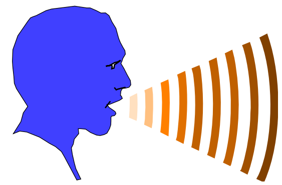


Video Processing & Communications

Error Control in Video Communications

Yao Wang
Polytechnic Institute of NYU, Brooklyn, NY11201
(with Significant Contribution from Amy Reibman)

Outline

- Characteristics of typical applications and networks
- Necessity/challenge for error control (impact of errors)
- Error resilient encoding
- Error concealment
- Encoder/decoder interactive error control
- Video streaming fundamentals

Video Communications Applications

- Interactive two-way visual communications
  - Low delay is essential, round trip delay under 150 ms desired, up to 400 ms may be acceptable
  - Real-time encoder/decoder essential
  - Audio/visual synchronization required to maintain lipsync
  - Some visual impairments may be acceptable
- One-way video streaming
  - Higher delay is OK (up to 10’s of seconds)
  - May not require real-time encoder
  - Many different rates and capabilities of decoder
- One-way video downloading
  - Video as a file; therefore no different than file downloading

Interactive two-way visual communications

- Ex. Teleconferencing, video telephony, virtual classroom
- Very stringent delay requirement
  - <=150 ms (one way) desired
  - 150-400 ms can be acceptable
  - >400 ms not acceptable
  - Audio and video must be in sync to maintain lip sync.
  - Both encoding and decoding must be completed in real-time.
- Only low to intermediate video quality is required
  - QCIF at 5-10 fps acceptable for video telephony
  - CIF at 10-20 fps satisfactory for video conferencing
  - Moderate amount of compression/transmission artifacts can be tolerated.
- Raw video has limited motion -> easier to code and conceal errors

One-Way Video Streaming

- Ex. TV broadcast, Multicast of a conference/event, Video streaming from Internet
- Except for live broadcast/multicast, can pre-compress the video, but decoding must be done in real-time
- Initial playout delay can be up to a few seconds
  - Receiver uses a large smoothing buffer to store several seconds of video frames before starting to display the first received frame
- Bit rate/video quality can vary widely depending on the applications
- Recipients of the same video source may be connected to the network with different access links (e.g. wireless modem to 100 mbps fast ethernet) and the receiving terminal may have varying computing power (palm vs. laptop vs. desktop)
  - Scalable coding desired

Challenge for Video Communications

- Real networks are unreliable
  - Wireless networks: random bit errors, long burst errors, and possibly link outages (can be quite high, around 30%)
  - Internet: packet loss and variable delay due to network congestion (very low 10^{-9} to around 10%)
  - Excessive delay = loss for real-time applications

- Real networks are heterogeneous in bandwidth and reliability

Network and Video Disconnect

- Many networks are engineered to reduce the high loss and error rates
  - Often this increases delay through the network
  - Even with streaming, video data is delay-sensitive
  - Once video decoding begins, it must continue or quality degrades

- Video may not be as vulnerable to packet losses and bit errors as data is
  - Data requires retransmission; any error or loss needs to be fixed
  - Video can be engineered to tolerate SOME loss and error

Steps involved in a Communication Session

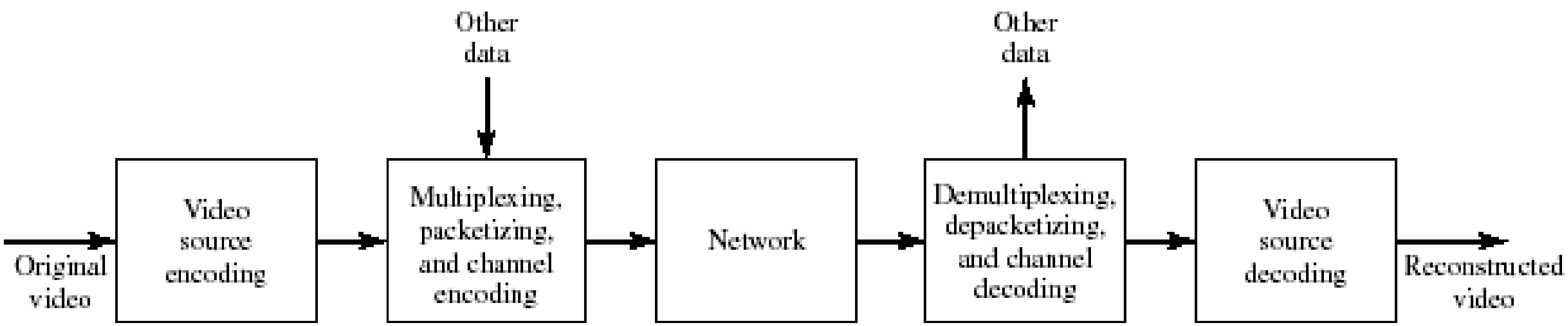


Figure 14.1 A typical video communication system.

End-to-End Delay

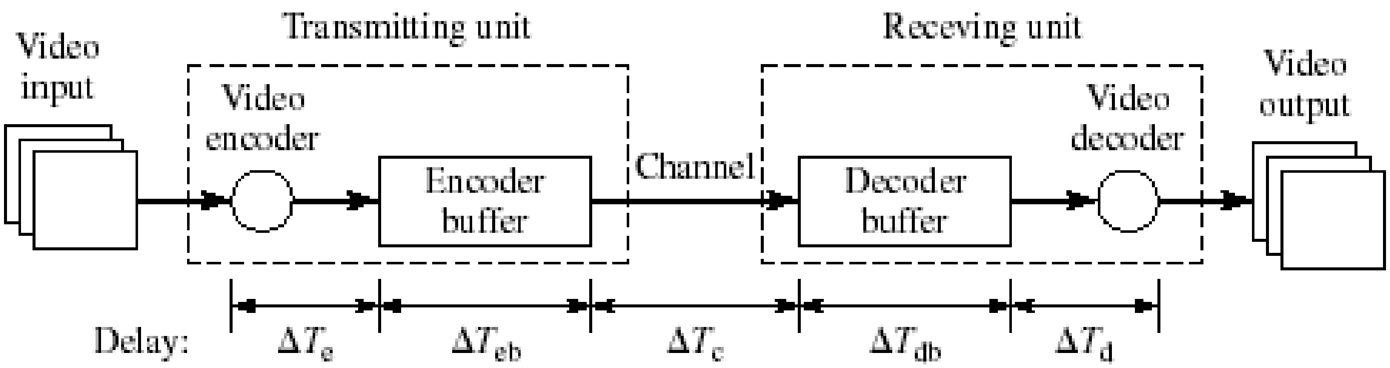
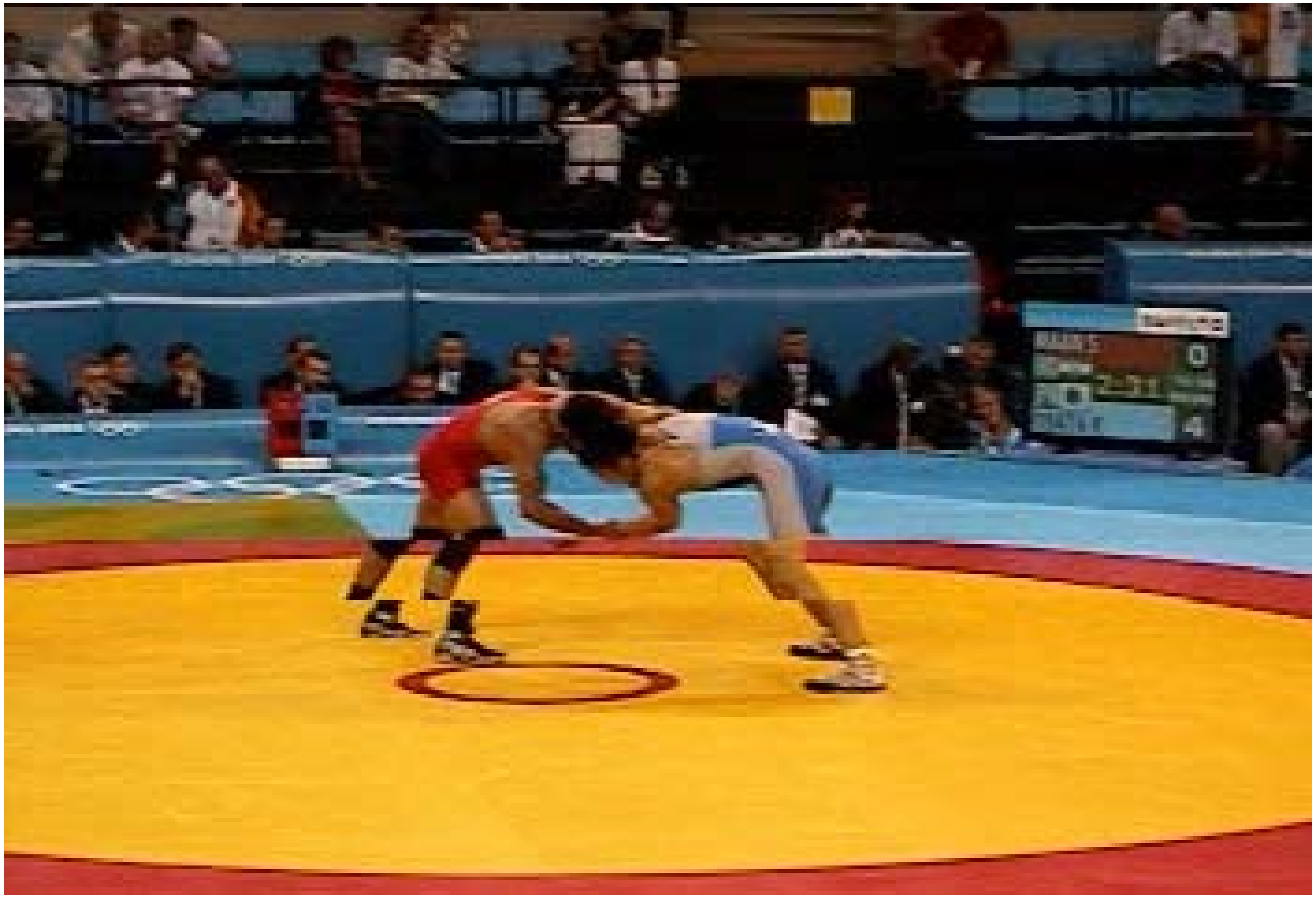


Figure 14.4 Factors contributing to the end-to-end delay in a video communication system. Adapted from A. Ortega and K. Ramchandran, Rate-distortion methods for image and video compression, *IEEE Signal Processing Magazine* (Nov. 1998), 15:23–50. Copyright 1998 IEEE.

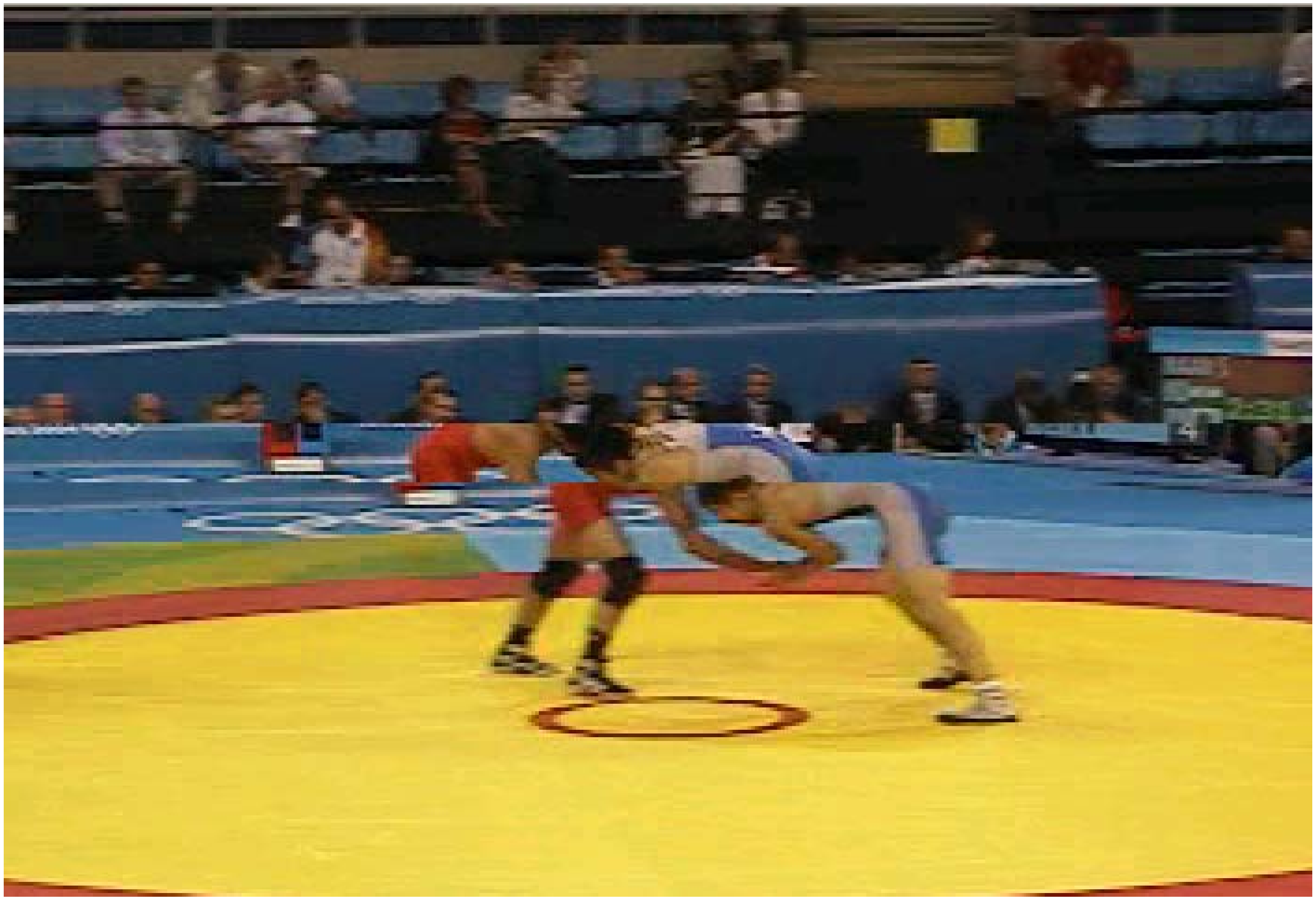
Causes of Packet Losses

- For wireless channels, FEC is necessary to reduce raw bit error rates
  - FEC along bits within each packet
  - Sufficient number of correctly received bits in each packet make it decodable
  - With (n,k) code, the number of errors must be <= (n-k)/2
  - Packets with more erroneous bits are usually dropped at the IP layer! (not passed to the video decoder)
- For wired networks, packet loss is mainly due to congestions
  - Packets arriving past the decoding deadline are effectively lost
  - Packets arriving in bursts can cause receiving buffer overflow
- For IP based networks (including wireless links), errors seen by the video decoder are typically packet loss (due to either network congestion, or packets with uncorrectable errors)

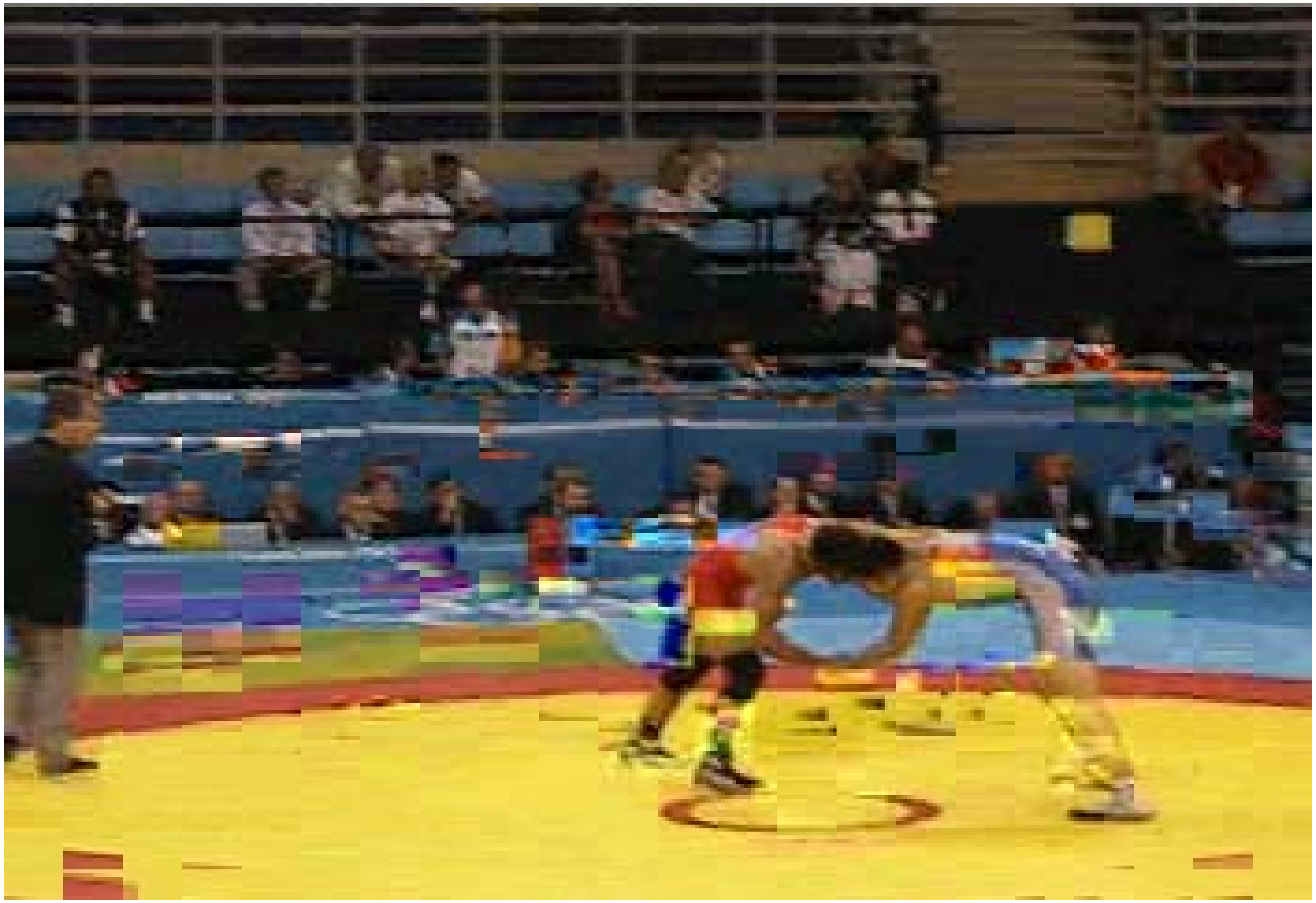
Packet losses



Packet losses



Packet losses



Compression

Bit errors

Variable-length decoding: example revisited

- Let A={1}; B={01}; C={001}; D={0001}

A A A C B A A D A A

1 1 1 001 01 1 1 0001 1 1

- 1110010111000111 sent to decoder
- 1110010101000111 received by decoder

1 1 1 001 01 01 0001 1 1

A A A C B B D A A

Problem with Variable Length Coding

- One bit error or packet loss could cause subsequent bits/packets non-decodable.
- Must segment and packetize the data to ensure subsequently received data can still be useful

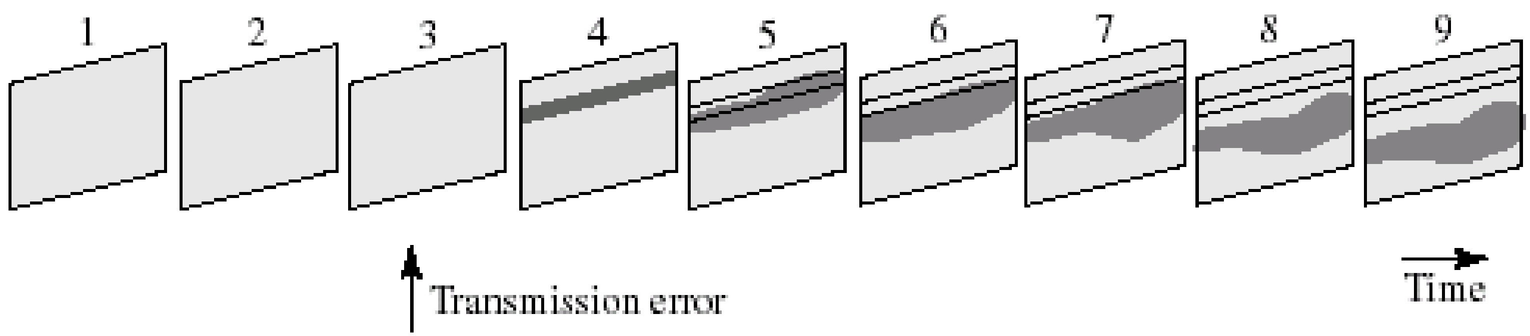
Compressed video data is sensitive to transmission errors

- Cause of Error propagation
  - Variable length coding
  - Temporal predictive coding
  - Spatial predictive coding
- All contribute to error propagation either within the same frame or also in following frames

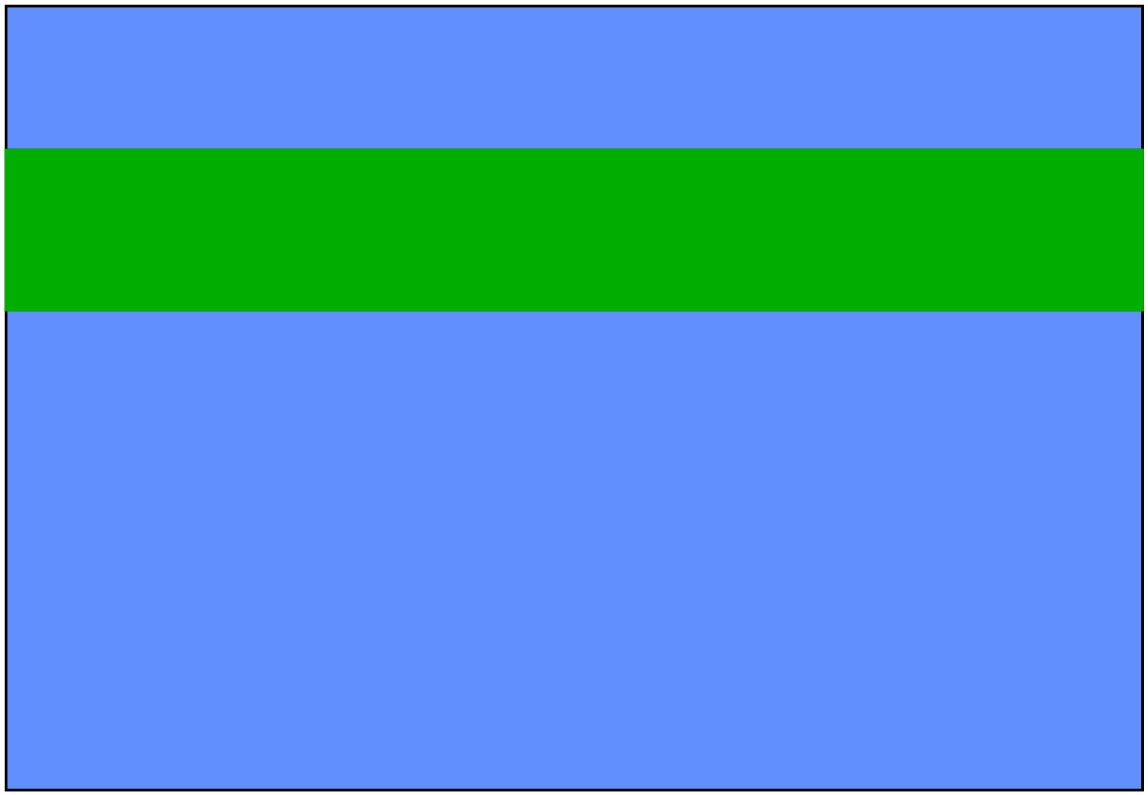
Packet losses: temporal impact

- A loss in a reference (I or P) frame will propagate with time
  - Predicting from an erroroneous frame will propagate errors
  - Motion compensation causes errors to propagate spatially!
- Loss in a B-frame will not propagate
  - No other frames use B-frames to predict
  - (not true with the hierarchical B structure in H.264)
- A correctly received I-frame will stop error propagation

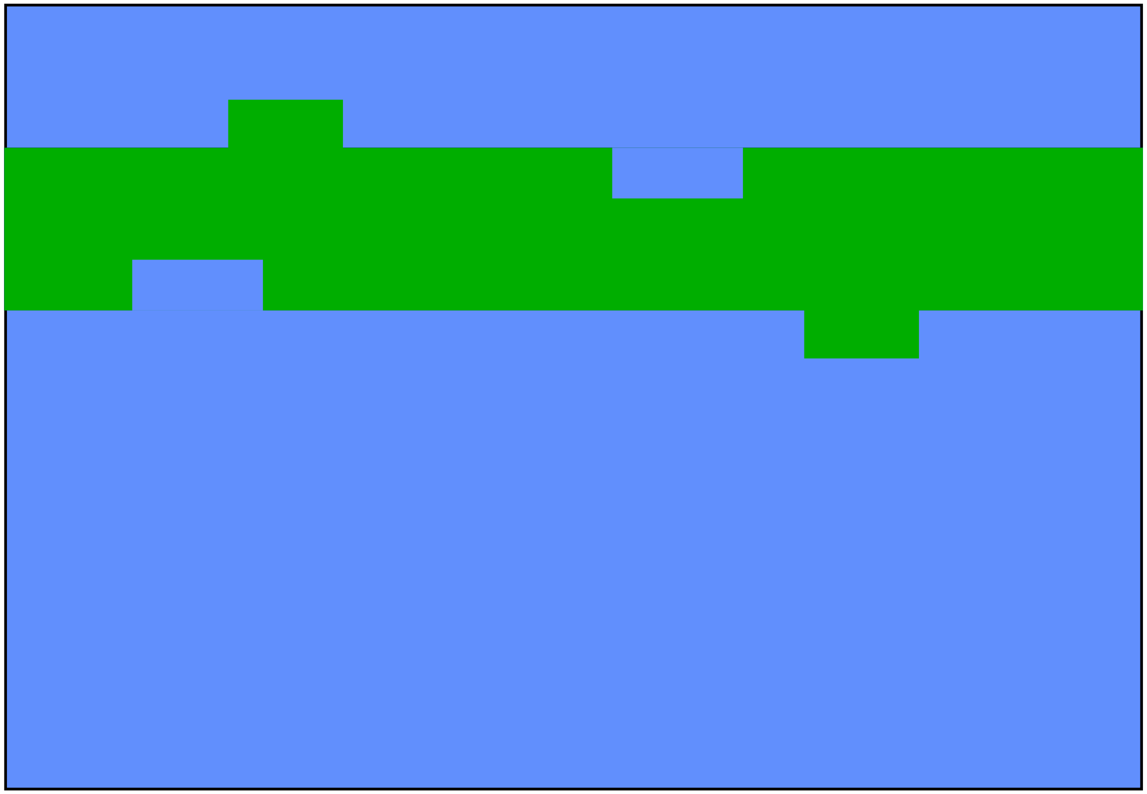
Spatial/Temporal Error Propagation



Spatial and temporal impact of a loss

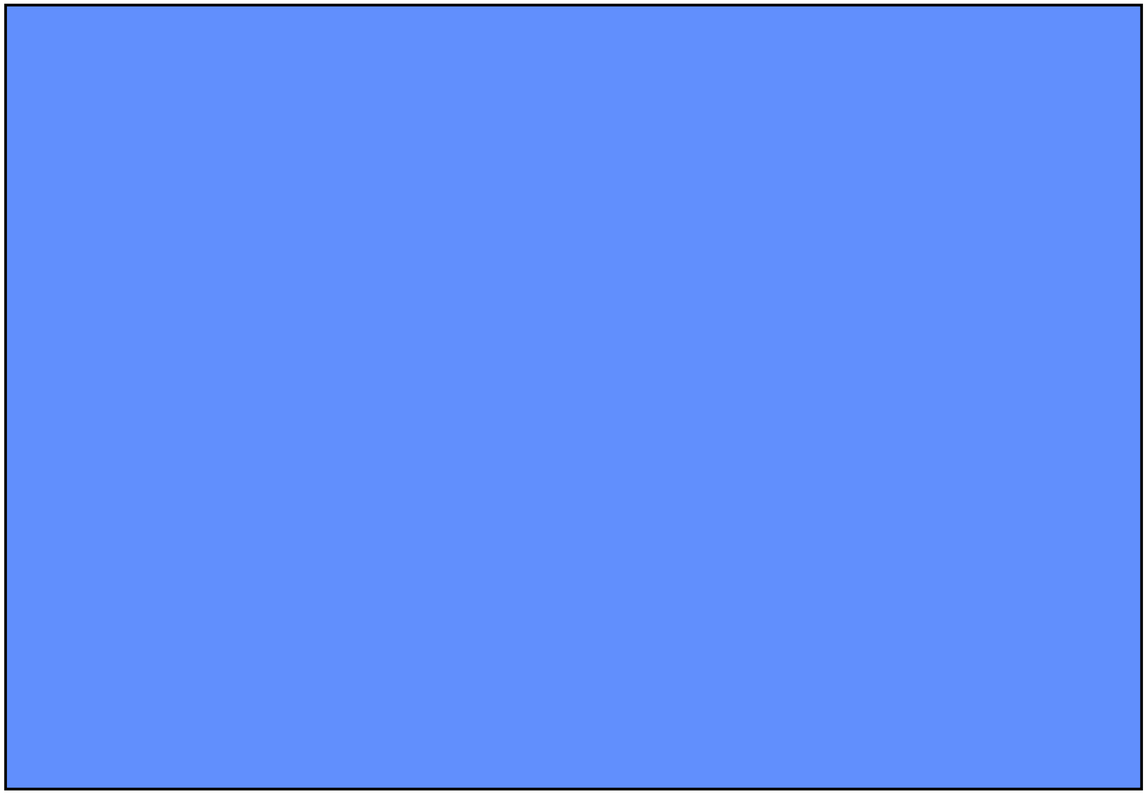


Spatial and temporal impact of a loss



Motion-compensation propagates error spatially and temporally

Spatial and temporal impact of a loss



I-frame
clears
out errors

Effect of Transmission Errors

Coded,
No loss

3%

5%

10%

Example reconstructed video frames from a H.263 coded sequence, subject to packet losses
Note that error seen in a frame may be due to losses in previous frames

Error Control Techniques for Video

- Transport level error control only
  - Error detection and correction through FEC
  - Retransmission (ARQ) of lost packets
  - Error-resilient packetization and unequal error protection (UEP)
- Error concealment (decoder only)
  - Recover lost/damaged regions at the decoder
- Error resilient encoding (encoder only or encoder+decoder)
  - Add redundancy to video bitstream to assist decoder recovery
- Encoder-decoder-network interactive error control
  - Feedback-based adaptive encoding
    - Ex. Reference picture selection, Selective intra update, rate shaping
  - Path diversity
    - Different bitstreams sent through separate paths
      - Lyered coding with baselayer sent on more reliable path
      - Multiple description coding with parallel paths

Transport Level Error Control

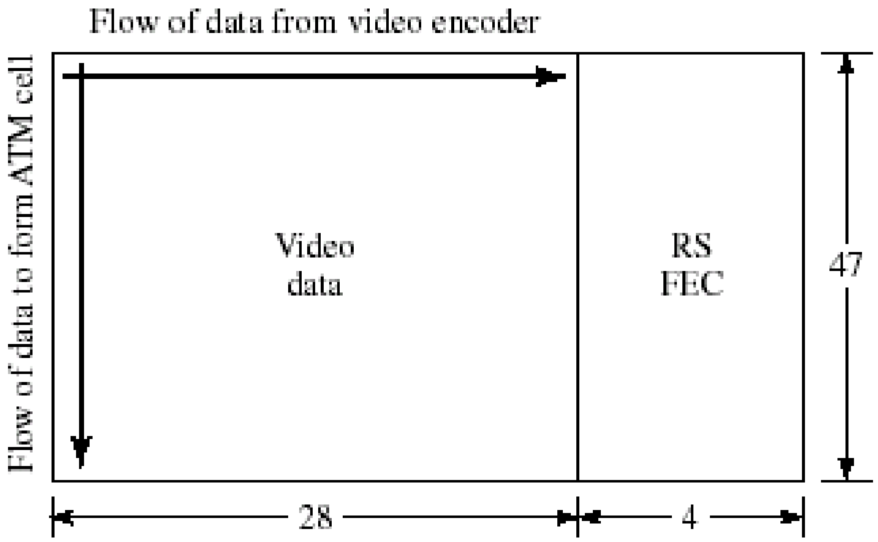
- Forward Error Detection and Correction (Channel Coding)
- Retransmission (Automatic Retransmission Request or ARQ)
- Error resilient packetization and multiplexing
- Unequal error protection

Forward Error Correction (FEC)

- Adding redundancy bits on compressed source bits to enable error detection and correction
- Simple example: Add a parity check bit at the end of a block of datastream, can detect all single bit errors
- Channel coding rate:
  - For every k source bits, add l channel bits, to create n=k+l bits -> channel coding rate r=k/n
  - Well designed code (e.g. Reed-Solomn code) can correct t=l/2 error bits in each n-bit block

Packet Level FEC

- Recall in IP-based networks, a video decoder mainly see packet losses
- FEC can be applied across packets to correct/detect packet losses
  - With packet losses, which packets are lost are known (called erasers).
  - With (n,k) code, receiving any k out of n packets can recover k source packets!



Shannon theorem for communication

- Shannon theorem for communication:
  - Source and channel codes can be designed separately:
    - Source coding minimizes the bit rate necessary to satisfy a distortion criterion (Shannon rate-distortion theory)
    - Channel coding adds just enough redundancy bits to reduce the raw channel error rate to the permitted level
  - Only valid for stationary source and channel and requires processing of infinitely long blocks of data (delay = infinity!)
- Practical system limitations
  - Video are not stationary: content changes in time!
  - Allowed channel coding length (FEC block length) is limited due to delay constraint and complexity constraint
  - Joint design of video coding and error control (including channel coding) can bring additional gain.
    - Unequal error protection
    - Multiple description coding

Delay-Constrained ARQ

- ARQ Basics:
  - receiver requests retransmission of a lost or erroneously delivered packet, incorporated in TCP
- For data transmission, ARQ is an effective mechanism for error control
- For video applications, ARQ must be limited to within the delay constraint of the application
  - How many retransmission attempts are acceptable depends on the round-trip time (RTT)
  - Should only apply ARQ to "important" packets (base-layer) (another way to achieve UEP)
- For broadcast/multicast applications, ARQ is inappropriate in general, although it can be deployed at the link layer

Error-Resilient Encoding

- Basic idea: intentionally insert redundancy in source coding to help recover from transmission errors
- Design goal: minimize the redundancy to achieve a desired level of resilience
- Error isolation (part of H.263/MPEG4 standard)
  - Inserting sync markers
  - Data partition
- Robust binary encoding
  - Reversible VLC (RVLC) (part of H.263/MPEG4 standard)
- Error resilient prediction
  - Insert intra-mode periodically (accommodated by the standard)
  - Independent segment prediction (part of H.263/MPEG4 standard)
- Layered coding with unequal error protection
- Multiple description coding

Basic Design issues for Error Isolation

- How far between each I-frame?
  - Speed of channel changing
  - Error resilience
  - Compression
- How far between synchronization points?
  - Overhead (less bit-rate for video)
  - Better resilience given packet losses or bit-errors
- How far between each B-frame?
  - Better compression
  - More memory, longer delay

Bit errors

- A single bit flips from zero to one, or one to zero
- Variable length decoder may get lost
  - Looks similar to a packet loss
- Decoder may not get lost
  - May be much MUCH worse!
  - Motion vector may change sign
  - Run-length may be errored
  - DC coefficient may change sign
- Bit errors should be avoided if at all possible

Variable-length decoding: example revisited

- Let A={1}; B={01}; C={001}; D={0001}

A A A C B A A D A A

1 1 1 001 01 1 1 0001 1 1

- 1110010111000111 sent to decoder
- 1110010101000111 received by decoder

1 1 1 001 01 01 0001 1 1

A A A C B B D A A

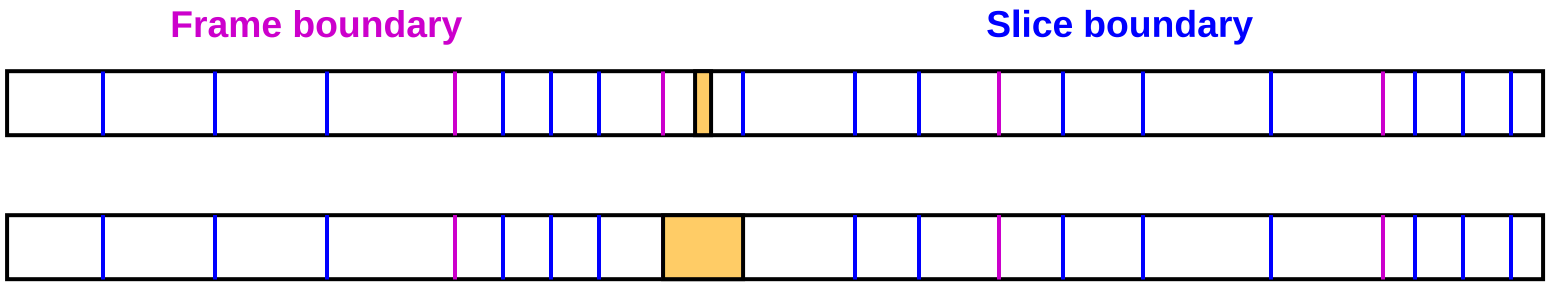
Inserting Synchronization Markers

- Variable length coding causes major problems!
  - Don’t know how much information was lost
  - Don’t know where to put newly decoded information
  - Are these bits coefficients? Motion vectors?
- Solution: insert synchronization codewords periodically
  - Easy to find: 000000000000000000000001
  - Picture_Start_Code, Slice_Start_Code
  - Thirty slices in each frame (MPEG-2)
  - Much larger slices in H.264

Video Slice

- Slice structure in video coding:
  - Each frame may be divided into multiple slices, with header at the beginning of each slice, allowing it to be decodable independent of previous slices.
  - By default (non-slice mode): H.264 put entire frame into one slice
- Slice size selection
  - Small slices improves robustness to channel errors, but reduces coding efficiency!
- Different slice modes:
  - Equal size in bytes (more complicated)
  - Equal size in pixels

Impact of Slice Size

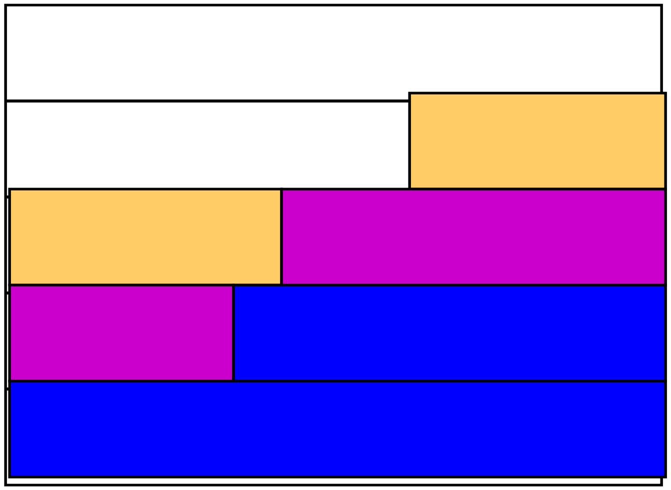


- A single loss will affect (at least) a slice of a frame
  - 16 pixels vertically, entire image horizontally

- More slice_start_codes:
  - Better quality with packet losses (less data lost)
  - Worst quality without packet losses (bits wasted)

Packetization vs. Slices

- Ideally packets should be aligned with slices so that one lost packet only affect one slice
- A packet should not cross video frames
- Using equal pixel size slices resulting in variable length (in bytes) packets



“Uncompressed” video

Bit error

With slice structure, a bit error only affect one slice, but can render remaining bits in the same slice undecodable!

Reversible Variable Length Coding

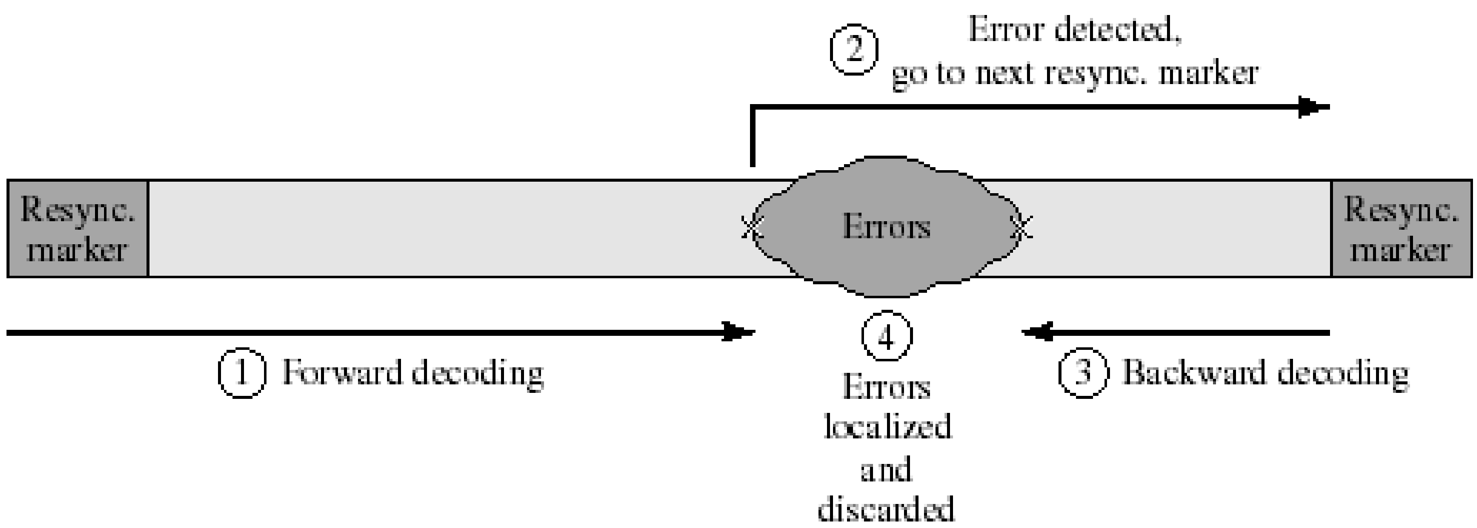


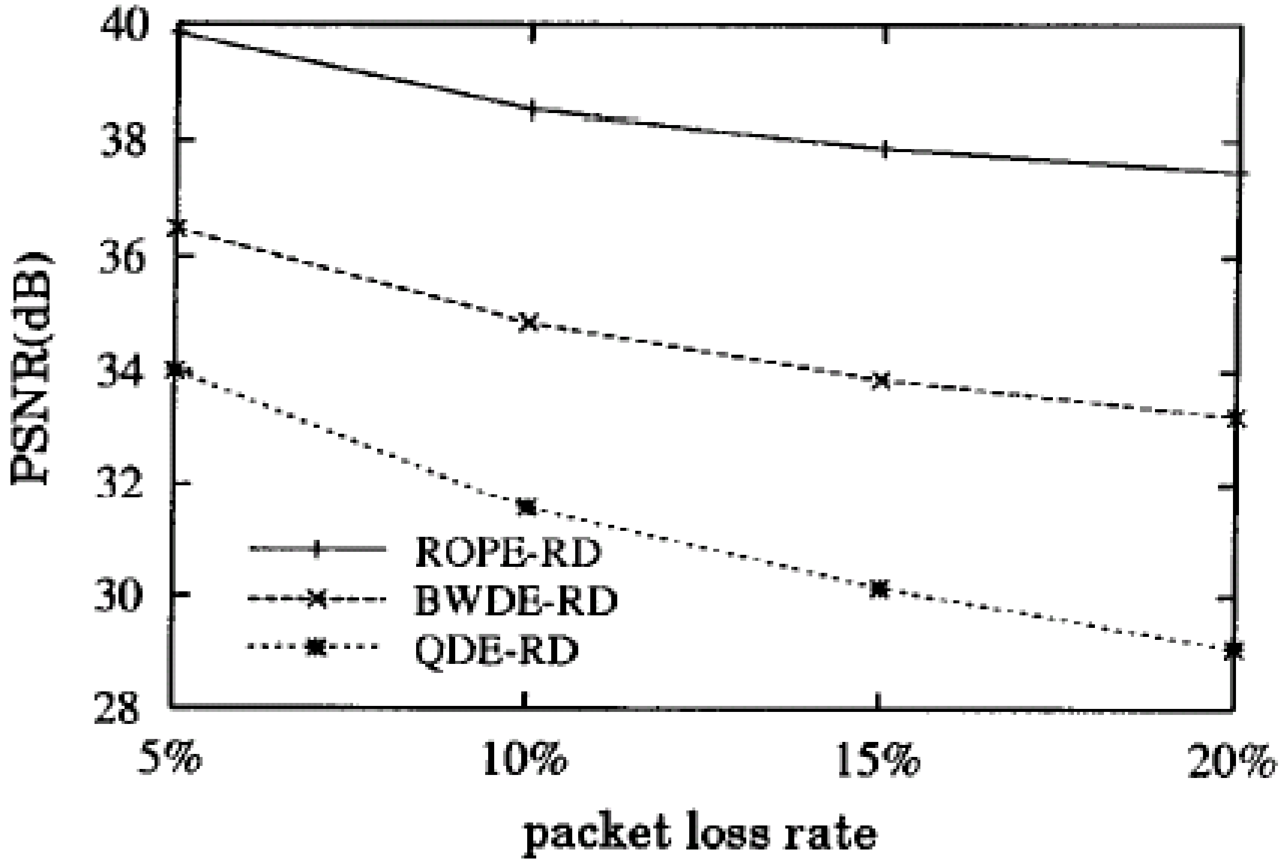
Figure 14.7 RVLC codewords can be parsed in both the forward and backward direction, making it possible to recover more data from a corrupted data stream.

Effective primarily for bit errors. Very small additional rate.
Increased decoder complexity (when implemented).

RD optimized mode decision considering packet loss

- Goal: How to best compress video when it will be transmitted across an unreliable network
  - Should this block be sent as an I-block or P-block?
  - Minimize the decoder distortion due to compression AND loss, subject to total rate
- *At encoder*, for each coding option (I or P block)
  - Compute rate
  - Compute joint distortion *at decoder*, for encoding and packet loss
- Basic principle can be extended in many ways
  - Include channel redundancy due to FEC/retransmission, Scalable coding, Multiple Description Coding, etc

Results: RD optimized video coding

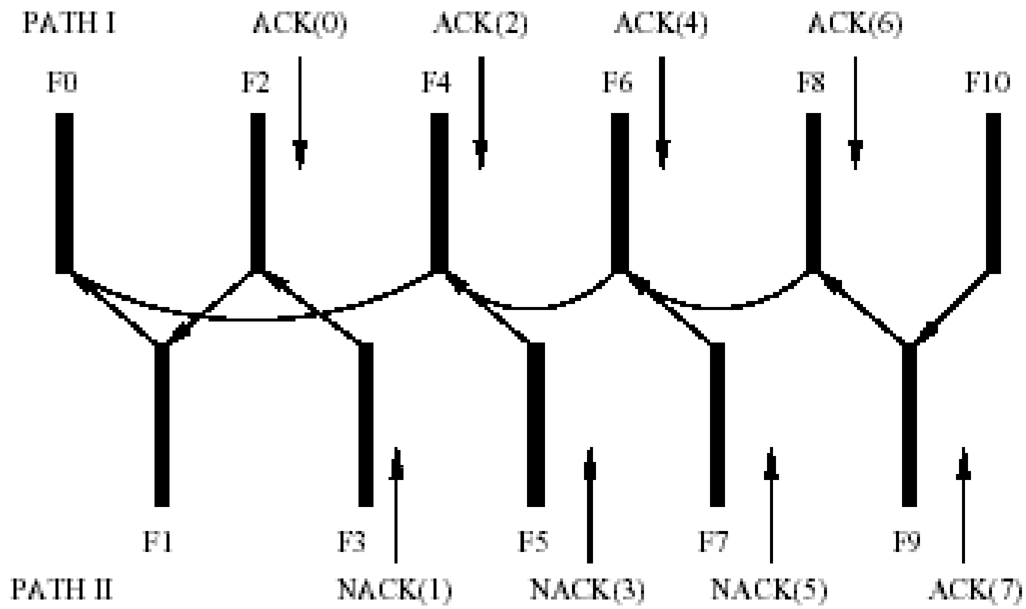


- Zhang, Regunathan, and Rose, “Video coding with optimal inter/intra-mode switching for packet loss resilience”, IEEE JSAC, June 2000, 18(6):966–76

Encoder-Decoder Interactive Error Control

- Coding parameter adaptation based on channel conditions
  - Change intra-rate based on average loss rates
- Reference picture selection (part of H.263/MPEG-4 standard)
  - Following a damaged frame (feedback info from receiver), use undamaged previous frame as reference frame for temporal prediction
- Error tracking
  - Determine which MBs are affected following a lost MB (feedback info), avoid using those MBs as reference pixels
- Requires a feedback channel, not necessarily involving extra coding delay

Reference Picture Selection



Even/odd frames sent on separate paths. Predict damaged frames based on NACK on each path, and use undamaged frames as reference pictures.

Compatible with the RPS option in H.263+.

Layered Coding with Unequal Error Protection

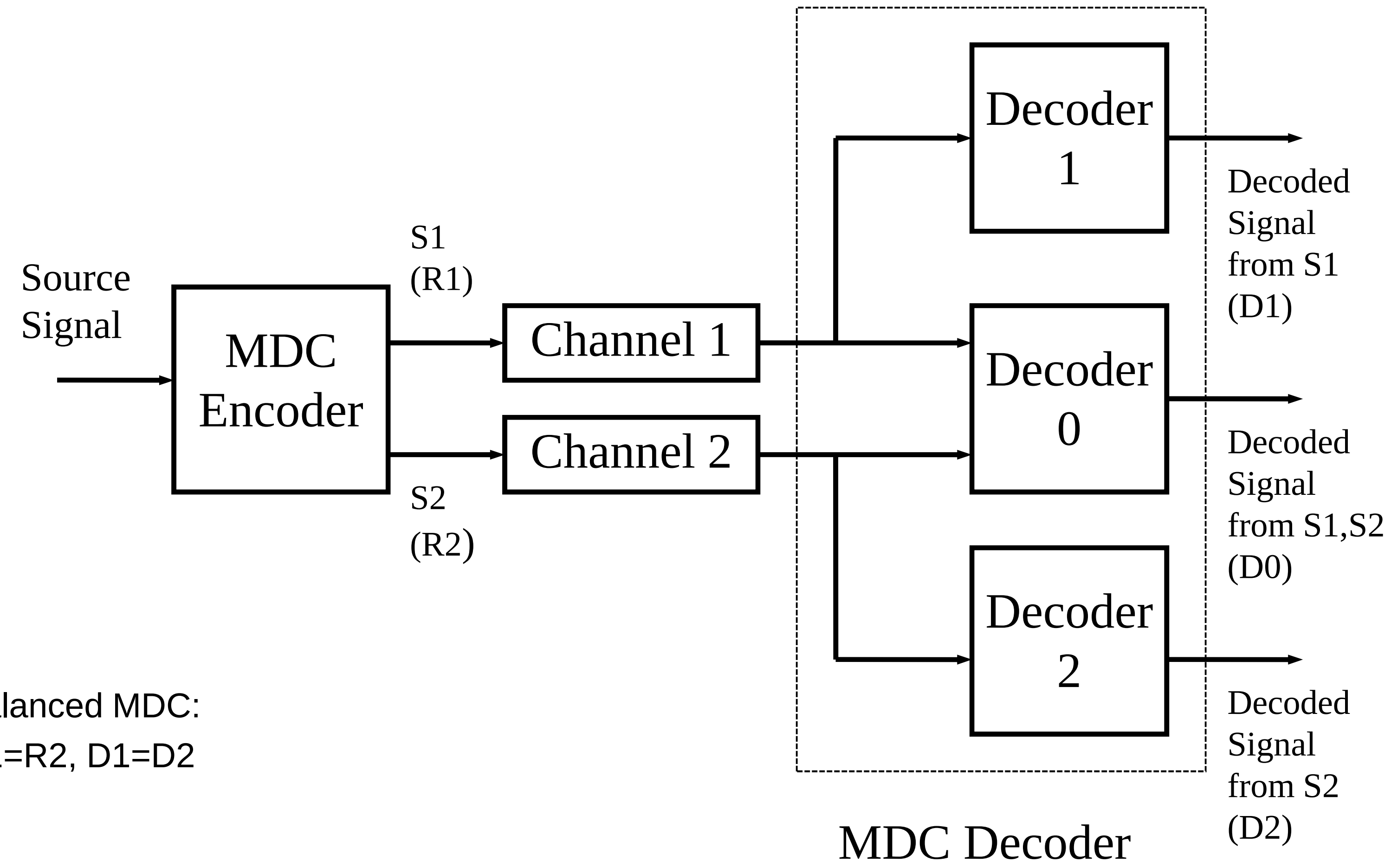
- LC+UEP
  - Base layer provides acceptable quality, enhancement layer refines the quality
  - Base layer stream is delivered through a reliable channel (by using ARQ and strong FEC or better transmission path)
  - Good for a network with differentiated service (Do NOT exist today over Internet, may become part of emerging wireless standards)

- Problems:
  - Any error in the base layer causes severe degradation
  - Repetitive ARQ may incur unacceptable delay, strong FEC may be too complex or cause extra delay
  - The enhancement layer is useless by itself
  - The increased bit-rate from scalable coding may be too high

Multiple Description Coding

- Assumptions:
  - Multiple channels between source and destination
  - Independent error and failure events
  - Probability that all channels fail simultaneously is low
  - Reasonable assumptions for the Internet and wireless networks, provided data are properly packetized and interleaved
- MDC: Generate multiple correlated descriptions
  - Any description provides low but acceptable quality
  - Additional received descriptions provide incremental improvements
  - No retransmission required -> low delay
  - However: correlation → reduced coding efficiency
- Design goal:
  - maximize the robustness to channel errors at a permissible level of redundancy

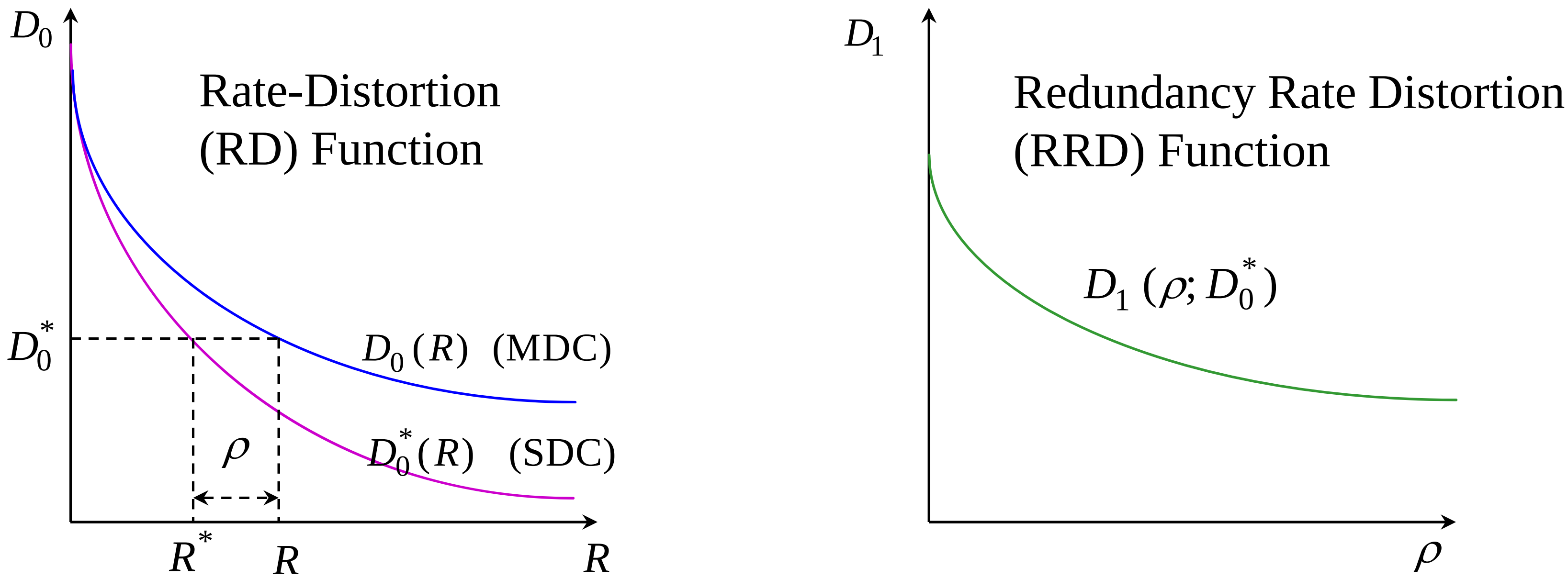
Generic Two Description Coder



Balanced MDC:

R1=R2, D1=D2

Redundancy Rate Distortion



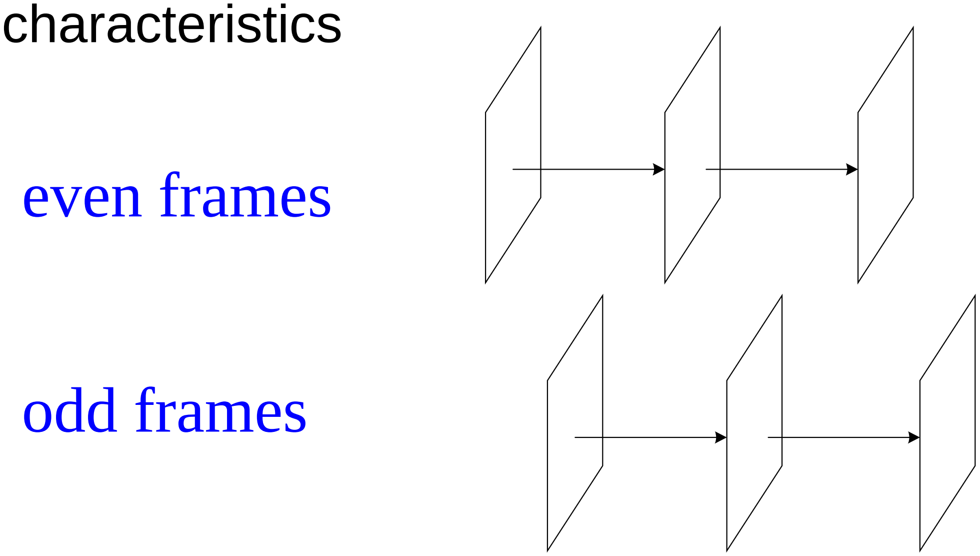
- Design criteria for MD coders
  - Minimize D_1 for a given ρ , for fixed R^* or D_0^* (minimizing the average distortion given channel loss rates, for given total rate)
  - Can easily vary the ρ vs. D_1 trade-off to match network conditions

Challenge in Designing MD Video Coder

- To achieve high coding efficiency, the encoder should retain the temporal prediction loop
- Prediction strategies are key to control trade-off between added redundancy and reduced compression efficiency
  - Predict from two-description reconstruction, or one?
- Prediction based on two-description reconstruction
  - Higher prediction efficiency
  - Mismatch problem at the decoder
- Prediction based on single-description reconstruction
  - Lower prediction efficiency
  - No mismatch problem
- One design strategy
  - Predict based on two-description reconstruction, but explicitly code the mismatch error

Video Redundancy Coding in H.263+

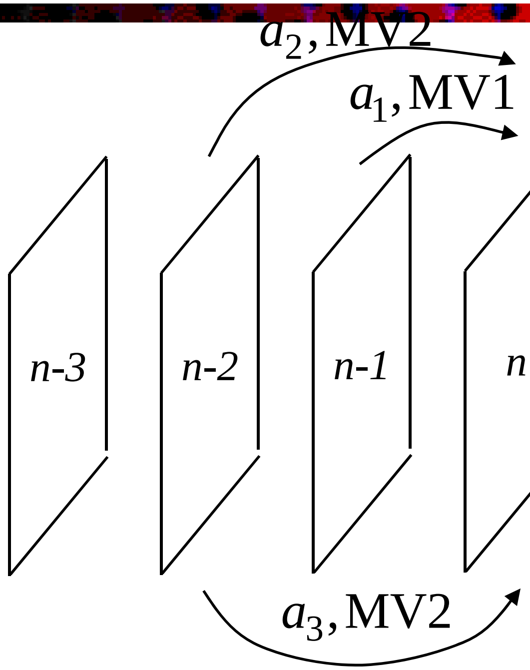
- Coding even frames and odd frames as separate threads
  - High redundancy (~30%) due to reduced prediction gain because of longer distance between frames
  - Hard to vary the redundancy based on channel loss characteristics



Multiple Description Motion Compensation (Wang and Lin, 2001)

- A description contains even (or odd) frames only, but each frame is predicted (central predictor) from both even and odd past frames
- Code the central prediction error
  - sufficient if both descriptions are received
- To avoid mismatch, a side predictor for even frames predicts only from the past even frame, and the mismatch signal (difference between central and side prediction) is also coded
- The predictors and the mismatch error quantizer control the redundancy of the coder, and can be designed based on the channel loss characteristics

Special Case: Two-Tap Predictor



Central predictor: $\hat{\psi}_0(n) = a_1\tilde{\psi}_0(n-1) + a_2\tilde{\psi}_0(n-2)$

Central prediction error: $e_0(n) = \psi(n) - \hat{\psi}_0(n) \rightarrow \tilde{e}_0(n)$

Side predictor: $\hat{\psi}_1(n) = a_3\tilde{\psi}_1(n-2)$

Mismatch error: $e_1(n) = \hat{\psi}_0(n) - \hat{\psi}_1(n) - q_0(n) \rightarrow \tilde{e}_1(n)$

Send: $\tilde{e}_0(n), \tilde{e}_1(n)$, MV1, MV2

Non-leaky predictor: $a_1 + a_2 = 1, a_3 = 1$

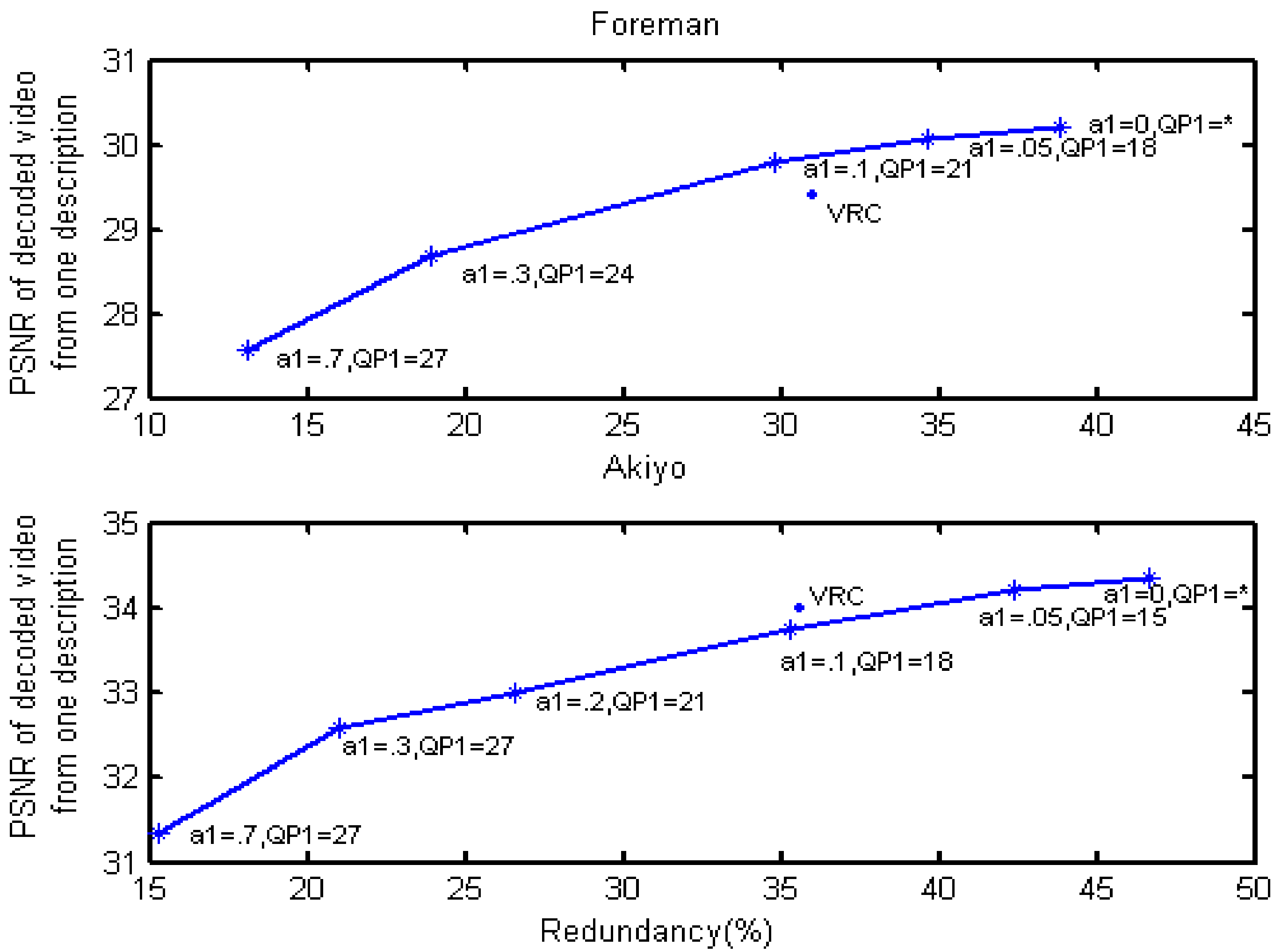
If both descriptions received (have both $\psi_0(n-1), \psi_0(n-2)$)

$$\psi_0(n) = \hat{\psi}_0(n) + \tilde{e}_0(n) = \psi(n) + q_0(n)$$

If one description is received (have only $\psi_1(n-2)$)

$$\psi_1(n) = \hat{\psi}_1(n) + \tilde{e}_0(n) + \tilde{e}_1(n) = \psi(n) + q_1(n)$$

RRD Performance of VRC and MDMC



Performance in Packet Lossy Networks

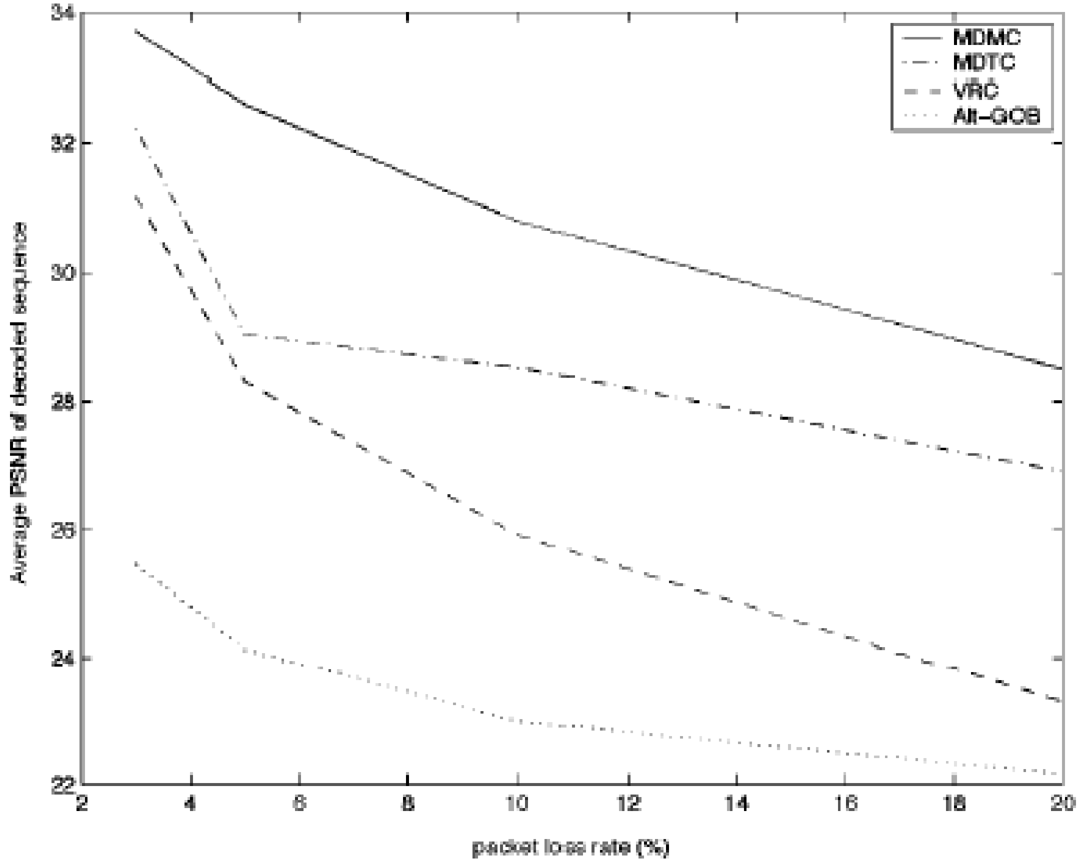


Fig. 13. PSNR of decoded sequences in different packet loss rates. Foreman, 7.5 fps, 144 kbps, two packets per frame.

Sample Reconstructed Frames
(10% Random Packet Loss, MDMC on top, VRC on bottom)

Decoder Error Concealment

- With proper error-resilience tools, packet loss typically lead to the loss of an isolated segment of a frame
- The lost region can be “recovered” based on the received regions by spatial/temporal interpolation → Error concealment
- Decoder optimization issue, not part of video coding standard!
- Decoders on the market differ in their error concealment capabilities

Error Concealment Techniques

- Basic idea:
  - Recover damaged regions by interpolating from surrounding (in the same frame and in nearby frames) regions
- Motion-compensated temporal interpolation
  - Replace damaged MB by its corresponding MB in reference frame
  - If the MV is also lost, need to estimate the MV first. One approach: copy the MV of the MB above
  - Simple and quite effective, if the data were appropriately partitioned
- Spatial interpolation
  - Estimate damaged MB from received neighboring MBs
  - Maximally smooth recovery (Wang and Zhu, 1993) estimates missing DCT coefficients so a combination of spatial and temporal smoothness measures is maximized
- Large amount of literature! (See textbook)

Sample Error Concealment Results

Without concealment With concealment

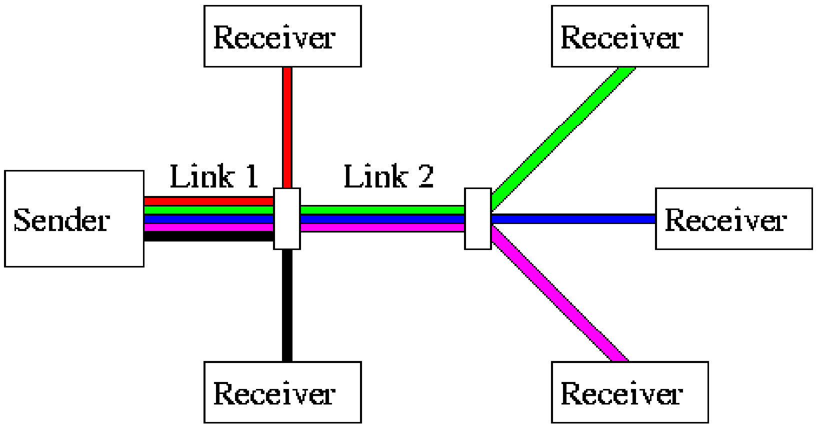
Video Streaming

- Different types of streaming services
- Receiver heterogeneity
- Network dynamics
  - rate adaptation (QoS control)
- Streaming protocols
- Delivery architectures

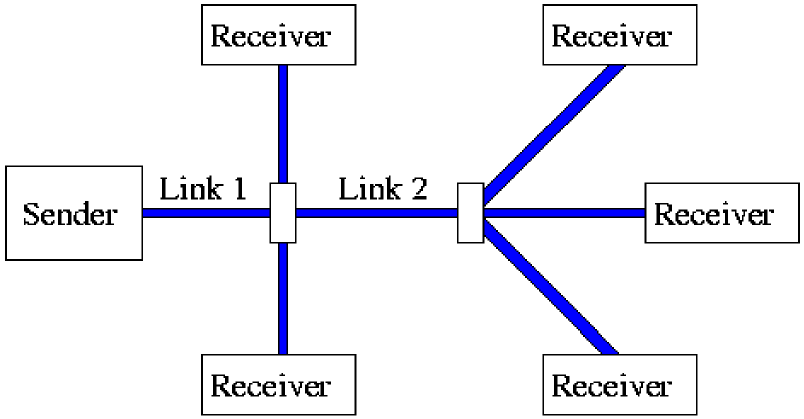
Categorization

- Live streaming
  - Sender captures live event and compresses in real time
  - Receiver streams video with a small time shift (tens of seconds), no fast forward
- Video-on-demand (non live streaming)
  - Videos are pre-compressed off line
  - Receivers can stream at different times (asynchronously), can have random access (fast forward, remind, pause, etc.)
- Unicast (point to point)
  - Server send to each receiver separately
- Multicast (one to many)
  - Server send the same video to multiple receivers

Unicast vs. Multicast



Unicast



Multicast

Pros and Cons?
How can multicast accommodate receivers with different down-link capacity?

Challenges

- Receivers of the same video differ in sustainable bandwidth and decoding/display capability
- The end-to-end throughput (sustainable bandwidth) and delay between server and each receiver changes in time
- Popular contents may be requested by many receivers
- How to handle residual packet losses (covered already)

Challenge 1: Receiver Inhomogeneity

- Receivers of the same video differ in sustainable bandwidth and decoding/display capability
- Possible solutions
  - Simulcast: code the same video to multiple versions with different rates
  - Layered / scalable video: code a video into multiple layers, adapt the number of layers to send based on the receiver bandwidth / display capability

Challenge 2: Network Dynamics

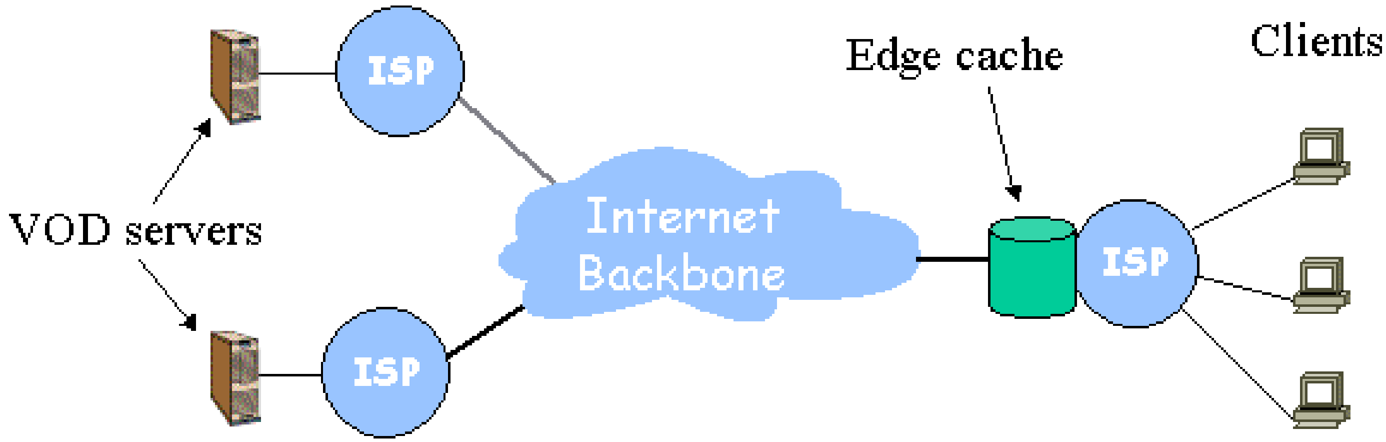
- The end-to-end throughput (sustainable bandwidth) and delay changes in time
  - Wireless link is inherently time varying due to fading/shadowing and mobility
  - Backbone network can suffer from congestion
- Require video rate adaptation and adaptive error control !

Video Rate Adaptation

- How to estimate sustainable rate
  - Sender estimates sustainable rate based on feedback
  - Receiver estimates based on receiving packet statistics and inform the sender the desired video rate (HTTP streaming)
- How to adapt video to the desired rate
  - Encoder rate control (only appropriate for live streaming and unicast)
  - Switch among multiple rate versions (adaptive HTTP streaming!)
  - Layered coding and send only a subset of layers

Challenge 3: Content Caching

- Popular contents may be requested by many receivers
- Replicate the content at multiple edge servers (CDN)
- ISP/access point may cache recently delivered packets so that other nodes served by this ISP can reuse them

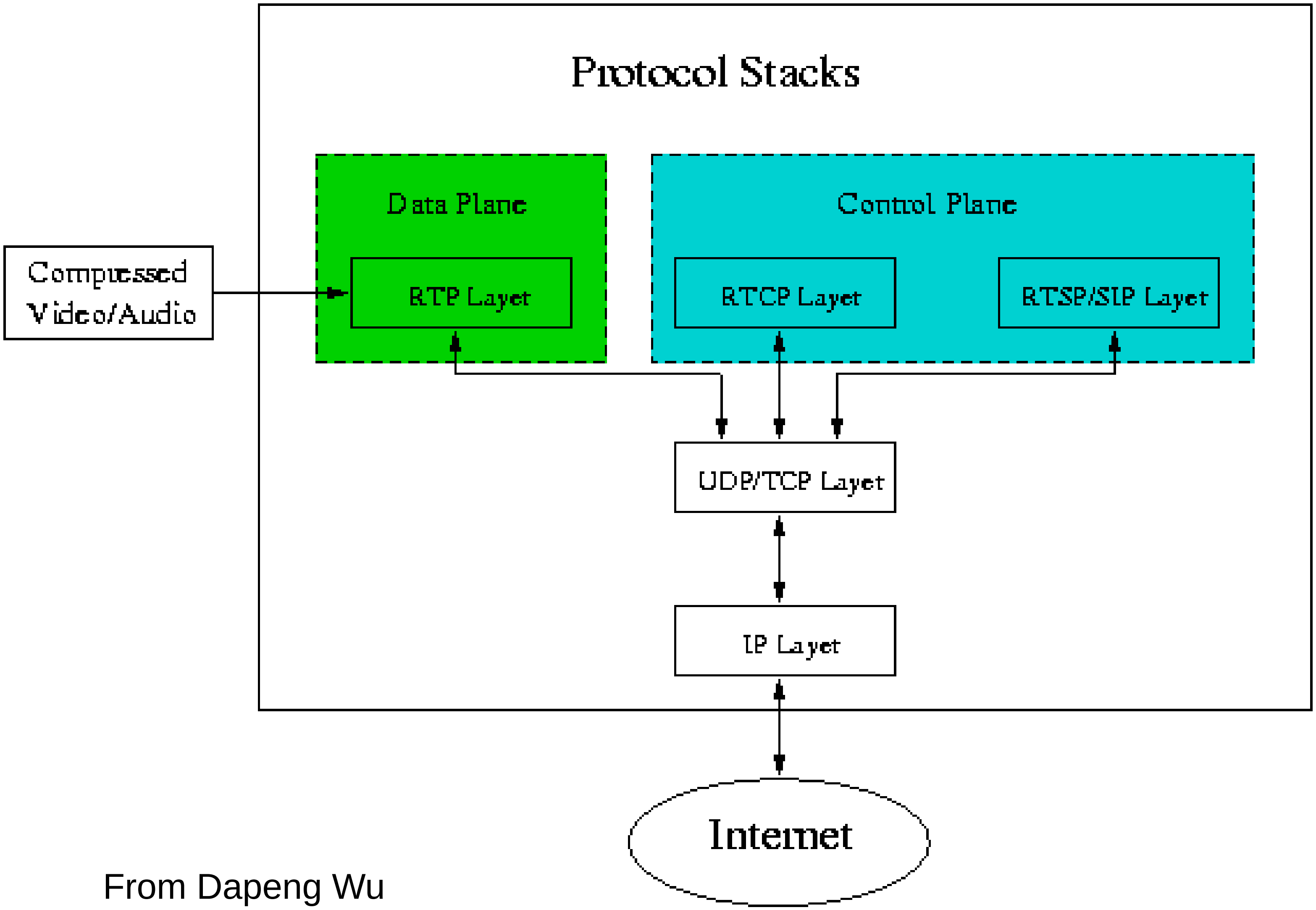


From Dapeng Wu

VOD: video-on-demand
ISP: Internet service provider

Related Protocols

- Network-layer protocol: Internet Protocol (IP)
- Transport protocol:
  - Lower layer: UDP & TCP
  - Upper layer: Real-time Transport Protocol (RTP) & Real-Time Control Protocol (RTCP)
- Session control protocol (application layer):
  - Real-Time Streaming Protocol (RTSP): RealPlayer
  - Session Initiation Protocol (SIP): Microsoft Windows MediaPlayer; Internet telephony
  - HTTP streaming



Application layer

Transport layer

Network layer

Link layer

Physical layer

From Dapeng Wu

Delivery Architecture

- Server-based
  - Using client-server model
  - Each receiver get her video from the server
  - Multiple servers form an overlay network (content delivery network or CDN)
- Peer-to-peer
  - Server send to some receivers
  - Those receivers help to deliver to other receivers
- Hybrid: Peer-assisted

Summary

- What are different types of video applications and their requirements
- What causes packet loss and delay
- What causes error propagation in video coding
- Transport level error control
  - Basic concept of channel coding (FEC)
  - Bit level and packet level FEC
  - Retransmission is effective within the delay constraint
- Error resilient encoding
  - Trade off coding efficiency for error resilience
  - Synchronization markers, slices, I-frames
  - Reversible variable length coding
  - Some techniques are only useful for bit-error dominated channels

Summary (Cnt’d)

- Encoder-decoder-network interactive error control
  - Adaptation of reference frames, intra-blocks
  - Requires feedback info, may not be available
- Error concealment
  - Does not involve extra redundancy, motion-compensated temporal concealment is simple and yet offers visible improvements
- Layered coding and unequal error protection
- Multiple description coding and multipath transport
- Choice of technique(s) depends on underlying application and network
- Video streaming fundamentals

References

- Y. Wang and Q. Zhu, “Error control in video communications – A review,” Proc. IEEE, 1998
- Y. Wang, A. R. Reibman, and S. Lin, “Multiple description coding for video delivery”, invited paper, Proc. IEEE, Jan. 2005.
- Y. Wang, J. Ostermann, Y.-Q. Zhang, Video processing and communications, Prentice Hall, 2002. Chap. 14.

Homework

- Reading assignment
  - Y. Wang, J. Ostermann, Y.-Q. Zhang, Video processing and communications, Prentice Hall, 2002. Chap. 14.
- Homeworks
  - Prob. 14.1-14.4, 14.9, 14.11, 14.12, 14.13