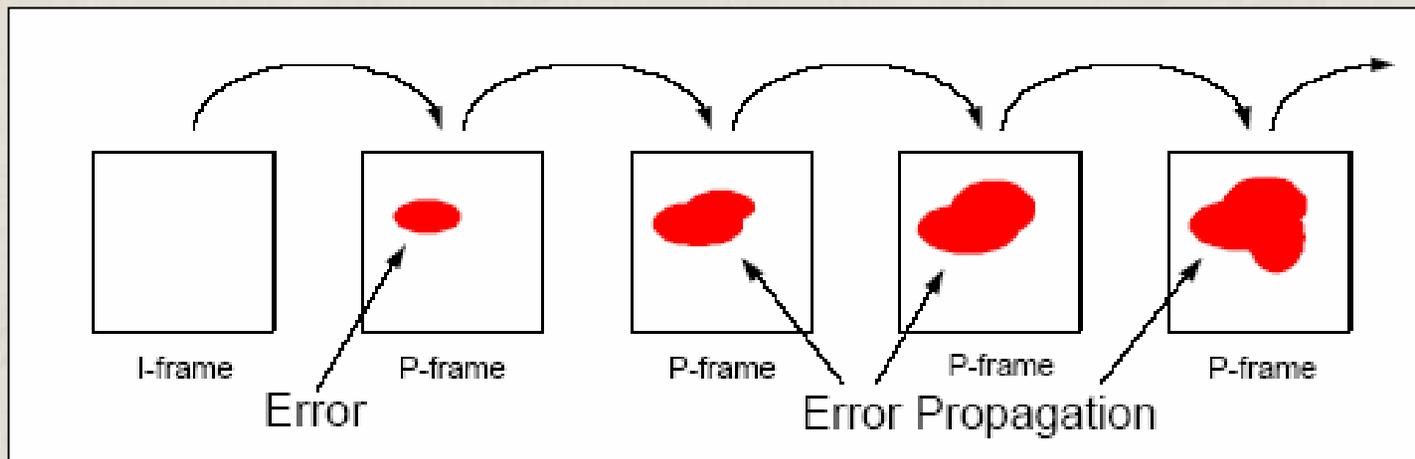
# Data Partitioning

- ◆ Observation: Bits closely following resync are more likely to be accurate than those farther away
- ◆ Idea: Place *most important information* immediately after resync (MV's, shape info, DC coeffs), and less important info later (AC coeffs)
- ◆ Contrasts with conventional approach where data is interleaved on a MB by MB basis



# Error Propagation Problem

- ◆ Decoder's state is different from encoder's, leading to incorrect predictions and error propagation
- ◆ E.g. error in MC-prediction or DC-coefficient prediction



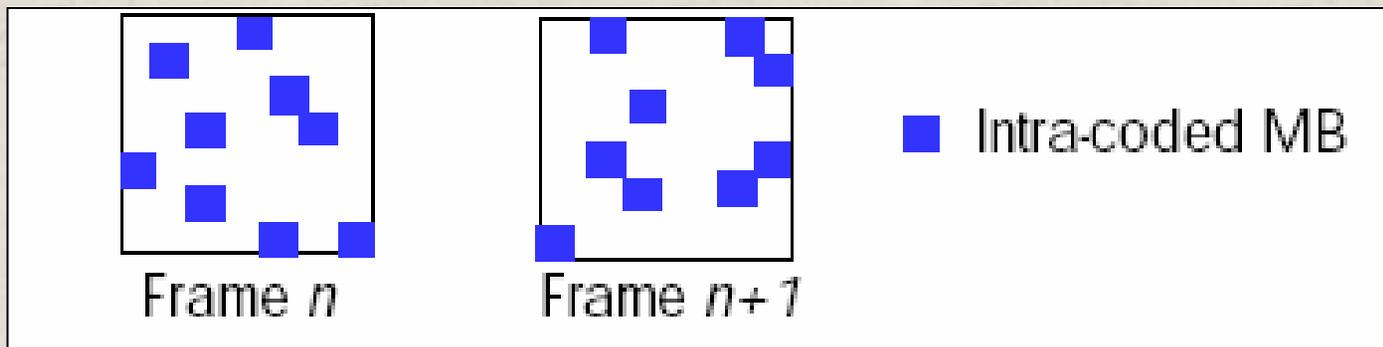
# Limit Error Propagation

## ◆ Periodic I-frames

- Example: I-frame every 15 frames
- - Limits error propagation to one GOP

## ◆ Partial intra-coding of each frame

- Partial: Individual macro-blocks (MBs) are intra-coding
- *Periodic intra-coding* of all MBs
- – A fraction of the MBs in each frame are intra-coded in some predefined order; after N frames all MBs are intra-coded



# Feedback Channel

---

- ◆ When feedback channel is available, *decoder detects error and can tell the encoder*:
  - **Reinitialize prediction (use I-frame)**
    - Simple, straightforward, dynamic (compared to fixed GOP)
    - However, requires higher bit rate for intra coding
  - **Which frame to use as reference for next prediction**
    - Encoder & decoder store multiple previously coded frames
    - Encoder chooses which previously coded frame to use as reference for prediction (e.g. only use correctly received frames)
  - Need a reliable feedback channel with short round trip time.

# Other Techniques

## ◆ Scalable Video Coding

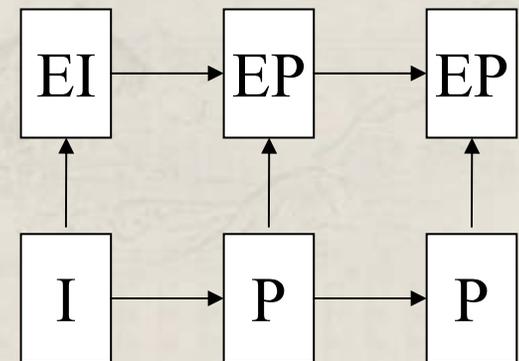
- Codes video into a *base layer* and one or more *enhancement layers*
- – Examples: Temporal, spatial, SNR (quality) scalability
- – *Prioritizes the video data*
- – *Different priorities can be exploited to enable reliable video delivery, e.g. unequal error protection, prioritized retransmission*

## ◆ However, *Internet is best-effort*

- Does not support QoS
- All packets are *equally likely to be lost*

## ◆ Furthermore, base layer is critical

- Other layers are useless if base layer is lost

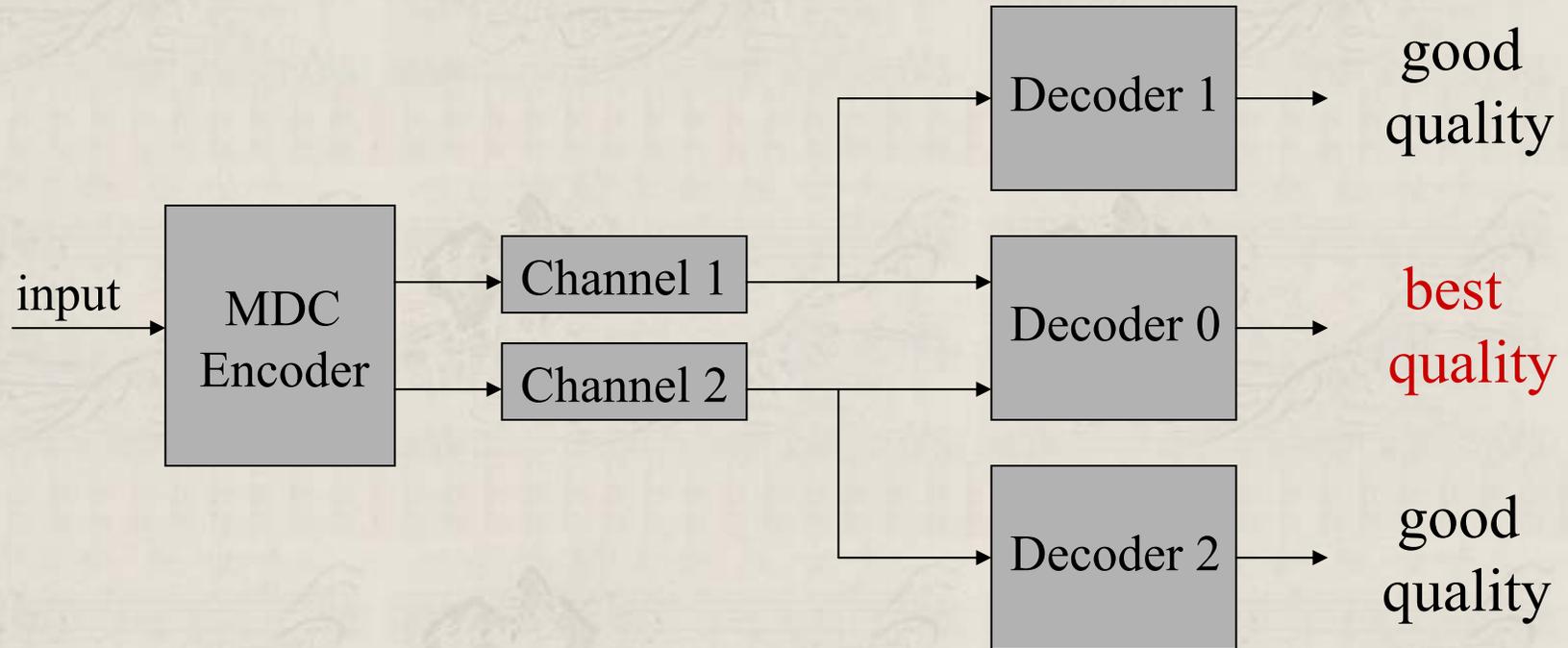


# Layer Coding

---

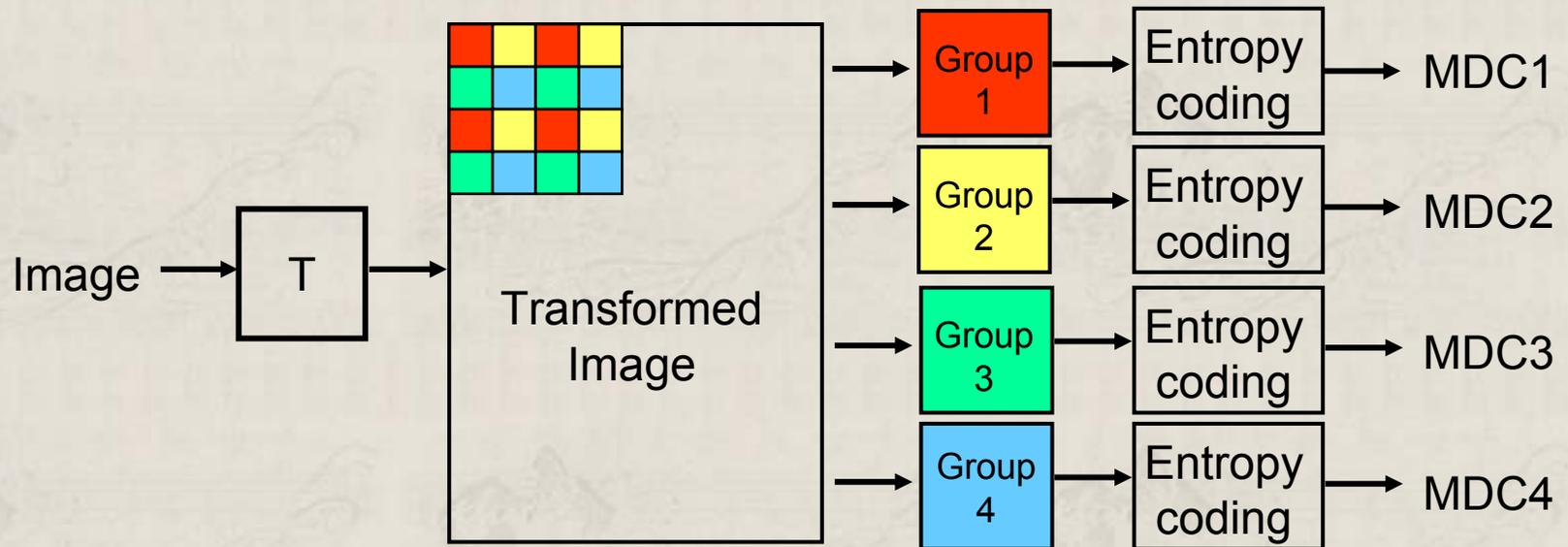
- ◆ Transport prioritization
  - Low priority cells may be dropped
  - Prioritized transmission power, e.g. wireless
  - Prioritized error protection
- ◆ Frequency domain partitioning
- ◆ Successive amplitude refinement
  - SNR scalability
- ◆ Spatial/Temporal resolution refinement
  - Spatial scalability; temporal scalability
- ◆ Coding modes and motion vectors are essential, hence should belong in the base layer

# Multiple Description Coding



- ◆ Parallel channels with similar and independent statistics
- ◆ Signal can be recovered from any one channel
- ◆ Quality improves with more channels
- ◆ Techniques
  - Spatial/transform domain sub-sampling
  - Nested quantization

# MDC: Interleaving



- ◆ Joint decoder: estimate lost blocks from received blocks
- ◆ This interleaving scheme can be applied directly in the time/spatial domain as well

# MDC: 256x256 Lena @ 1.1 bpp



a	b
c	d

Coding gain: 7.1dB

MSE: 0.035

a) 4 descriptions:

33.04 dB

b) 3 descriptions:

31.00 dB

c) 2 descriptions:

29.60 dB

d) 1 description:

26.67 dB

# MDC: 512x512 Barbara @ 1 bpp

---



4 descriptions: 32.55 dB



3 descriptions: 28.92 dB



# MDC: 512x512 Barbara @ 1 bpp

---



2 descriptions: 26.97 dB



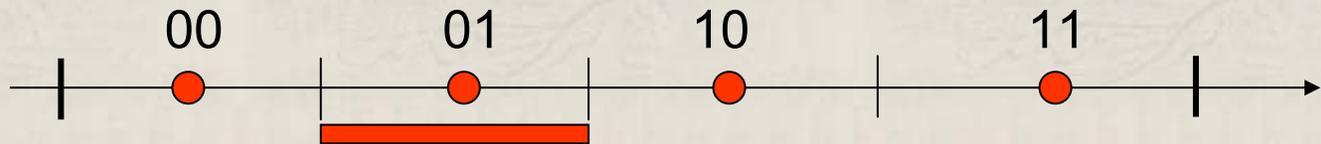
1 descriptions: 23.97 dB



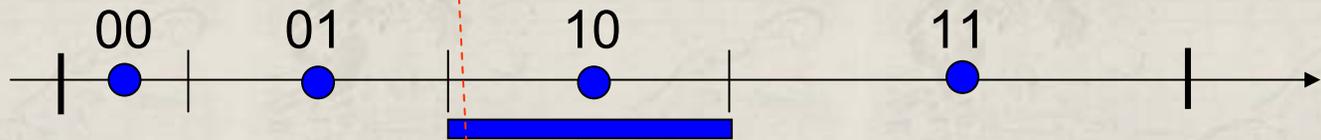
# MDC: Nested Quantization

- ◆ How to generate multiple descriptions:
  - Multiple description scalar quantizer (MDSQ)
    - Two quantizers with straddled bins.

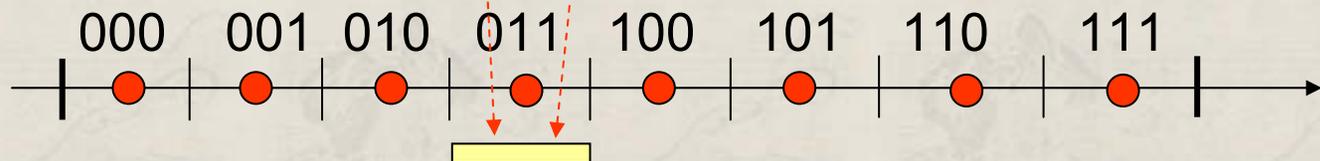
Bins and reconstruction levels of Quantizer 1:



Bins and reconstruction levels of Quantizer 2:



Bins and reconstruction levels of the joint finer quantizer:  
(if both descriptions are received)



# Transport Level Control

---

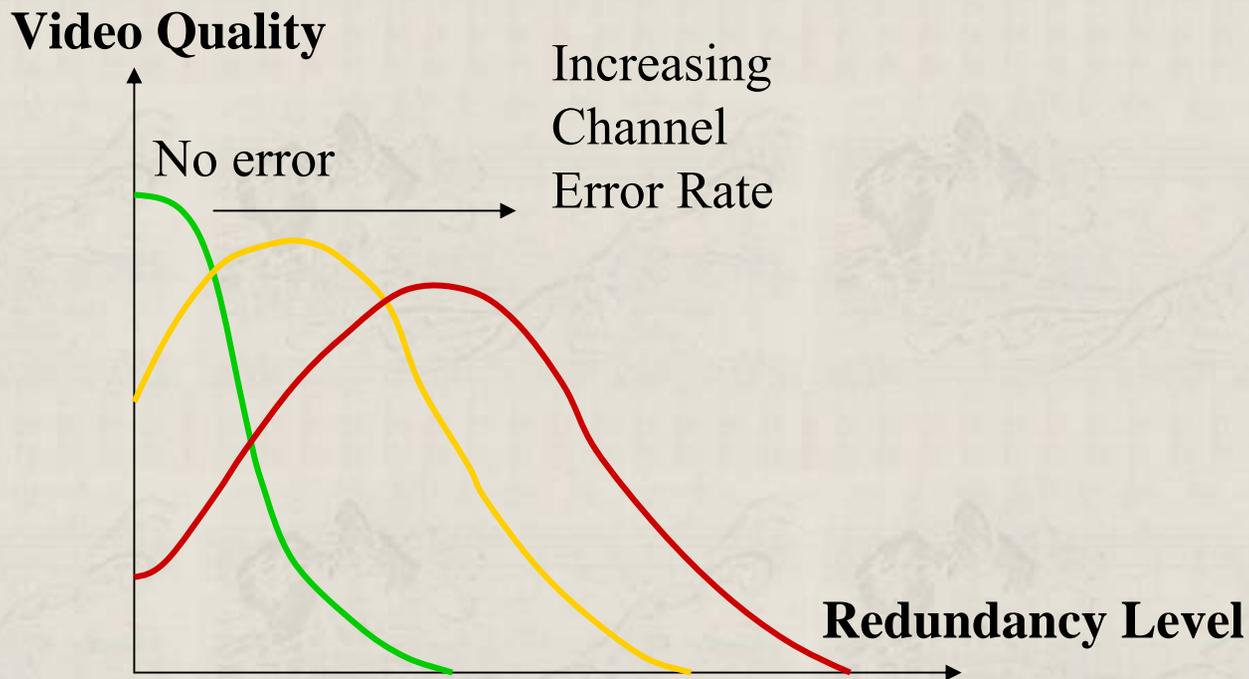
- ◆ Robust packetization
  - Coding modes repeated in successive packets
- ◆ Spatial block interleaving
  - Adjacent blocks are packed into non-successive packets
- ◆ Dual transmission of important information
  - Picture headers, motion vectors, quantization matrix

# Interactive Error Concealment

---

- ◆ Selective encoding
  - Avoid using corrupted regions for prediction
  - H.263: reference picture selection mode
  - When error rate is high, use more intra coding and shorter slices
- ◆ Re-transmission without waiting
  - Keep decoding while a trace of affected pixels is recorded
  - Upon arrival of retransmitted data, correct the affected pixels
  - Can achieve perfect recovery without the associated delay
- ◆ Multi-copy re-transmission
  - For really high error rate

# Error-Resilient Redundancy



- ◆ Fixed compression bit-rate and varying channel error rates

# Summary

---

- ◆ Retransmission
- ◆ Forward Error Correction (FEC)
  - R-S Code
  - Unequal error protection
- ◆ Error concealment
  - Spatial interpolation
  - Temporal interpolation
  - MC temporal interpolation
- ◆ Error-resilient video coding
  - Loss of synchronization: re-sync marker, RVLC, partition
  - Error Propagation: I frames, partial intra, reference selection
  - Others: scalable/layer coding, multiple description coding

# Multimedia Streaming over Internet & Wireless IP Networks



Trac D. Tran

ECE Department

The Johns Hopkins University

Baltimore, MD 21218



# Courtesy

---

- ◆ Prof. James F. Kurose, University of Massachusetts – Amherst
- ◆ Prof. Keith W. Ross, Polytechnic University
- ◆ Prof. Jie Liang, Simon Fraser University
- ◆ Prof. Bernd Girod, Stanford University
- ◆ Prof. Yao Wang, Polytechnic University
- ◆ Dr. John Apostolopoulos, HP Labs

# Outline

---

- ◆ Multimedia streaming
  - History, motivation, properties, challenges
- ◆ Review of the Internet and networks
- ◆ Congestion and rate control
  - General approaches
  - H.263
  - H.263+ & MPEG4
- ◆ Buffer control
  - Hypothetical reference decoder
- ◆ Wireless multimedia streaming
  - Challenges



# Streaming Media: a Huge Success

---

- ◆ Hundreds of thousands of streaming media servers deployed
- ◆ More than 1 million hours of streaming media content produced per month
- ◆ Hundreds or millions streaming media players
- ◆ RealPlayer
  - Most popular Internet application second only to Internet Explorer  
*[Media Metrix]*
  - More than 400 million unique registered users
  - More than 200,000 new users per day
  - Open source code
- ◆ WindowsMedia Player

# Streaming Media: A Brief History

---



**1992** □

Multicast Backbone: MBone □

RTP version 1

Audio-cast of 23<sup>rd</sup> IETF mtg

**1994** □

Rolling Stones concert on MBone

**1995** □

ITU-T Recommendation H.263

RealAudio launched

**1996** □

Vivo launches Vivo Active

Microsoft announces NetShow

RTSP draft submitted to IETF

**1997** □

RealVideo launched

Microsoft buys VXtreme

Netshow2.0 released

RealSystem5.0 released

RealNetworks IPO

**1998** □

RealNetworks buys Vivo

Apple announces QuickTime Streaming

RealSystem G2 introduced

PacketVideo founded

**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*

WindowsMedia7

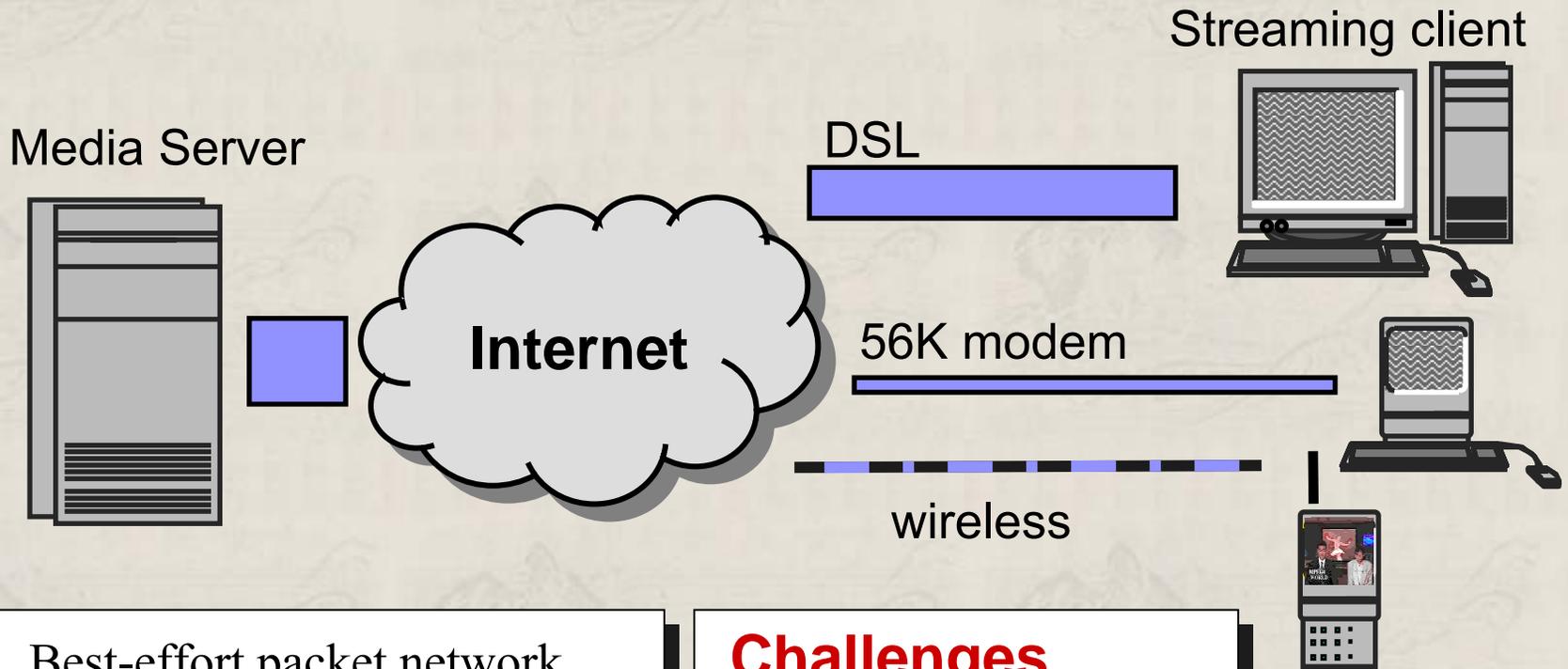
RealSystem8

**2006**

Cingular Wireless provides on-demand streaming video services

WindowsMedia11 (codenamed Polaris)

# Internet Media Streaming



Best-effort packet network

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

## Challenges

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

# Multimedia Communications Applications

---

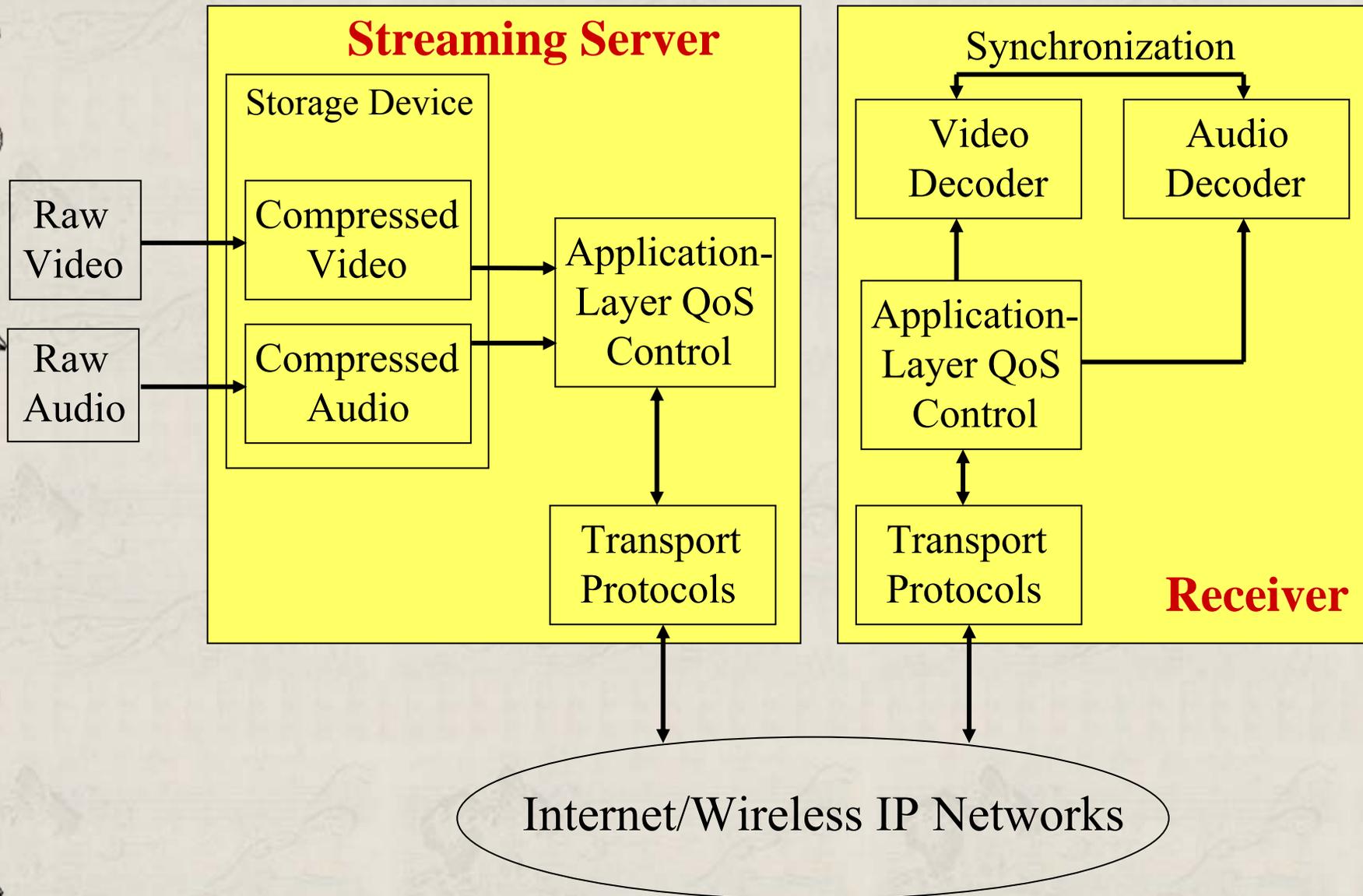
## Classes of applications

- ◆ Streaming stored audio and video
- ◆ Streaming live audio and video
- ◆ Real-time interactive audio and video

## Fundamental characteristics

- ◆ **Delay sensitive**
  - end-to-end delay
  - delay jitter
- ◆ **Loss tolerant:** infrequent losses cause minor glitches
- ◆ Traditional data communications is loss intolerant but delay tolerant

# Architecture for Media Streaming

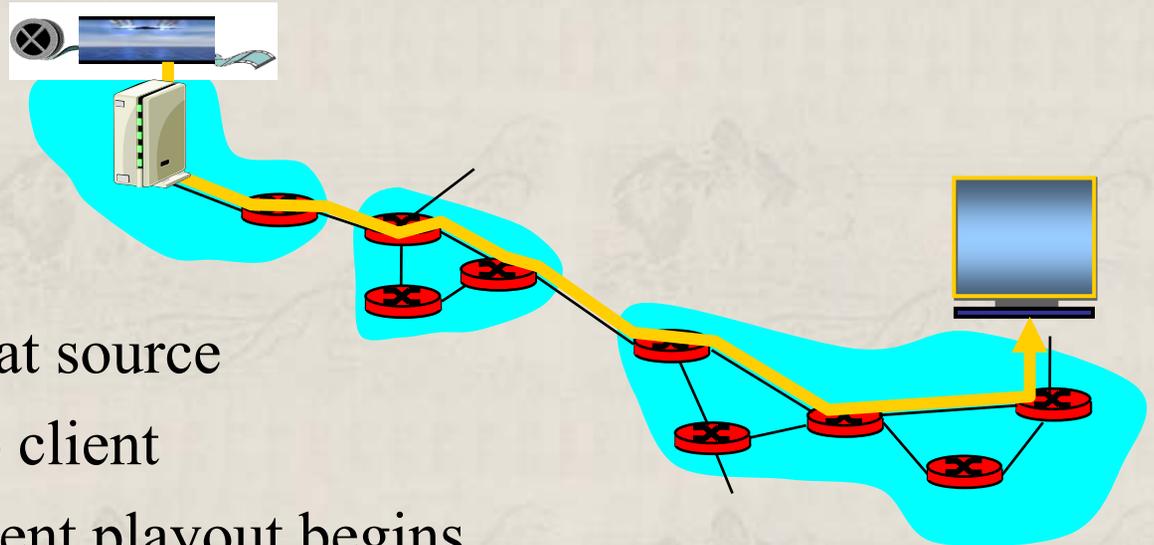


# Media Streaming Components

---

- ◆ Video compression
- ◆ Application-layer QoS control
- ◆ Continuous media distribution services
- ◆ Streaming servers
- ◆ Media synchronization mechanisms
- ◆ Protocols for streaming media
- ◆ Streaming media over wireless IP networks

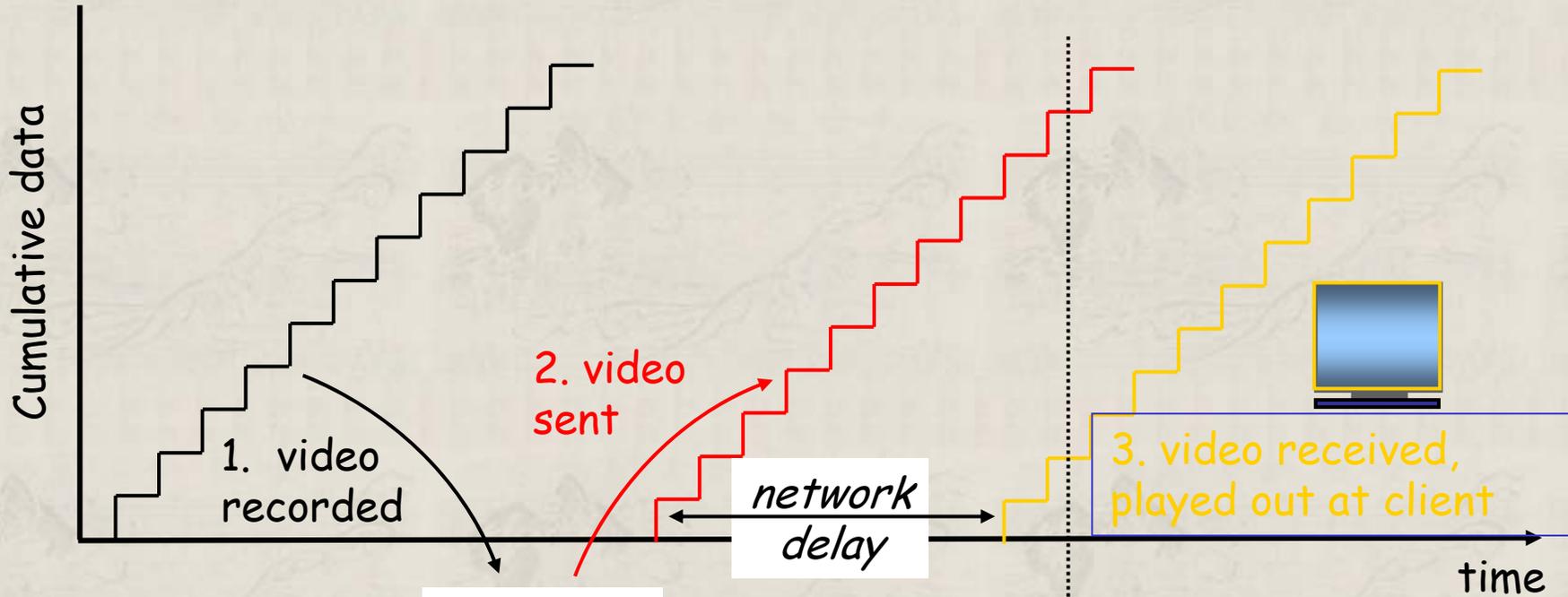
# Streaming Stored Multimedia



## Streaming

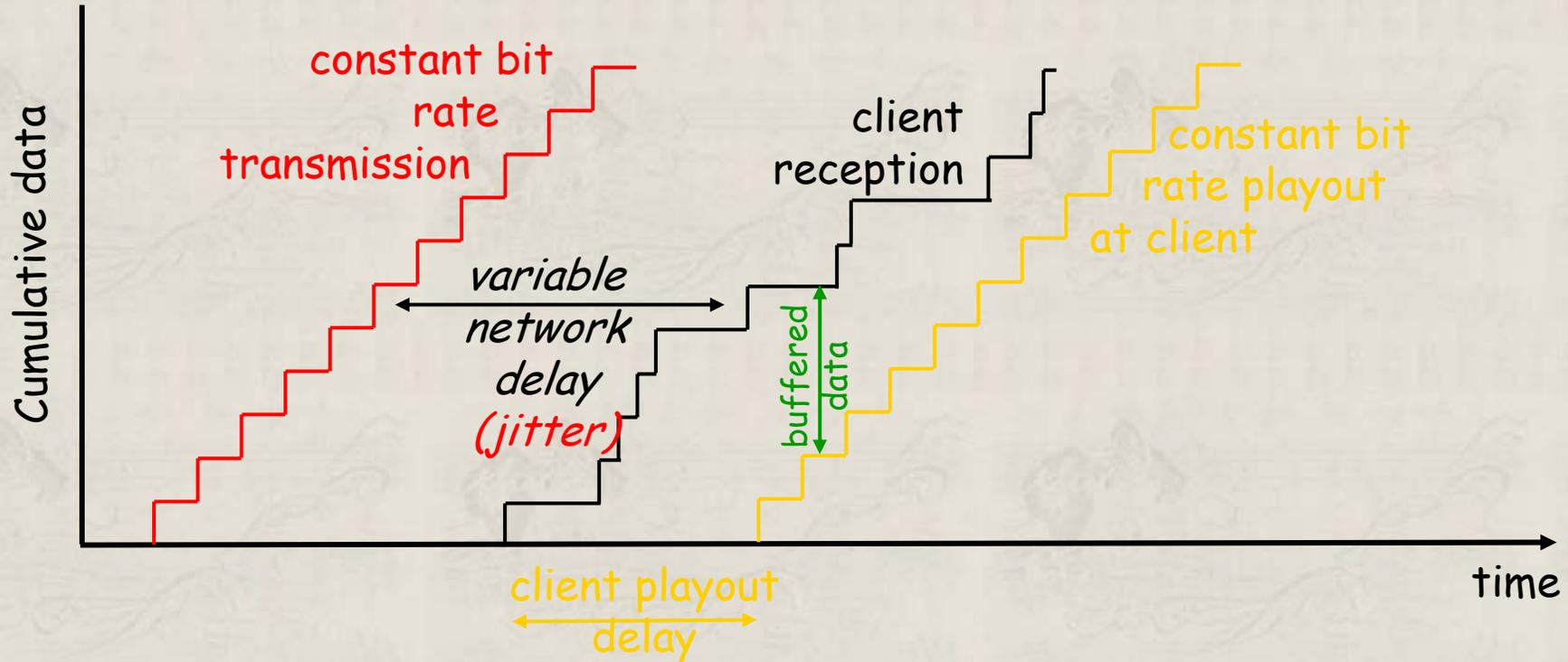
- ◆ media stored at source
- ◆ transmitted to client
- ◆ streaming: client playout begins *before* all data has arrived

# Streaming Stored Multimedia

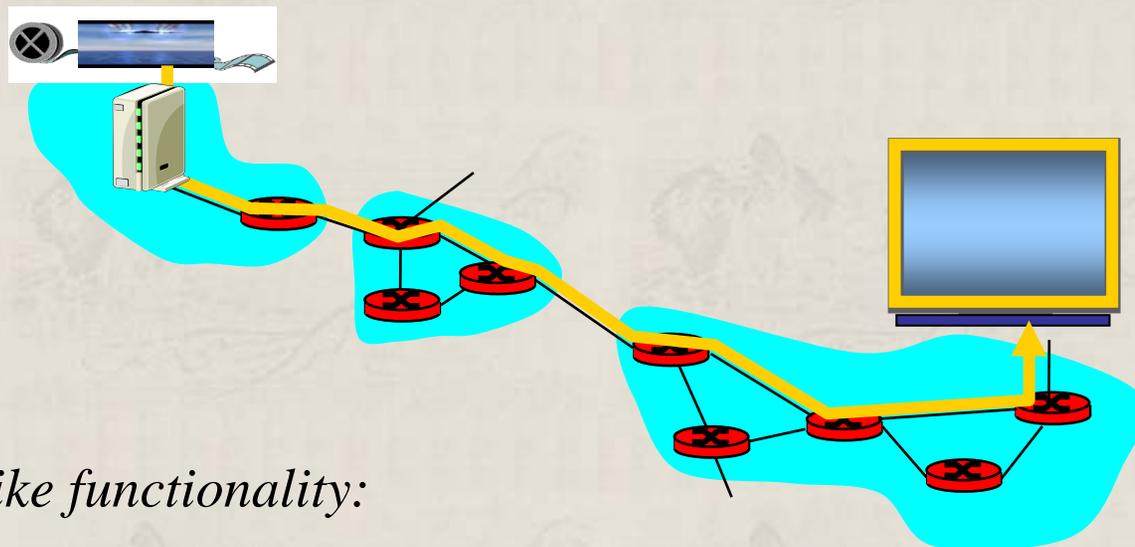


*streaming*: at this time, client playing out early part of video, while server still sending later part of video

# Delay Jitter



# Streaming Multimedia: Interactivity



- ◆ *Need VCR-like functionality:*
  - client can pause, rewind, fast forward, push slider bar...
  - 10 sec initial delay OK
  - 1-2 sec until command effect OK
  - Can be implemented by RTSP protocol (more later)

# Streaming Live Multimedia

---

## Examples

- ◆ Internet radio talk show
- ◆ Live sporting event

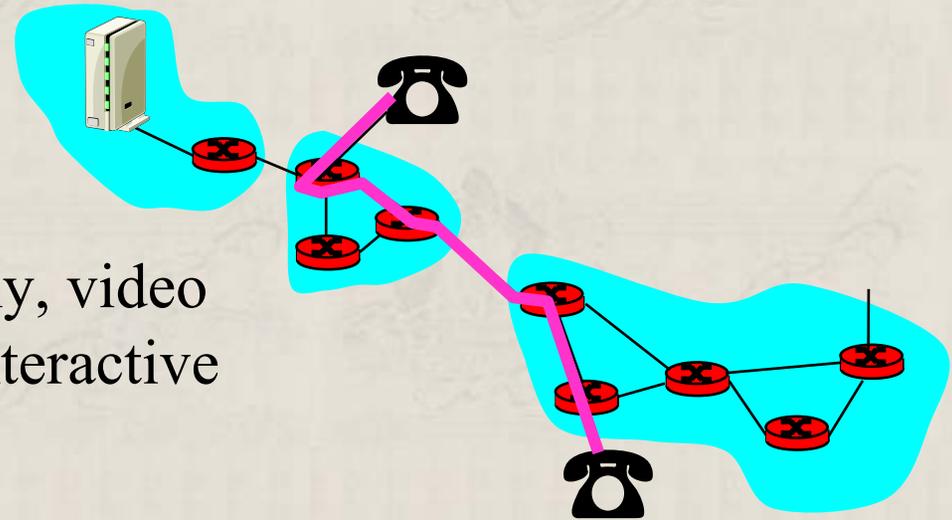
## Streaming

- ◆ playback buffer
- ◆ playback can lag tens of seconds after transmission
- ◆ still have timing constraint

## Interactivity

- ◆ fast forward impossible
- ◆ rewind, pause possible!

# Interactive, Real-Time Multimedia



- ◆ **Applications:** IP telephony, video conference, distributed interactive worlds

- ◆ **End-end delay requirements:**

- audio: < 150 ms good, < 400 ms OK
  - includes application-level (packetization) and network delays
  - higher delays noticeable, impair interactivity



# Challenges of Streaming Media

---

## Bandwidth

- The bandwidth of the Internet is time-varying
- Need rate control algorithms to match the channel rate

## End-to-end delay

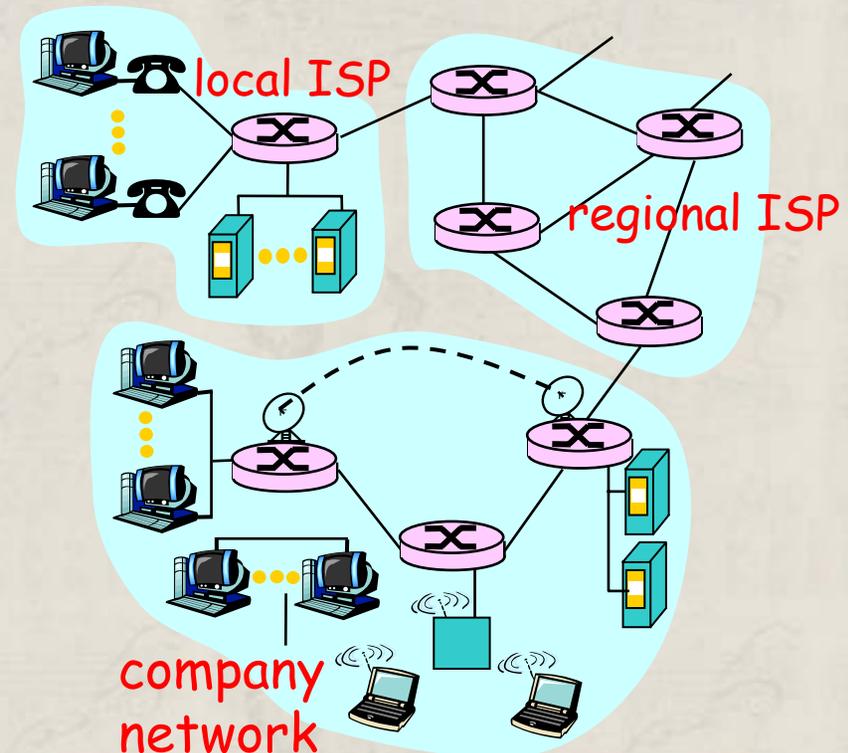
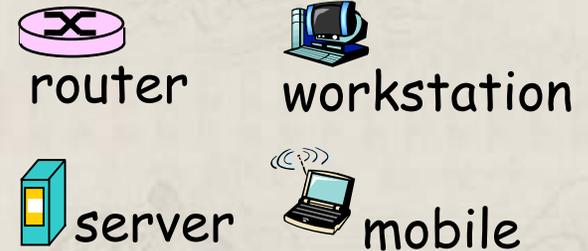
- Need buffer control to deal with delay and delay jitter

## Transmission Loss

- Compressed bitstream is sensitive to transmission loss
- Need error control to recover from the loss

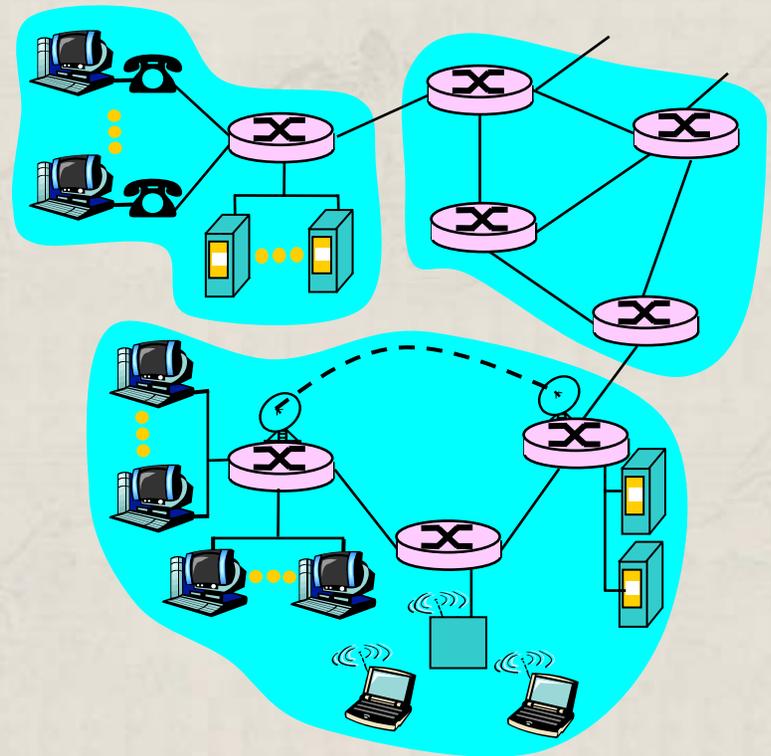
# Internet

- ◆ Millions of connected computing devices
- ◆ *Hardware*
  - Servers, routers, workstations, mobile terminals
  - Routers: forward data packets to their destinations
  - Communication links
    - fiber, copper, radio, satellite
- ◆ *Software*
  - Distributed applications
    - web surfing, streaming ...
  - Protocols
    - Control the sending and receiving of messages



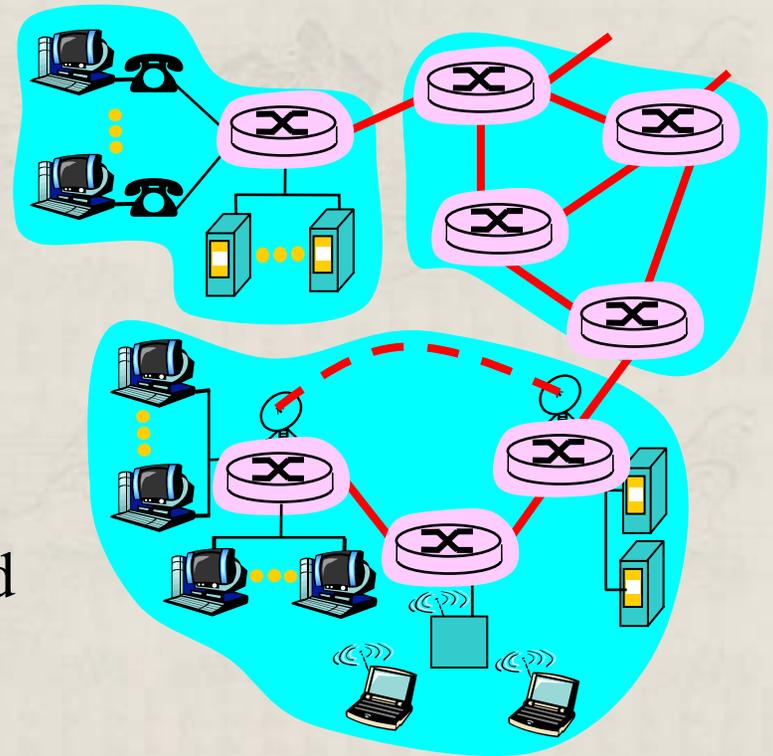
# Network Structure

- ◆ **Network edge**
  - End systems or hosts (clients, servers)
  - run applications such as web browser
- ◆ **Network core**
  - routers
  - network of networks
- ◆ **Access networks, physical media**
  - communication links



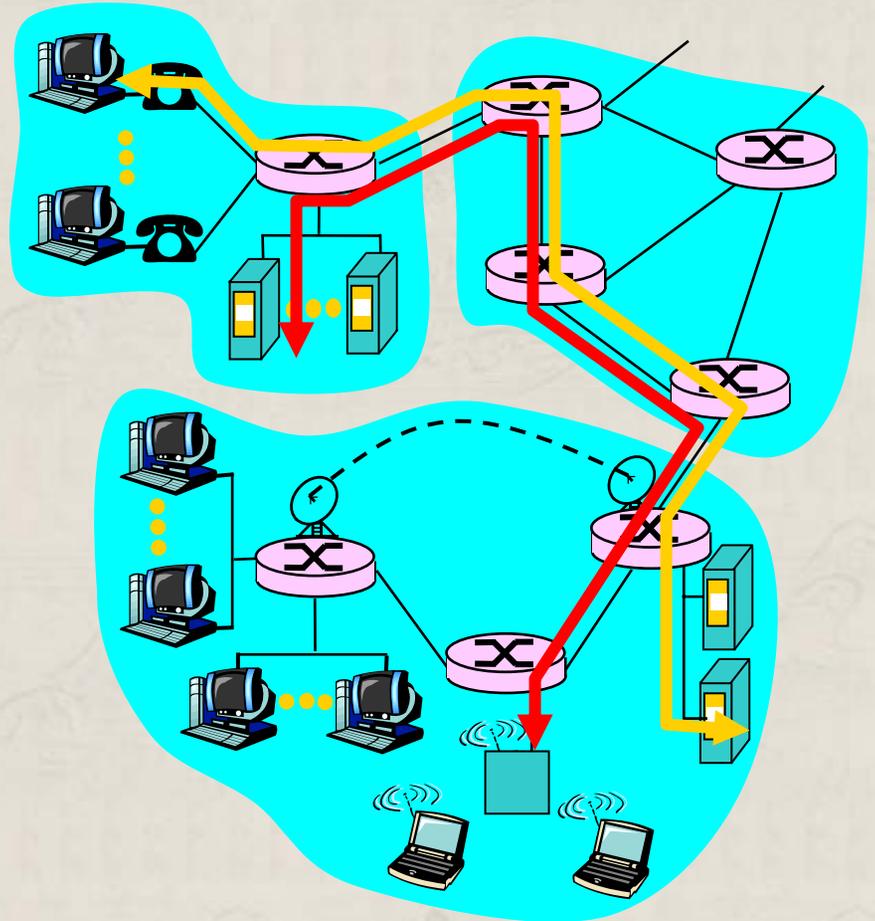
# Network Core

- ◆ Mesh of routers that connect end systems
- ◆ How to build network core?
  - **Circuit switching**
    - Resources are reserved
    - Example: telephone
  - **Packet-switching**
    - Resources are not reserved
    - Example: Internet



# Network Core: Circuit Switching

- ◆ Each link contains many circuits
- ◆ A dedicated circuit is **reserved** during the complete duration of a connection
- ◆ All involved nodes have the same data rates
- ◆ Short delay
- ◆ **Call setup required**
- ◆ Dividing link bandwidth into “pieces”
  - frequency division
  - time division

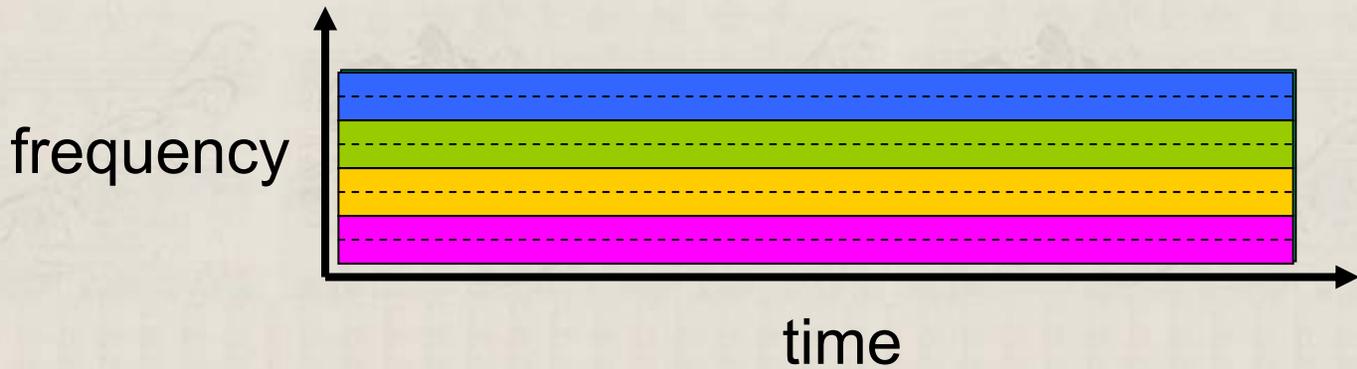


# Circuit Switching: FDM and TDM

FDM: Freq-division multiplexing

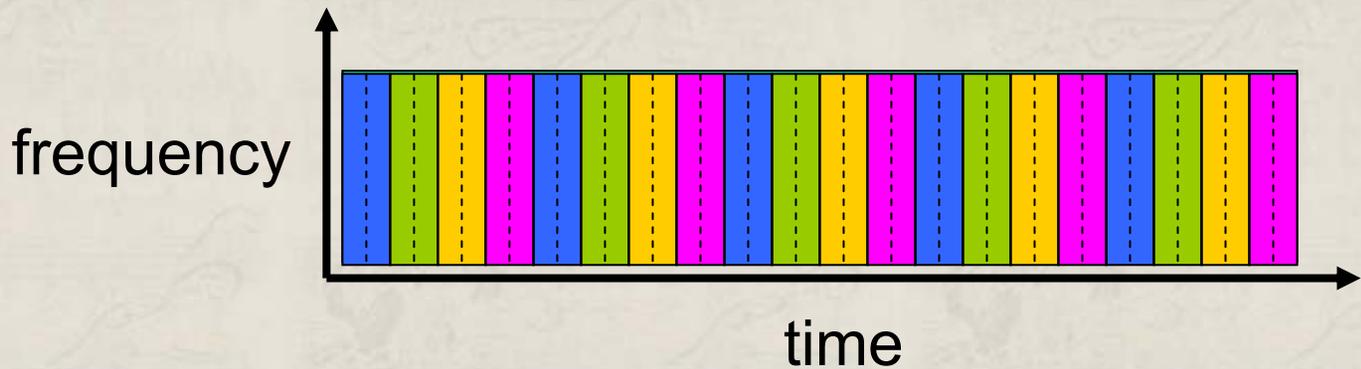
Example:

4 users



4kHz per circuit in telephone network

TDM: Time-division multiplexing



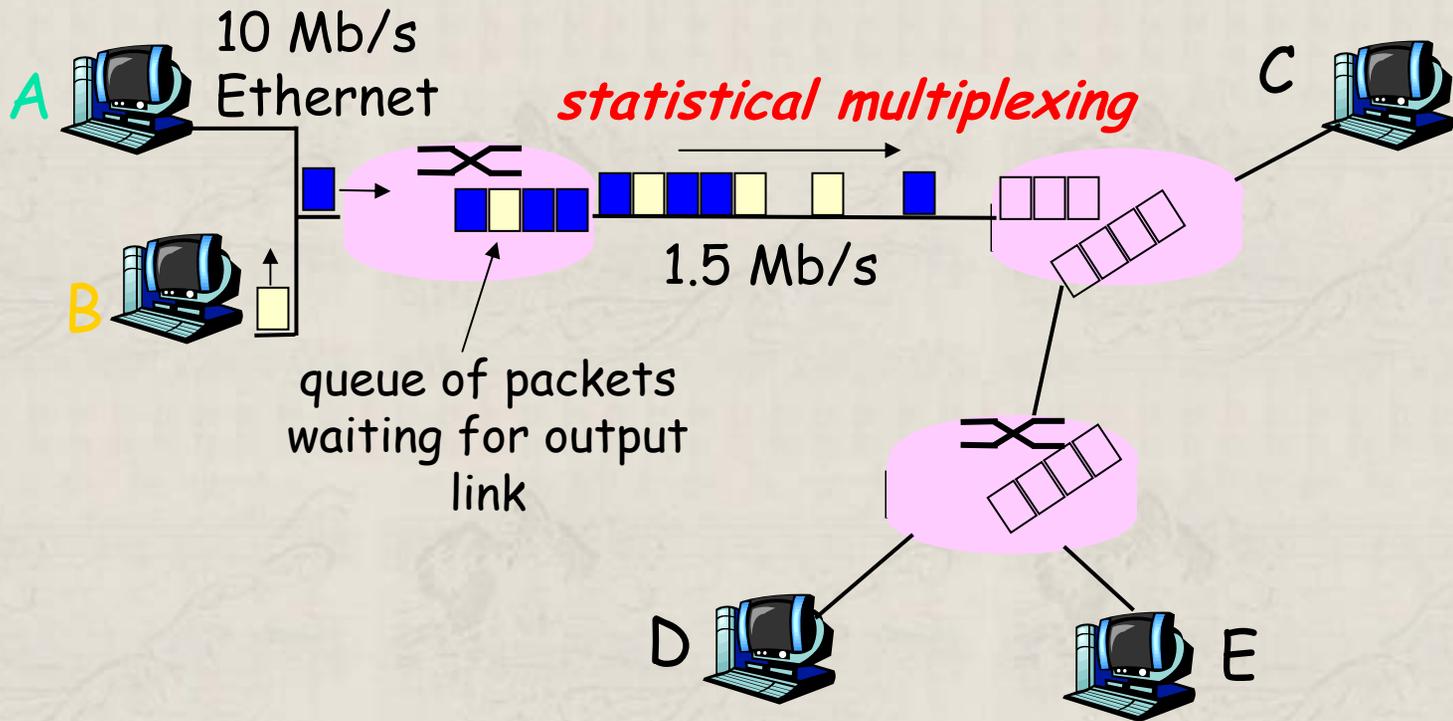
# Network Core: Packet Switching

---

Each end-end data stream divided into *packets*

- ◆ Packets from different users *share* network resources
- ◆ each packet uses full link bandwidth
- ◆ Packet travels through the link and packet switches (routers)
- ◆ Allow more users to use the network
- ◆ Buffers/queues required at routers
  - One output buffer for each link
- ◆ If the link is busy: packets are queued in the buffer
- ◆ Packets are dropped if buffer is full

# Packet Switching

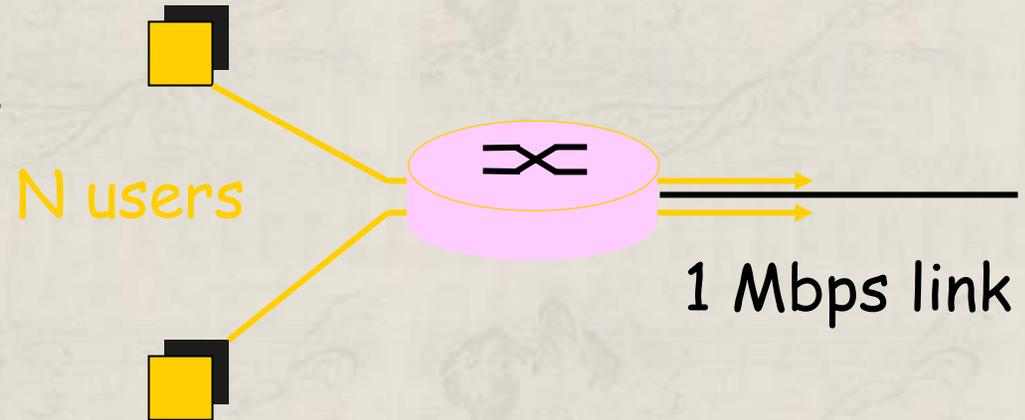


Sequence of A & B packets does not have fixed pattern → *statistical multiplexing*.

# Packet Switching vs Circuit Switching

Packet switching allows more users to use network!

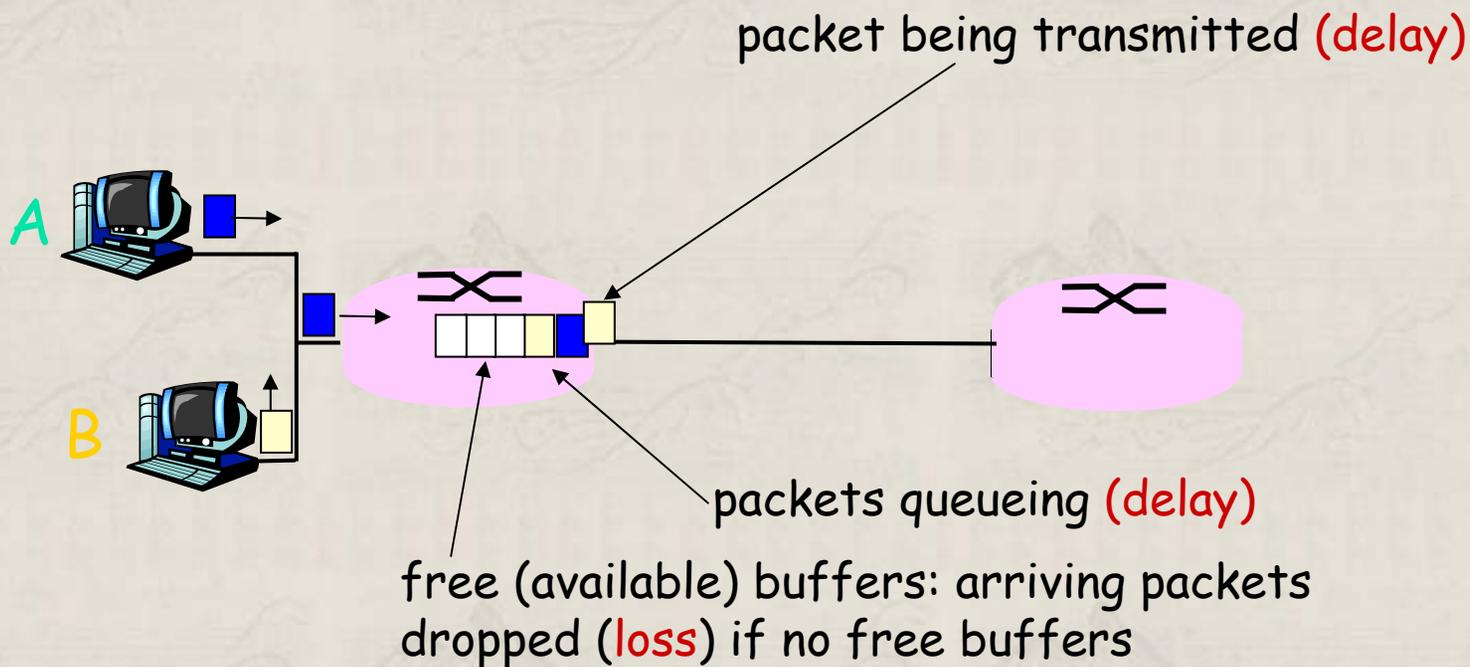
- ◆ 1 Mb/s link
- ◆ Each user:
  - 100 kb/s when “active”
  - active 10% of time
- ◆ Circuit-switching
  - 10 users
- ◆ Packet switching
  - With 35 users, probability of  $> 10$  active users is  $< .0004$
  - Almost as good as circuit switching with only 10 users



# Packet Loss and Delay?

Packets *queue* in router buffers

- ◆ Queuing delay: packets queue, wait for turn
- ◆ Loss: packet arrival rate exceeds output link capacity



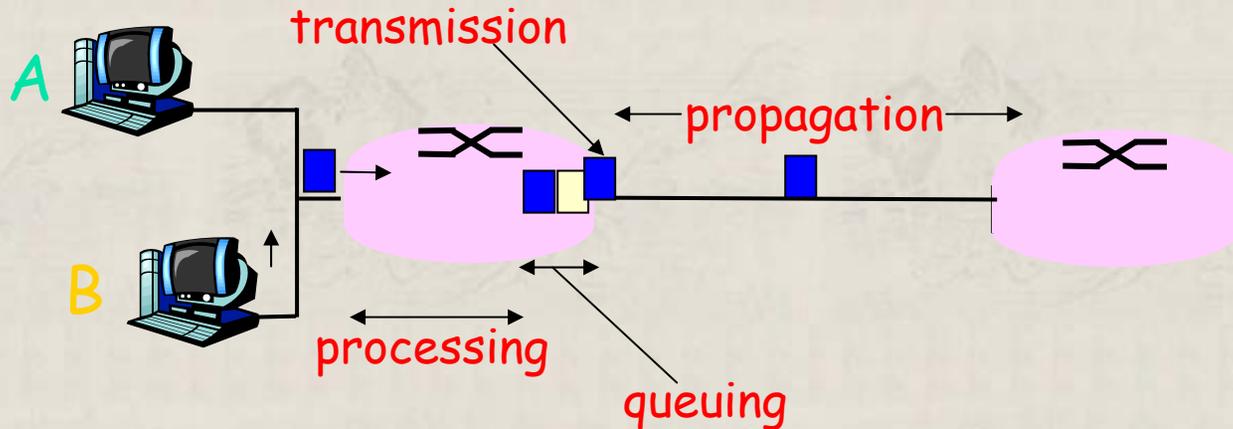
# Sources of Packet Delay

## ◆ Processing delay

- Check bit errors (FEC)
- Examine header
- Determine output link

## ◆ Queuing delay

- time waiting at output link for transmission
- depends on congestion level of router



# Packet Delay

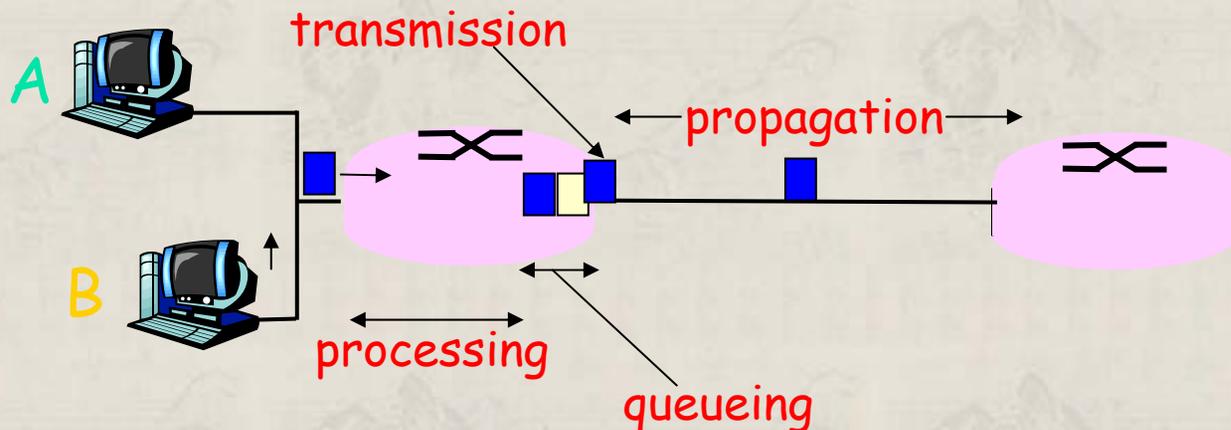
## Transmission delay

### Store-and-forward delay

- ◆  $R$  = link bandwidth (bps)
- ◆  $L$  = packet length (bits)
- ◆ time to send bits into link =  $L/R$

## Propagation delay

- ◆  $d$  = length of physical link
- ◆  $s$  = propagation speed in medium ( $\sim 2 \times 10^8$  m/sec)
- ◆ propagation delay =  $d/s$



# Nodal Delay

$$d_{\text{nodal}} = d_{\text{proc}} + d_{\text{queue}} + d_{\text{trans}} + d_{\text{prop}}$$

- ◆  $d_{\text{proc}}$  = processing delay
  - typically a few microseconds or less
- ◆  $d_{\text{queue}}$  = queuing delay
  - depends on congestion
- ◆  $d_{\text{trans}}$  = transmission delay
  - =  $L/R$ , significant for low-speed links
- ◆  $d_{\text{prop}}$  = propagation delay
  - a few microseconds to hundreds of msecs

# How to Build a Network?

---

- ◆ Layered architecture
  - Divide tasks into different layers
  - Each layer talks to its neighboring layers through a well-defined interface
  - Simplify the design and implementation of protocols
  - Higher layers are logically closer to the user
  - Lower layers are more related to the physical manipulation of the data for transmission

# CBR vs VBR

---

- ◆ Video: Different frames have different amount of complexities and motions
- ◆ CBR video coding: (CBR: Constant bit rate)
  - Spend the same number of bits on each frame
  - → Variable quality (PSNR) from frames to frames
- ◆ VBR video coding: (VBR: variable bit rate)
  - Spend different number of bits on different frames
  - Necessary if constant quality is desired
- ◆ The choice of CBR or VBR coding depends on the channel
- ◆ CBR Channels
  - Example: Telephone network, Digital TV
- ◆ VBR channels
  - Example: Internet, wireless network, DVD

# Video Coding for Storage

---

- ◆ Goal: Store a video in storage media with a total of  $R$  bits
  - Example: DVD, 2 hour movie in 4.7 GB
- ◆ How do we encode the video to meet this storage constraint?
- ◆ Naïve Approach:
  - Allocate equal number of bits to each frame:  $R/N$  bits/frame
  - Problem: Variable quality
- ◆ Multi-pass Approach:
  - Video coding for storage does not require causal processing
  - Can examine the entire sequence and re-encoding
- ◆ Multi-pass coding can provide much better performance than single-pass coding

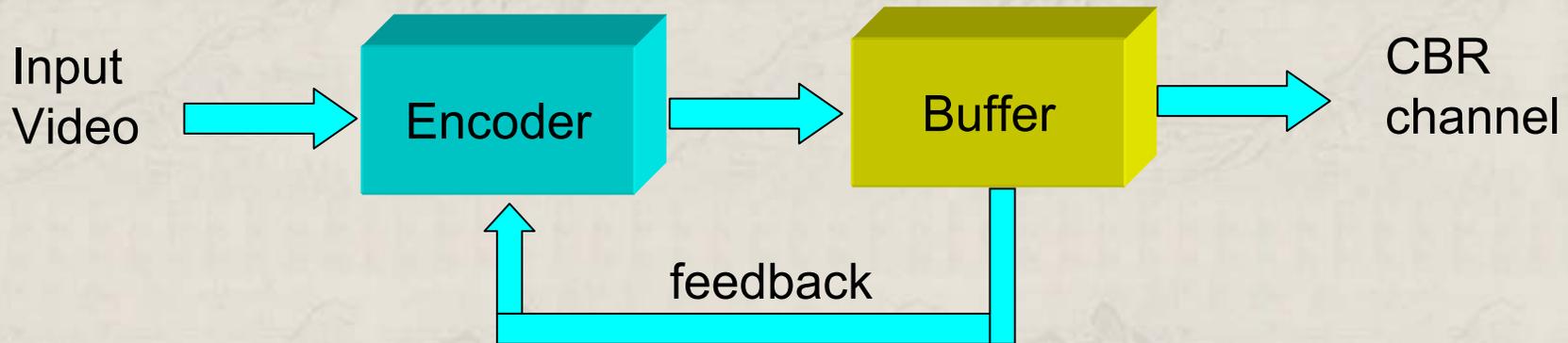
# Multi-Pass Encoding for Storage

---

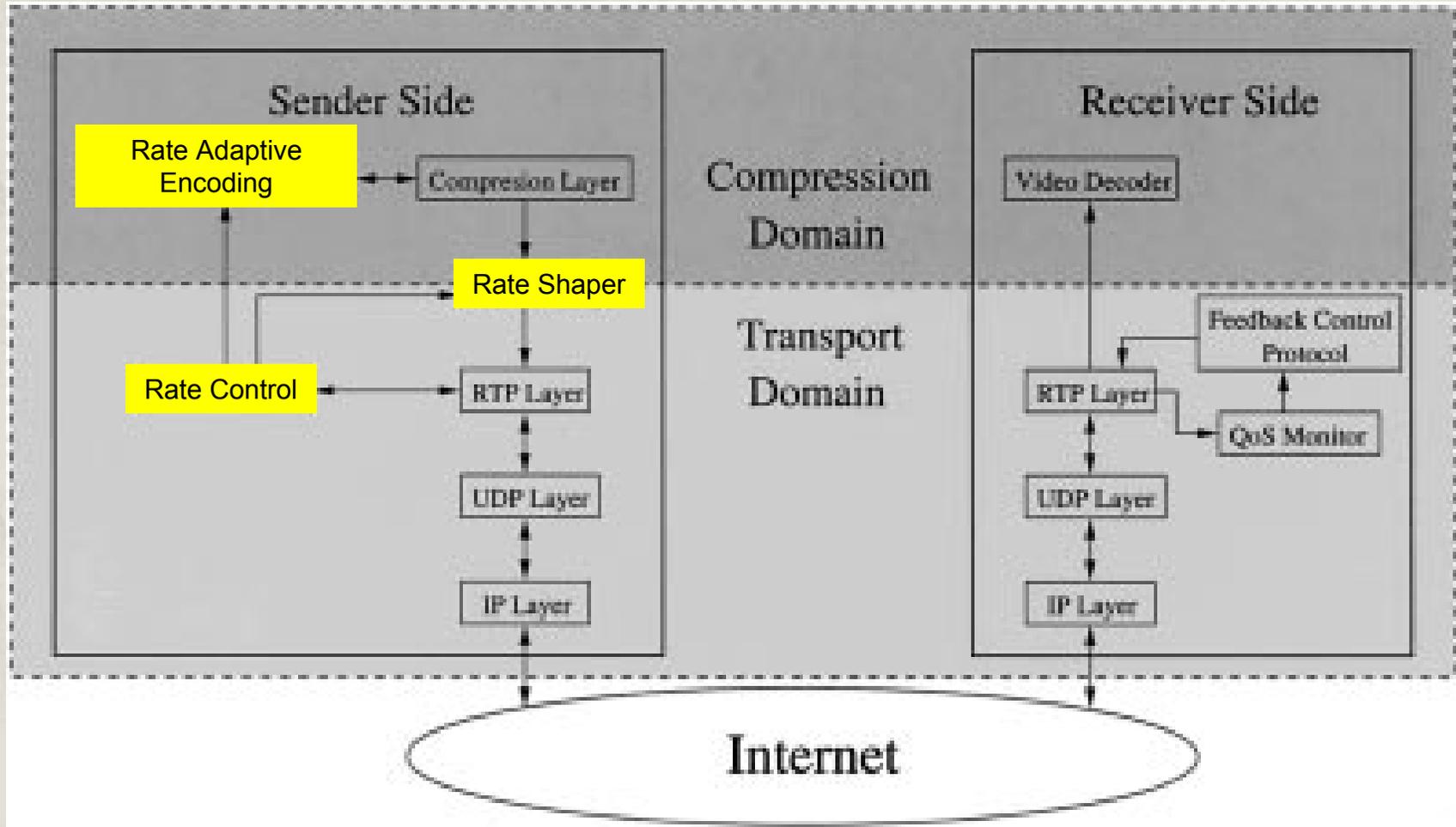
- ◆ Code entire video sequence with VBR coding
- ◆ Gather and analyze statistics
- ◆ If total bits  $> R$  (the allowed max bits)
  - Identify complex portions of video sequence.
  - Re-allocate bits for each frame.
  - Re-encode entire video sequence. Go to 2.

# Video Coding for DTV

- ◆ Digital Television Channel:
  - CBR: 20Mb/s
  - Need buffer to regulate the generated bit rate.
  - Use buffer feedback to adjust quantization:
    - Increase quantization step if buffer level too high
    - Reduce quantization step if buffer level too low



# Streaming Video



Layered architecture for transporting real-time video over Internet. Applied to pre-encoded video by removing the video encoding part.

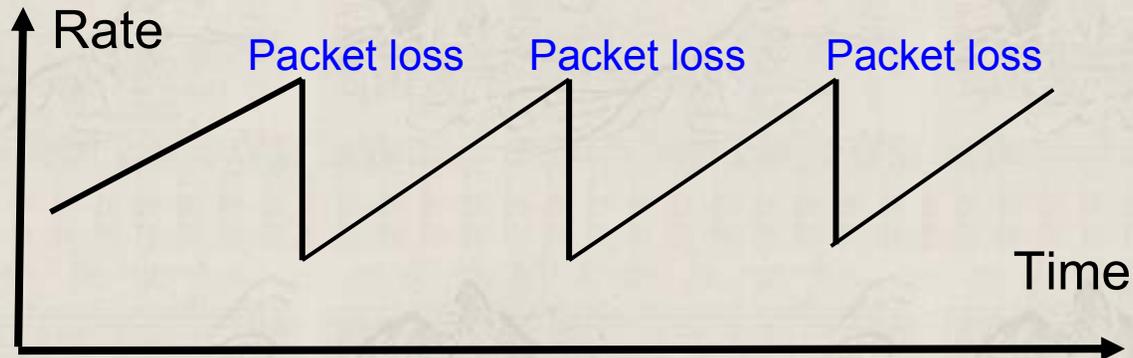
# Congestion Control for Streaming

---

- ◆ Three levels
  - Transport Layer Rate control
    - Match video rate with the available bandwidth
  - Rate control for video encoding
    - Maximize the video quality under a given encoding rate constraint: classical source coding problem
  - Rate shaper:
    - Follows either the source coding approach or the transport approach
    - Example:
      - Server selective frame discard: based on channel information

# Rate Control of TCP?

- ◆ TCP's rate control
  - AIMD: Additive-Increase Multiplicative-Decrease
  - Increase the rate slowly if there is no packet loss
  - Decrease the rate by 50% if there is a packet loss.



TCP's rate is highly fluctuating: saw-tooth pattern

- Exactly match the TCP rate control is not good for streaming media
- TCP cannot be used for most streaming media because of its delay
- UDP is used in most cases
- But UDP does not have any congestion control
- ➔ Need to implement streaming media congestion control at higher layer

# Transport Layer Rate Control

---

- ◆ *Estimate* the available bandwidth
- ◆ *Match video rate* to available bandwidth
- ◆ Rate control may be performed at
  - Sender
  - Receiver
  - Both sides
- ◆ Available bandwidth may be estimated by
  - Probe-based methods
  - Model-based (equation-based) methods

# Source-Based Transport Layer Rate Control

---

- ◆ Source *explicitly adapts* the video rate
- ◆ *Feedback* from the receiver is used to estimate the available bandwidth
  - Feedback information includes packet loss rate
- ◆ Methods for estimating available bandwidth based on packet loss rate
  - Probe-based methods
  - Model-based methods

# Probe-based Methods

---

- ◆ Basic idea
  - *Use probing experiments to estimate the available bandwidth*
- ◆ Example: monitor the packet loss rate  $r$ 
  - If ( $r < threshold$ ), increase transmission rate
  - If ( $r > threshold$ ), decrease transmission rate
- ◆ Simple, ad-hoc
- ◆ Estimate the bandwidth **implicitly**

# Model (Equation)-based Methods

- ◆ Estimate the bandwidth **explicitly**
- ◆ Goal: *Ensure fair competition* with concurrent TCP flows on the network, e.g. fair sharing of bandwidth
- ◆ Basic idea:
  - Model the **average throughput** of a TCP flow instead of the instantaneous throughput
  - Decide the sending rate by the following formula:

$$\lambda = \frac{1.22 \times MTU}{RTT \times \sqrt{\rho}}$$

$\lambda$ : Throughput of TCP.

MTU: Maximum Transmit Unit (packet size)

Default: 576 bytes

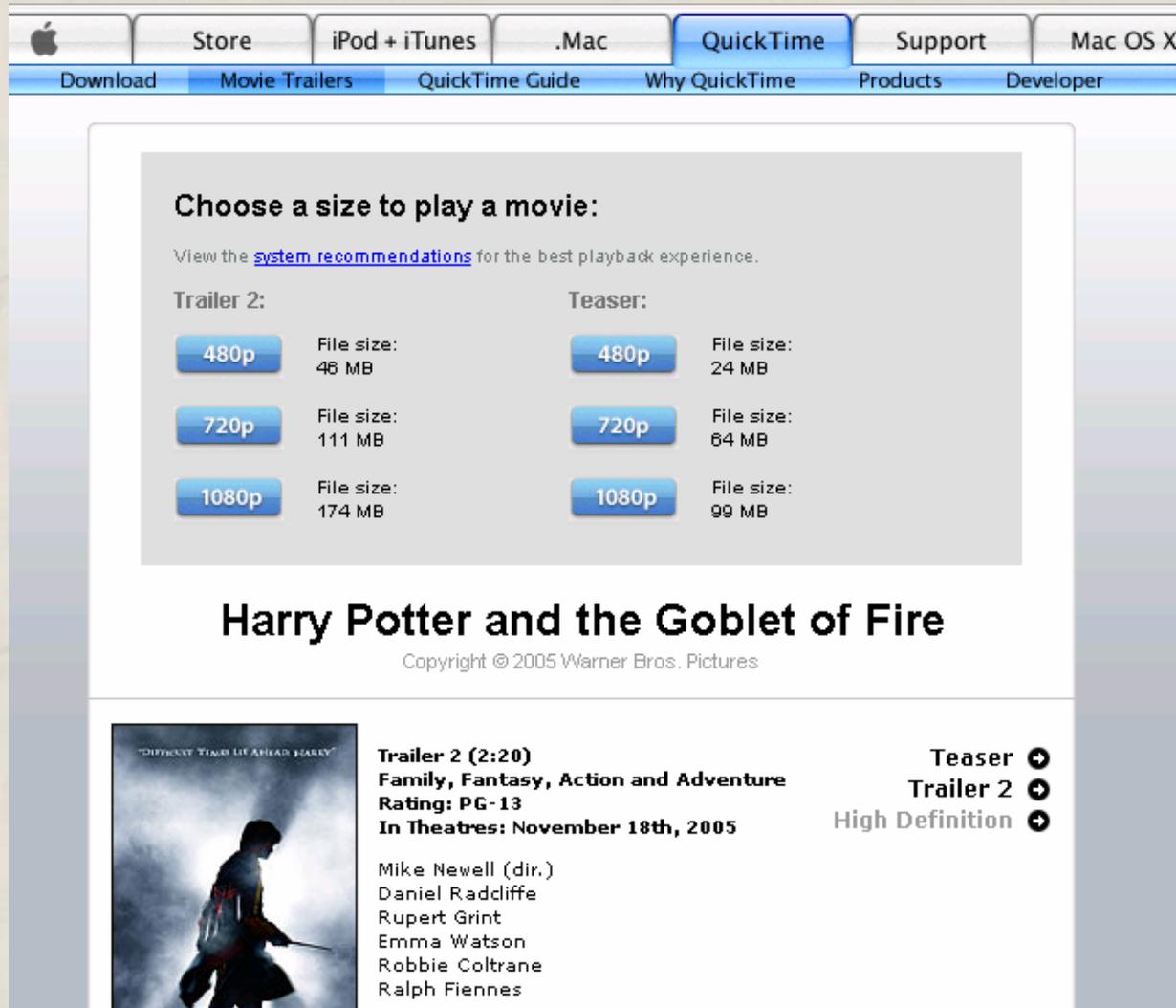
RTT: Round trip time

$\rho$ : Packet loss ratio.

- ◆ Similar characteristics to TCP flow, “fair” to other TCP flows

# Receiver-based Transport Layer Rate Control

- ◆ Receiver selects the video rate from a number of rates



The screenshot shows the Apple QuickTime website interface. At the top, there is a navigation bar with links for Store, iPod + iTunes, .Mac, QuickTime (selected), Support, and Mac OS X. Below this is a secondary navigation bar with links for Download, Movie Trailers (selected), QuickTime Guide, Why QuickTime, Products, and Developer.

The main content area is titled "Choose a size to play a movie:". Below this title, there is a link to "View the [system recommendations](#) for the best playback experience."

The content is organized into two columns: "Trailer 2:" and "Teaser:". Each column has three options for video quality, each with a corresponding file size:

Quality	File size
480p	46 MB
720p	111 MB
1080p	174 MB

Below the quality selection options, the movie title "Harry Potter and the Goblet of Fire" is displayed, along with the copyright information "Copyright © 2005 Warner Bros. Pictures".

At the bottom of the page, there is a section for the movie trailer. On the left is a small image of the movie poster. To the right of the image, the following information is provided:

- Trailer 2 (2:20)**
- Family, Fantasy, Action and Adventure**
- Rating: PG-13**
- In Theatres: November 18th, 2005**
- Mike Newell (dir.)
- Daniel Radcliffe
- Rupert Grint
- Emma Watson
- Robbie Coltrane
- Ralph Fiennes

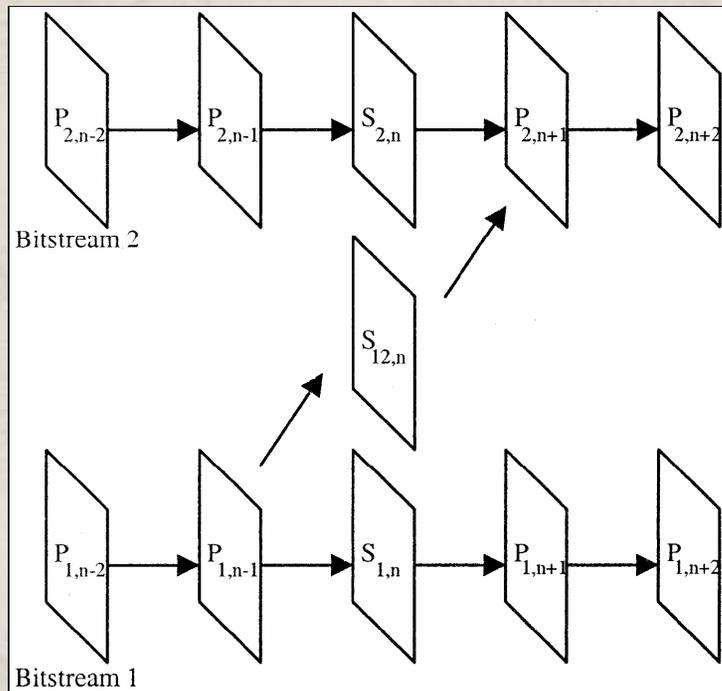
On the right side of this section, there are three dropdown menus:

- Teaser
- Trailer 2
- High Definition

# Receiver-based Transport Layer Rate Control

## ◆ Multi-rate Switching

- The previous method only allows the rate to be chosen at the beginning of each session
- Multi-rate switching enables dynamic switching within a session



## ◆ How to achieve this?

- Prior standards: switch at I frames
- H.264: SP/SI frames
  - Lower cost.

# Rate Control for Video Encoding

---

- ◆ Goal:
  - Maximize the video quality under a given encoding rate constraint
- ◆ A classical source coding problem
- ◆ Video bit rate may be *adapted* by:
  - *Varying the quantization: most useful*
  - Varying the frame rate
  - Varying the spatial resolution
  - Adding/dropping layers (for scalable coding)

# Rate Control in Different Standards

---

- ◆ H.261, MPEG-1, MPEG-2:
  - Cannot change the frame rate
  - Varying the quantization step-size is the only way
  - Not suitable for low bit rate
- ◆ H.263, MPEG-4, H.264:
  - Can change the frame rate
    - Discard a frame if the video data rate is too high
  - Suitable for low bit rate
  - MPEG-4's object-based coding provides more flexibilities for rate control

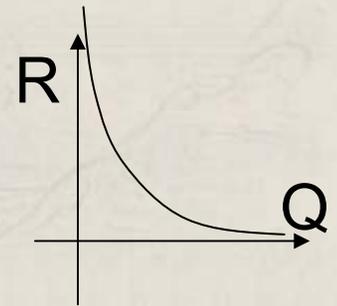
# How to adjust Quantization Stepsize?

---

- ◆ How to adjust quantization parameter (QP) to achieve the target bit rate?
  - Need Rate-distortion (R-D) theory
  - Lagrangian method is frequently used

# Rate Control in MPEG4 VM5

- ◆ T. Chiang and Y.-Q. Zhang, “A new rate control scheme using quadratic rate distortion model,” *IEEE Trans. Circuits and Systems for Video Technology*, Vol. 7, No. 1, pp. 246 – 250, Feb. 1997.
- ◆ Adopted by MPEG Verification Model (VM) 5.0 in Nov. 1996
- ◆ Rate-quantization step model



$$R = aQ^{-1} + bQ^{-2}$$

- ◆ Rationale

- General form of RD curve (high rate assumption)

$$R(D) = \ln\left(\frac{1}{\alpha D}\right)$$

- Taylor expansion

$$\begin{aligned} R(D) &= \left(\frac{1}{\alpha D} - 1\right) - \frac{1}{2}\left(\frac{1}{\alpha D} - 1\right)^2 + R_3(D) \\ &= -\frac{3}{2} + \frac{2}{\alpha}D^{-1} - \frac{1}{2\alpha^2}D^{-2} + R_3(D) \end{aligned}$$

# Rate Control in MPEG4 VM5

- ◆ Three sets of parameters,  $\{a_1, b_1\}$ ,  $\{a_2, b_2\}$ , and  $\{a_3, b_3\}$ , for I, P, B frames, respectively.
- ◆ Use linear regression to get  $a_i, b_i$ :
  - Collect  $R_{ij}, Q_j$ : the bits and  $Q$  step of each previously encoded frames in each category

$$\begin{aligned} R_{11} &= a_1 Q_1^{-1} + b_1 Q_1^{-2} \\ R_{12} &= a_1 Q_2^{-1} + b_1 Q_2^{-2} \\ &\dots \\ R_{1n} &= a_1 Q_n^{-1} + b_1 Q_n^{-2} \end{aligned} \quad \longrightarrow \quad \begin{bmatrix} R_{11} \\ R_{12} \\ \vdots \\ R_{1n} \end{bmatrix} = \begin{bmatrix} Q_1^{-1} & Q_1^{-2} \\ Q_2^{-1} & Q_2^{-2} \\ \vdots & \vdots \\ Q_n^{-1} & Q_n^{-2} \end{bmatrix} \begin{bmatrix} a_1 \\ b_1 \end{bmatrix}, \text{ or } r = A \begin{bmatrix} a_1 \\ b_1 \end{bmatrix}$$

$$\longrightarrow \begin{bmatrix} a_1 \\ b_1 \end{bmatrix} = (A^T A)^{-1} A^T r$$

Drawbacks:

1. High rate model;
2. Frame-level QP.

# Rate Control in H.263+ TMN8 and MPEG4 VM8

- ◆ J. Ribas-Corbera and S. Lei, "Rate control in DCT video coding for low-delay communications," *IEEE Trans. Circuits and Systems for Video Technology*, Vol. 9, No. 1, pp. 172-185, Feb. 1999.
- ◆ TMN: Test Model Near-term
- ◆ Advantages:
  - Macro-block level rate control
  - Suitable for low bit rate

Rate model for low bit rate:

$$R(Q) = \frac{c\sigma^2}{Q^2}, \quad \sigma^2 : \text{variance of a DCT coefficient.}$$

Algorithm

Assign QP based on the standard deviation  $V_i$  of each macroblock:  $V_i \rightarrow Q_i$

The total bits generated by the  $i$ -th macro-block:

$$B_i = A \left( K \frac{V_i^2}{Q_i^2} + C \right)$$

A: 256, # of pixels in each MB

C: overhead by motion vectors.

K, C can be estimated.

# Rate Control in H.263+ TMN8 and MPEG4 VM8

## ◆ Distortion Model

- Assuming uniform quantizer
- Average distortion:  $N$ =total number of macroblocks

$$D = \frac{1}{N} \sum_{i=1}^N \frac{1}{12} Q_i^2$$

## ◆ Problem formulation

$$\operatorname{argmin}_{Q_1, Q_2, \dots, Q_N} \frac{1}{N} \sum_{i=1}^N \frac{1}{12} Q_i^2$$

$$\text{subject to } \sum_{i=1}^N B_i = B.$$

Select QP for each MB such that the total bits for the current frame is  $B$

# Rate Control in H.263+ TMN8 and MPEG4 VM8

- ◆ Use Lagrangian multiplier:

$$\begin{aligned}
 J &= \frac{1}{N} \sum_{i=1}^N \frac{1}{12} Q_i^2 + \lambda \left( \sum_{i=1}^N B_i - B \right) \\
 &= \frac{1}{N} \sum_{i=1}^N \frac{1}{12} Q_i^2 + \lambda \left( \sum_{i=1}^N \left( A \left( K \frac{V_i^2}{Q_i^2} + C \right) \right) - B \right)
 \end{aligned}$$

- ◆ Set the derivative with respect to  $Q_i$  to be 0:

$$Q_i^2 = (\sqrt{12AKN\lambda}) V_i$$

- ◆ Plug in to the bit rate constraint

$$\sum_{i=1}^N B_i = B$$

$$\sqrt{\lambda} = \frac{AK \sum_{i=1}^N V_i}{\sqrt{12AKN(B - ANC)}} \quad \longrightarrow \quad Q_i = \sqrt{\frac{AK}{B - ANC} V_i \sum_{k=1}^N V_k}$$

# Rate Control in H.263 TMN10

- ◆ G. Sullivan and T. Wiegand, “Rate-distortion optimization for video compression,” *IEEE Signal Processing Magazine*, Vol. 15, No.6, pp. 74-90, Nov. 1998.
- ◆ T. Wiegand and B. Girod, “Lagrange multiplier selection in hybrid video coder control,” *Proceedings of 2001 International Conference on Image Processing*, Vol. 3, pp. 542-545, Oct. 2001.
- ◆ Overall goal of video rate control
  - Minimize the distortion for a given bit rate
- ◆ Many things need to be optimized in video coding
  - MB Coding Mode
    - INTRA: code the MB as intra block, no motion estimation.
    - SKIP: Use the co-located MB in the previous frame as the reconstruction for the current MB.
    - INTER 1MV: Use 1 motion vector for the MB, encode the MV and residual
    - INTER 4MV: Use 4 motion vectors for the MB. One for each 8x8 block.
  - Motion Estimation Accuracy
    - Integer pixel, half pixel, 1/4 pixel.
  - Quantization Parameter (QP)
    - Same QP as previous MB
    - Previous QP with a minor adjustment: QP+1, QP-1, QP+2, QP-2.

# Rate Control in H.263 TMN10

- ◆ Rate constrained Coding Mode Decision for the k-th MB:
  - Minimize the Lagrangian **coding mode cost**

$$J = D_{REC}(MB_k, MODE_k | Q) + \lambda_{MODE} R_{REC}(MB_k, MODE_k | Q)$$

$D_{REC}(MB_k, MODE_k | Q)$ : Distortion of the k - th MB at  $MODE_k$  and step Q.

$R_{REC}(MB_k, MODE_k | Q)$ : Rate for the MB at  $MODE_k$  and step Q.

Possible options for  $MODE_k$ :

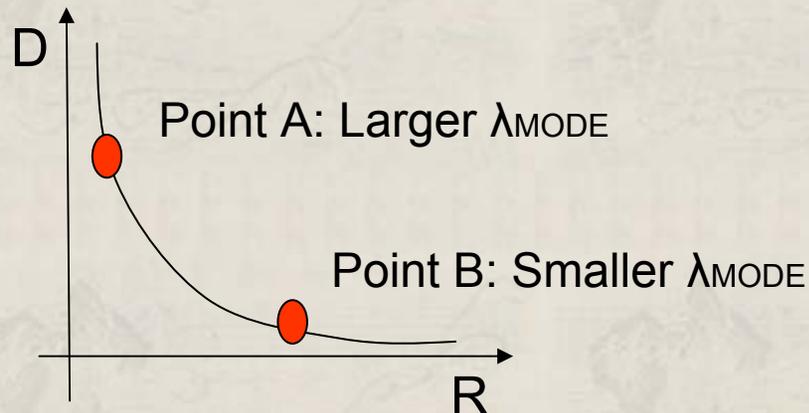
- INTRA, SKIP, INTER 1MV, INTER 4MV
- ◆  $\lambda_{MODE}$ : Lagrangian multiplier

# Rate Control in H.263 TMN10

- ◆ At high rate, the relationship between RREC and DREC is a log function

$$R(D) = a \ln\left(\frac{b}{D}\right)$$

- ◆  $\lambda$ MODE: Lagrangian multiplier:
  - Larger  $\lambda$ MODE gives higher priority to the reduction of rate, leading to solution with lower rate (and therefore larger distortion)
  - Smaller  $\lambda$ MODE gives higher priority to the reduction of distortion, leading to solution with less distortion, but higher rate

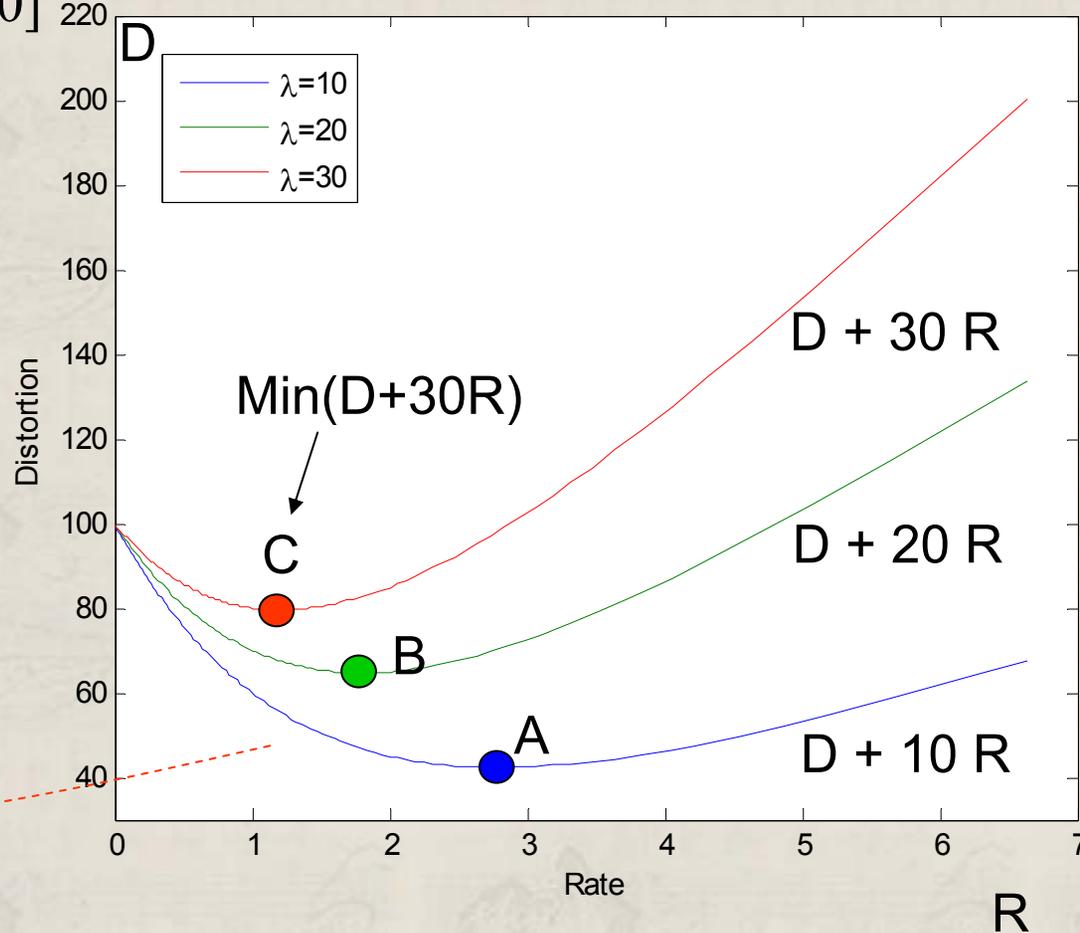
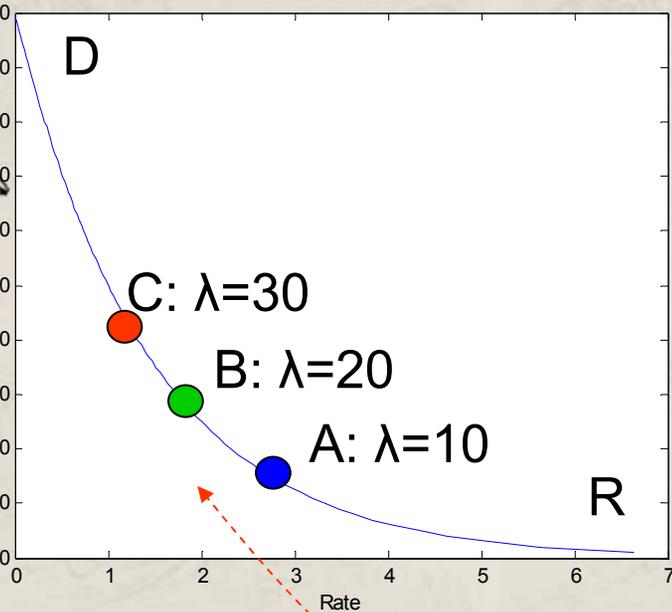


# Example

$$R(D) = \log_2\left(\frac{100}{D}\right), \quad D \in [1, 100]$$

R-D curve

Lagrangian Cost:  $D + \lambda R$



Different  $\lambda$  leads to different optimal operating points on the R-D curve.

# Rate Control in H.263 TMN10

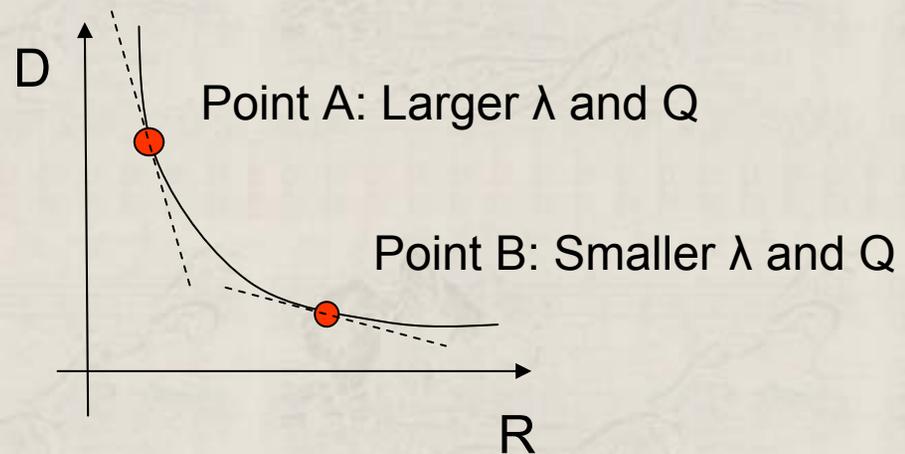
- ◆ Further interpretation of the Lagrangian multiplier  $\lambda$ :

$$R(D) = a \ln\left(\frac{b}{D}\right)$$

$$J = D + \lambda R = D + \lambda a \ln\left(\frac{b}{D}\right)$$

- ◆ When  $J$  is minimized:

$$\lambda = \frac{1}{a} D = -\frac{dD}{dR}$$



$\lambda$ : The negative slope of the optimal point on the distortion-rate curve.

**Another perspective:**

Smaller  $\lambda \rightarrow$  Larger Rate  $\rightarrow$  Smaller quantization step Q!

Larger  $\lambda \rightarrow$  Smaller Rate  $\rightarrow$  Larger quantization step Q!

# Rate Control in H.263 TMN10

- ◆ Smaller (larger)  $\lambda$  corresponds to smaller (larger) quantization step  $Q$ !
- ◆ What's the exact relationship between  $\lambda_{\text{MODE}}$  and  $Q$ ?
- ◆ The relationship between the distortion and the quantization parameter (at high rate):

$$D(Q) = \frac{1}{12} Q^2$$

Plug into the R(D) formula:

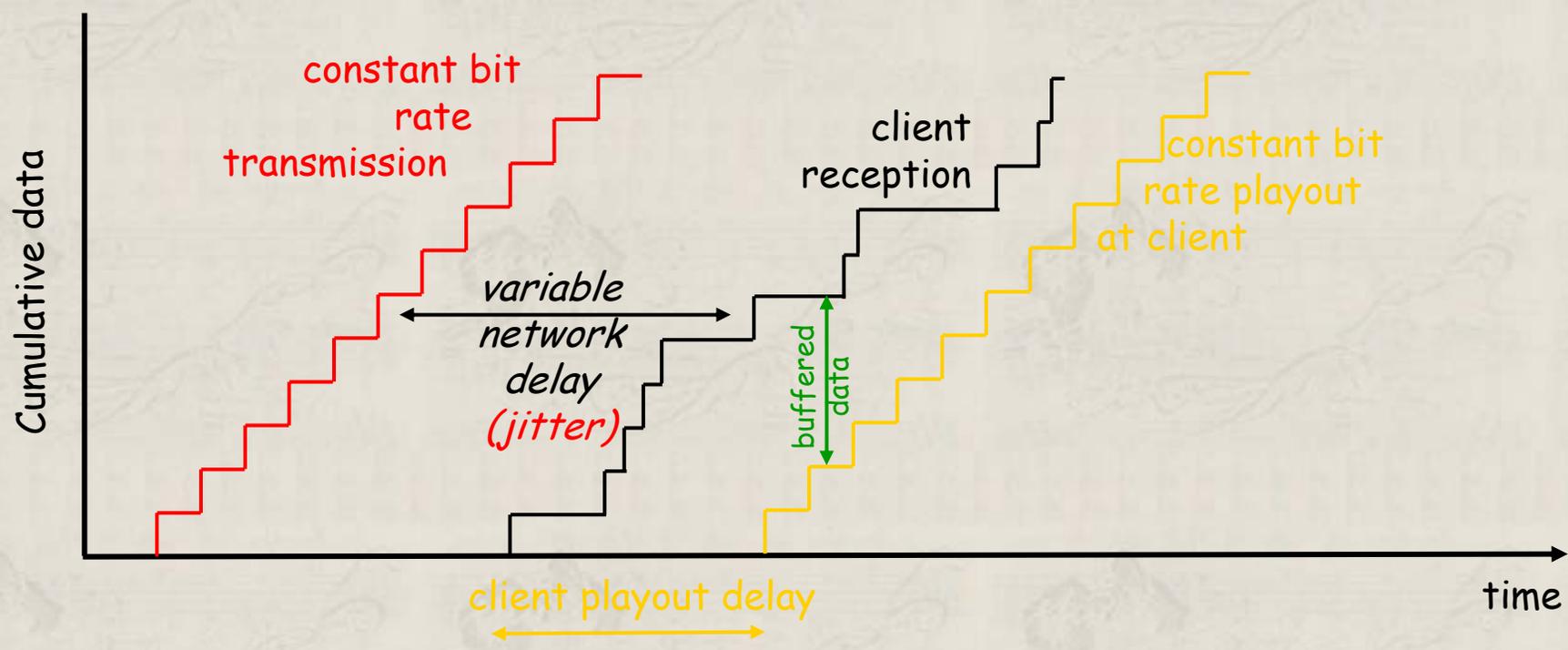
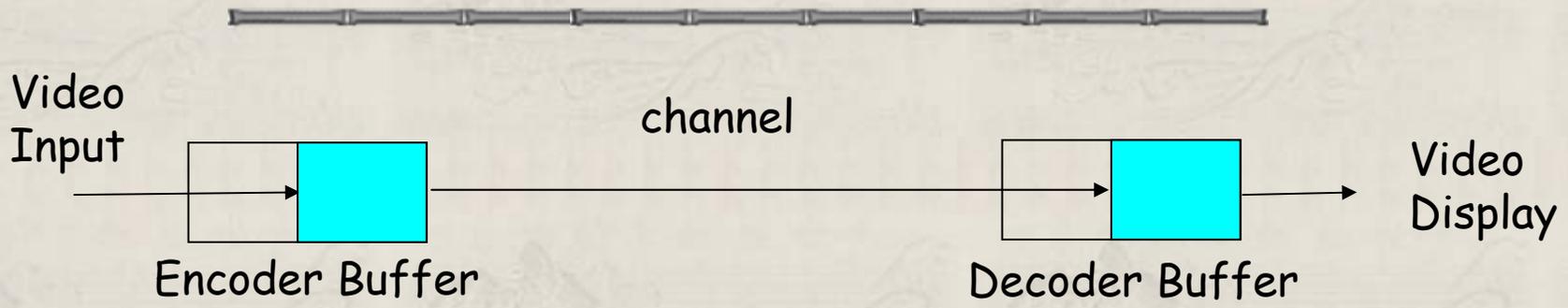
$$R(D) = a \ln\left(\frac{b}{D}\right) \quad \longrightarrow \quad R(Q) = a \ln\left(\frac{12b}{Q^2}\right)$$

$$\longrightarrow \quad \lambda = -\frac{dD}{dR} = -\frac{dD(Q)/dQ}{dR(Q)/dQ} = -\frac{1/6Q}{-2a/Q} = \frac{a}{3} Q^2$$

In H.263,

$$\lambda_{\text{MODE}} = 0.85Q^2$$

# Buffers in Video Transmission



# Advantages of Buffering

---

- ◆ Jitter reduction
- ◆ Error recovery through retransmission
- ◆ Error resilience through interleaving:
  - Transform burst error into isolated error to facilitate error concealment
  - Especially useful for streaming audio
- ◆ Smoothing throughput fluctuation
- ◆ Buffer Outage
  - Buffer overflow
  - Buffer underflow
- ◆ Our interests: How to decide the decoder buffer size?

# Hypothetical Reference Decoder

---

- ◆ Defined by video coding standards (H.263, MPEG, H.264)
- ◆ Goal: impose basic buffering constraints on the bit-rate variations of compliant bit streams.
- ◆ Video coding standards require encoders to control generated bit-rate such that a hypothetical reference decoder (HRD) of a given buffer size can decoder the bit stream without buffer overflow or underflow.
- ◆ **Leaky bucket parameters: Model for buffer control**
  - R: peak transmission bit rate
  - B: buffer size
  - F: Initial decoder buffer fullness (related to playout delay)

# Hypothetical Reference Decoder

## Usage:

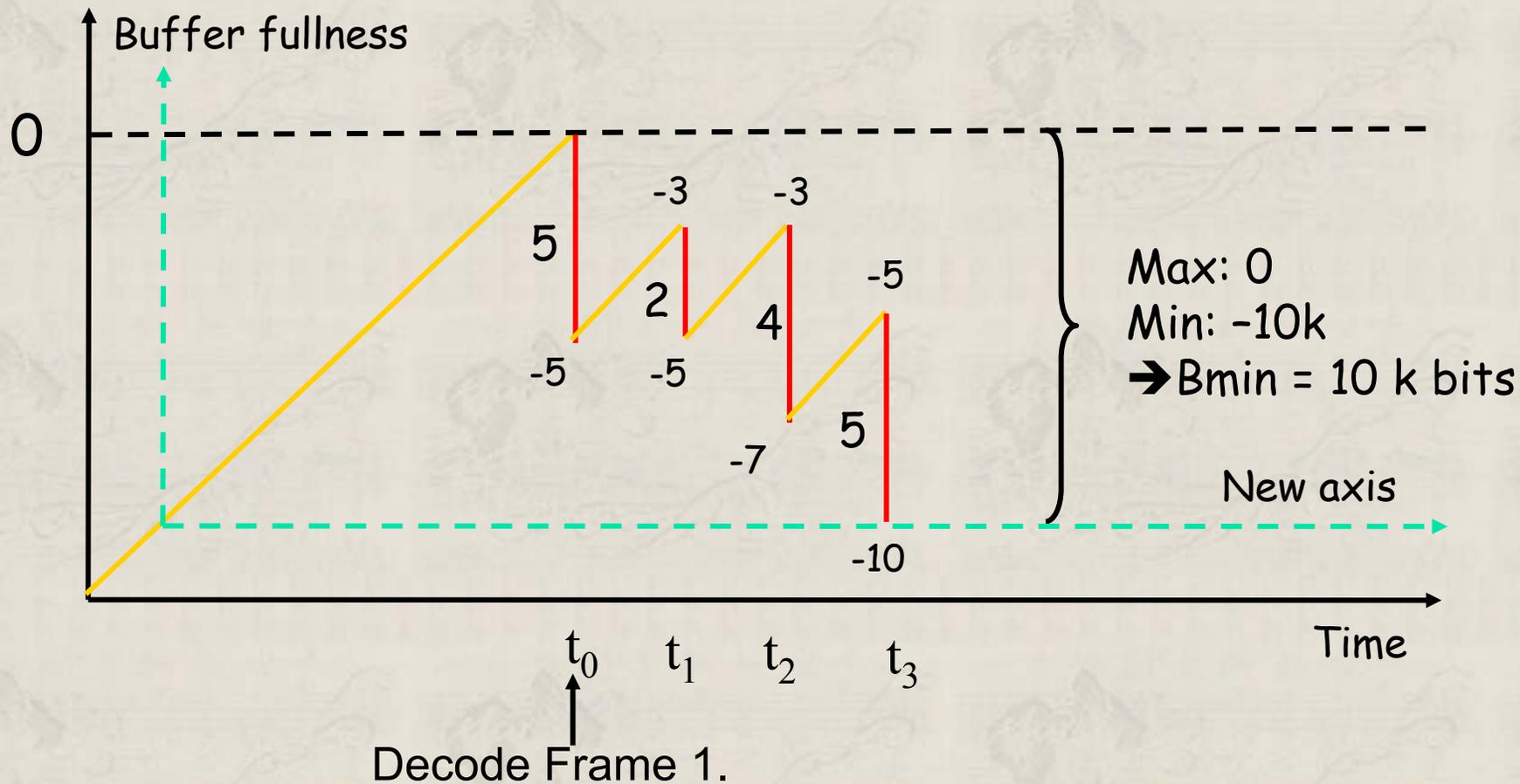
- ◆ After encoding a video sequence, can find the valid (R, B, F) that can be used to decoder it. Send this to decoder
- ◆ Useful for the decoder to determine whether it can decode a bit stream and what playout delay is needed
- ◆ Bmin: Minimal buffer size
- ◆ Fmin: Minimal initial buffer fullness
- ◆ An algorithm to find Bmin and Fmin from the given compressed video sequence  $\{b_0, b_1, \dots, b_{N-1}\}$  and transmission rate R:
  - Decoding the sequence from buffer level 0 without considering overflow and underflow, find the highest level and the lowest level for the sequence.
  - **Bmin = HIGH – LOW;**
  - **Fmin = –LOW;**

# Hypothetical Reference Decoder Example

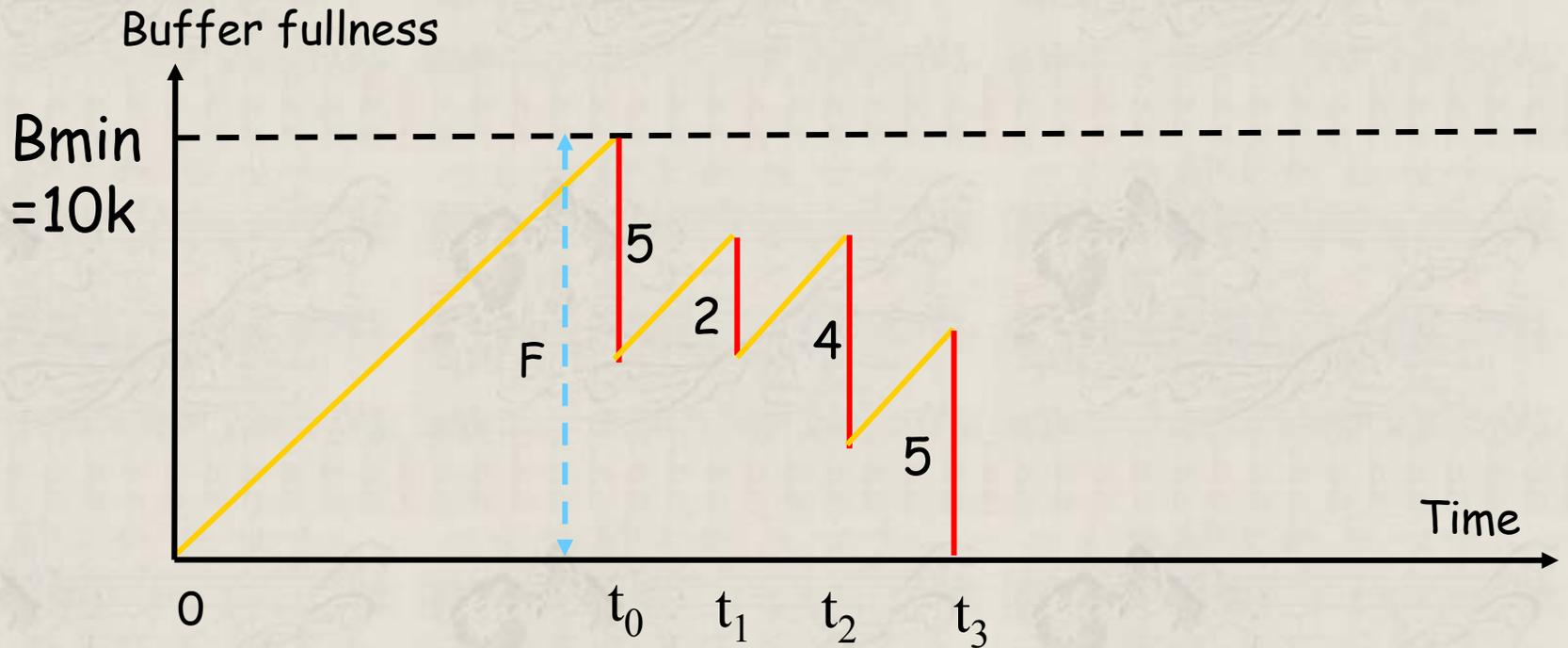
Frame rate 30 fps, or  $T = 1 / 30$  sec,

Frame bits  $b_i = \{5k, 2k, 4k, 5k\}$ ,

Consider rate  $R = 60$  k bps  $\rightarrow RT = 2k$  bits / frame interval.



# Hypothetical Reference Decoder Example



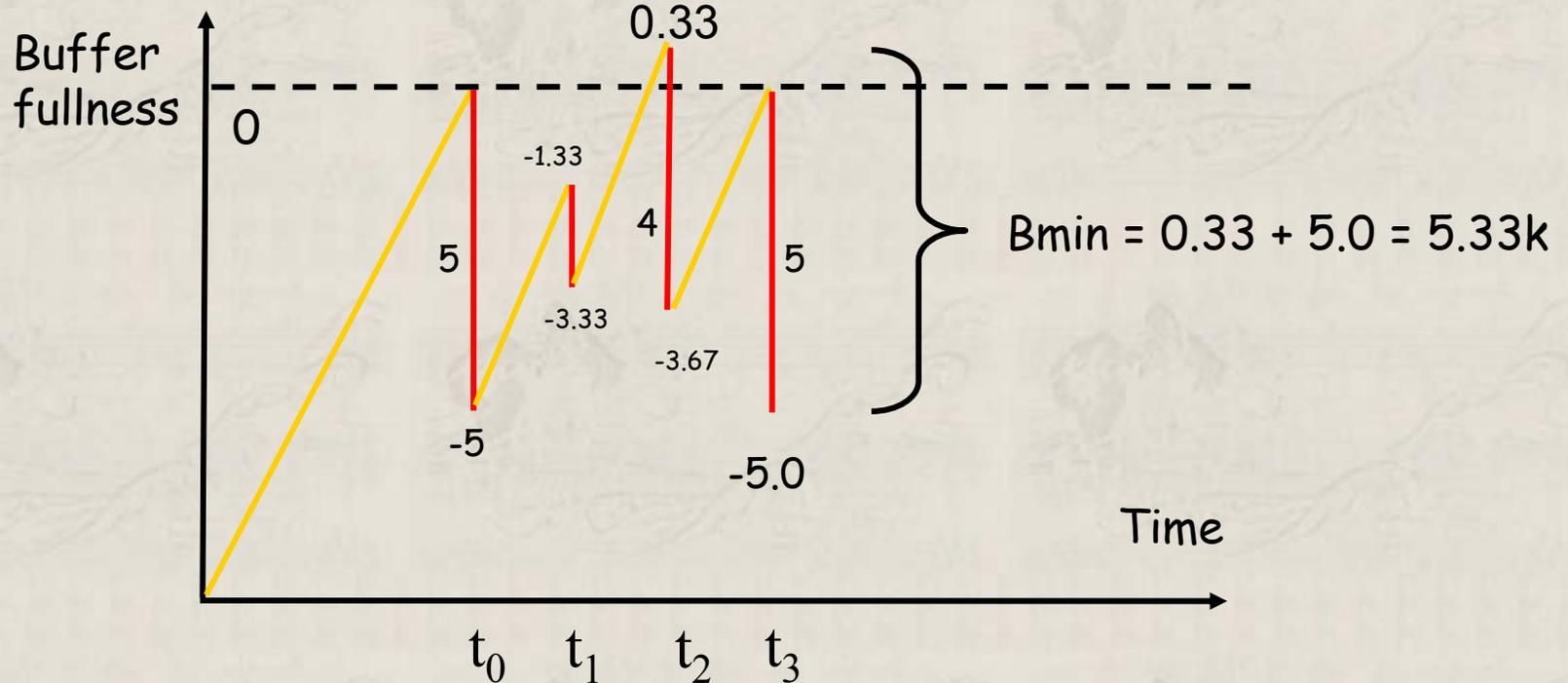
What is the minimal initial buffer fullness  $F_{min}$ ?

$F_{min} = B_{min} = 10k$  in this example.

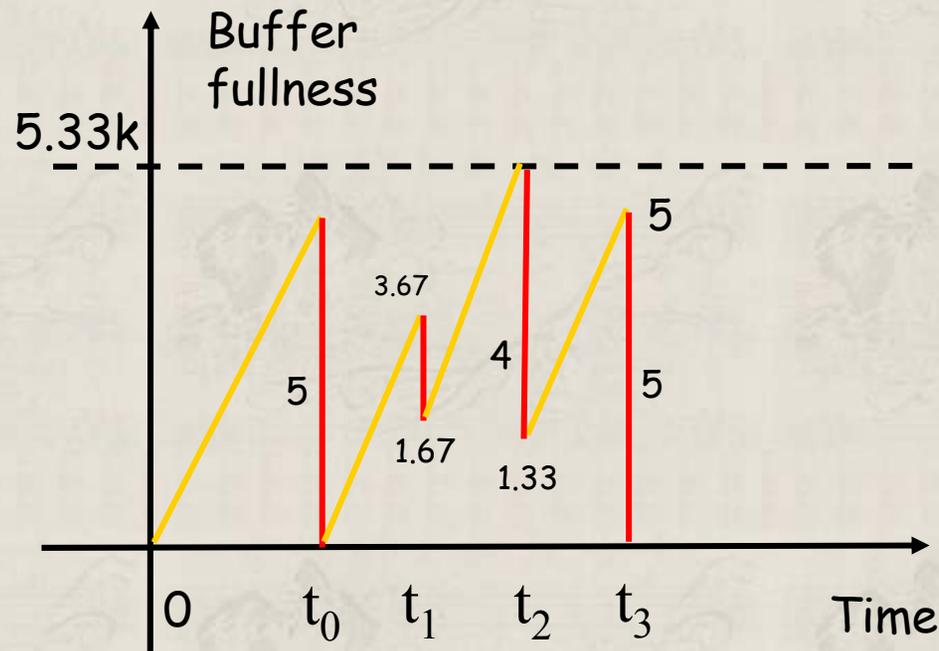
Decoder Playout delay:  $10k / 60k = 1/6$  sec

# Hypothetical Reference Decoder Example

Frame rate 30 fps, or  $T = 1 / 30$  sec, Frame bits  $b_i = \{5k, 2k, 4k, 5k\}$ ,  
Consider rate  $R = 110$  k bps  $\rightarrow RT = 3.67$  k bits / frame interval.



# Hypothetical Reference Decoder



What is the minimal initial buffer fullness?

$F_{min} = 5k$  bits

Decoder Playout delay:  $5k / 110k = 1/22$  sec

# Hypothetical Reference Decoder

---

- ◆ Summary:
- ◆  $R = 60$  kbps,  $B_{\min} = 10$  k bits,  $F_{\min} = 10$  k bits
- ◆  $R = 110$  kbps,  $B_{\min} = 5.33$  k bits,  $F_{\min} = 5$  k bits

*Higher channel rate → Smaller buffer required*

Key Observation:

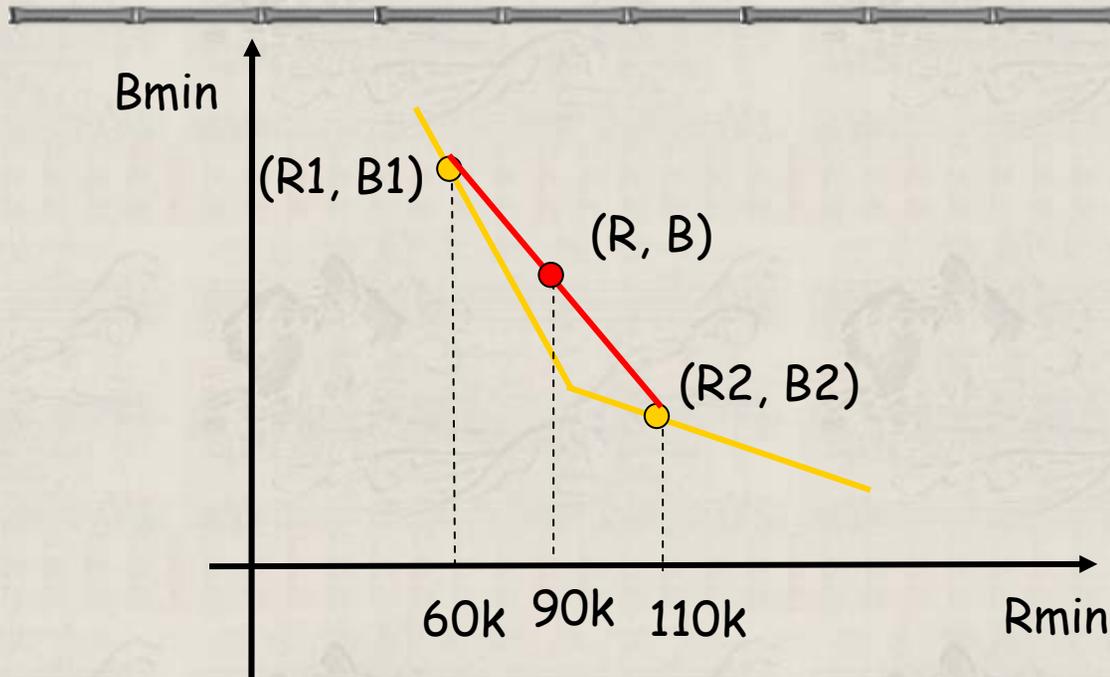
A given video stream can be decoded by many leaky bucket settings.

# H.264 HRD

---

- ◆ Multiple valid  $(R_i, B_i, F_i)$  are provided
- ◆ A decoder can choose the most suitable leaky bucket to decode the bit stream.
- ◆ A decoder can use interpolation to find a valid leaky bucket for itself.
- ◆ One encoded sequence can be decoded by receivers of different configurations
- ◆ J. Ribas-Corbera, P. A. Chou, S. L. Regunathan, “A generalized hypothetical reference decoder for H.264/AVC,” *IEEE Trans. Circuits and Systems*, Vol. 13, No. 7, pp. 674-687, Jun. 2003.

# Relationship between Bmin and Rmin



- ◆ The Bmin (R) curve is **piece-wise linear** and **convex**
- ◆ Given points (R1, B1) and (R2, B2) on the curve, can use **linear interpolation** to find a valid point (R, B) that also contains the video stream:

$$\frac{R_2 - R}{R - R_1} = \frac{B_2 - B}{B - B_1} \quad \longrightarrow \quad B = \frac{(R_2 - R)B_1 + (R - R_1)B_2}{R_2 - R_1}$$

$B \geq B_{\min}(R)$  by convex property of the curve.



# Wireless Multimedia Streaming: Challenges

---

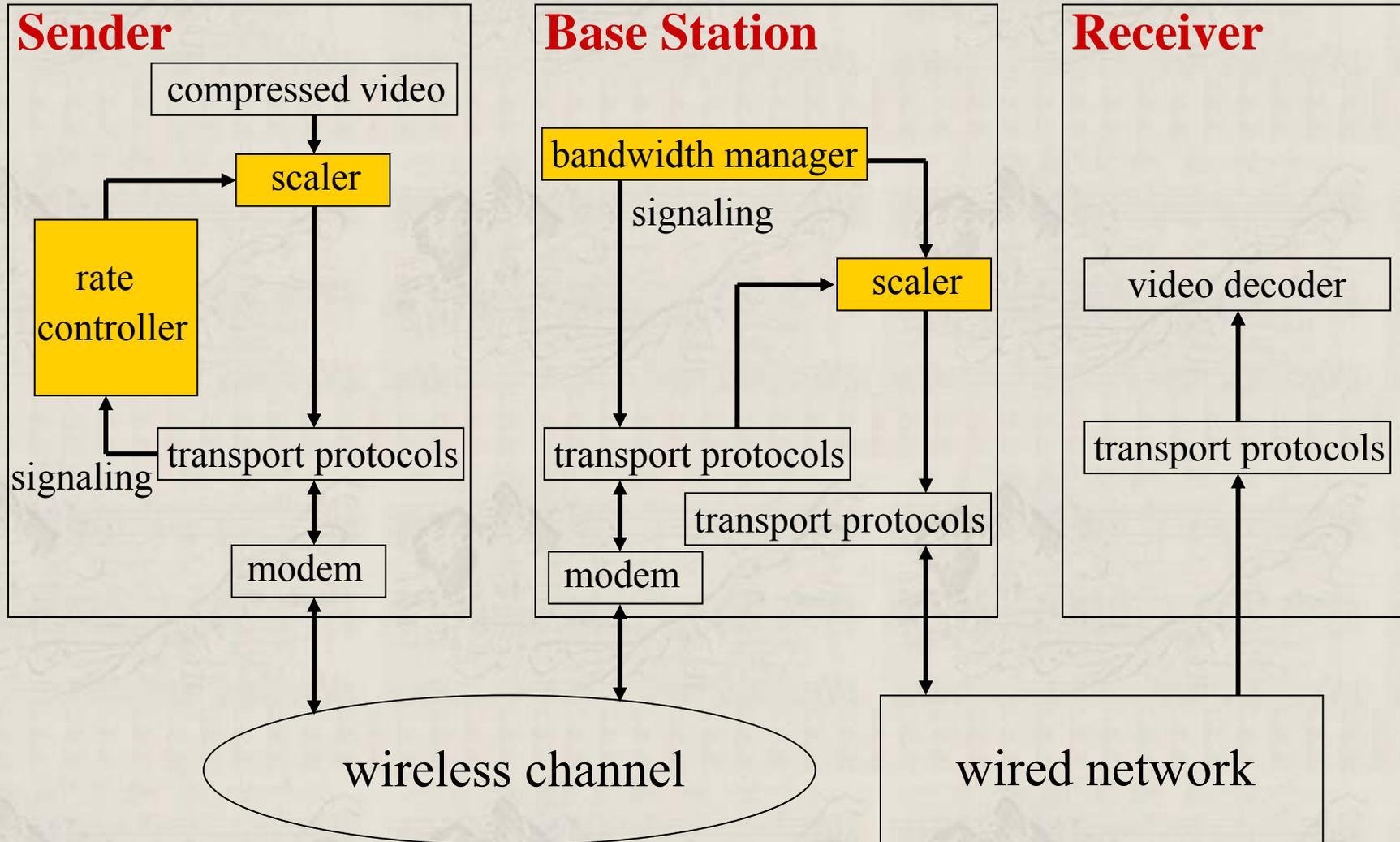
- ◆ Bandwidth fluctuations
  - Multipath fading, cochannel interference, noise disturbances, changing distances
  - Mobile devices moves between cells, networks: hand-off problems
- ◆ High bit-error rate
  - More lossy channels comparing to wired links
  - Small-scale (multipath) and large-scale (shadowing) fading
- ◆ Heterogeneity of mobile transmitters/receivers
  - Latency, processing capability, power limitations, bandwidth limitations

# Key Desirable Features

---

- ◆ Graceful quality degradation
  - Scalable video coding and communications
  - Perceptual quality gracefully degraded under severe channel conditions
- ◆ Efficiency
  - Excess bandwidth maximizing quality
  - Excess computational power for error concealment
- ◆ Fairness
  - Network resources shared in either utility-fair or max-min fair manner

# Network-Aware Applications



Scalable video from mobile device to wired terminal

# Adaptive Service

---

## ◆ Goal

- Provide scaling of a scalable video sub-stream based on the resource availability conditions in the network

## ◆ Functions

- Reserve minimum bandwidth for base layer
- Adapt enhancement layers based on available bandwidth and fairness policy

## ◆ Advantages

- Adaptivity to network heterogeneity
- Low latency & low complexity
- Lower call blocking & handoff dropping probability

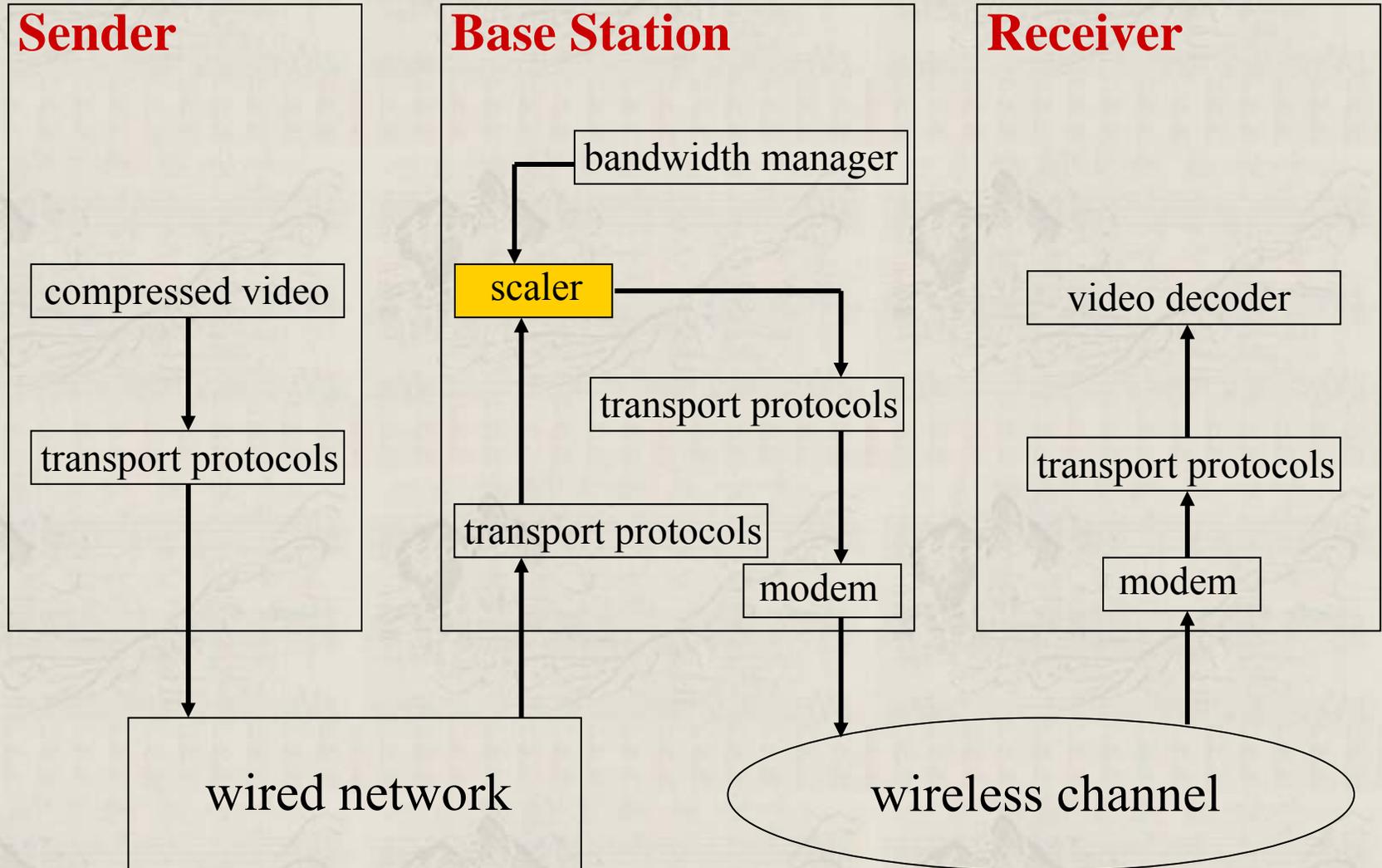


# Adaptive Service: Required Components

---

- ◆ Service contract
- ◆ Call admission control & resource reservation
- ◆ Mobile multicast mechanism
- ◆ Sub-stream scaling
- ◆ Sub-stream scheduling
  - Prioritized packet scheduler
- ◆ Link-layer error control
  - FEC, ARQ

# Sub-stream Scaling



Scalable video from wired terminal to mobile device

# Summary

---

- ◆ Brief survey of major approaches & mechanisms for Internet as well as wireless streaming of multimedia
- ◆ A thorough understanding of the entire streaming architecture is beneficial for the development of advanced signal processing techniques
- ◆ Many challenges lead to many open opportunities in the near future