Video Communication and Video Streaming II: Error Resilient Video Coding

John G. Apostolopoulos
Streaming Media Systems Group
Hewlett-Packard Laboratories

Outline

- Review of video streaming from last lecture
- Error concealment
- Error resilient video coding
  - Two basic error-induced problems:
    1. Bitstream synchronization
    2. Incorrect state and error propagation
  - Multiple description video coding
- Appendix: Protocols for media streaming over the Internet

Review of Video Streaming over the Internet

- Problem: Internet only offers best-effort service
- No guarantees on:
  - Bandwidth
  - Delay jitter
  - Loss rates
- Specifically, these characteristics are *unknown* and *dynamic*
- Goal: Design a system to reliably delivery high-quality video over the Internet

Problems in Video Streaming over the Internet

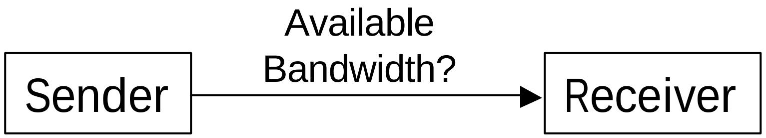
Problems to be addressed include *unknown* and *dynamic*:

- Bandwidth problem → Rate control
- Delay jitter → Playout buffer
- Loss → Error control

Problems in Video Streaming over the Internet (cont.)

Problems to be addressed include *unknown* and *dynamic*:

- *Bandwidth problem ® Rate control*


Sender

Available Bandwidth?

Receiver

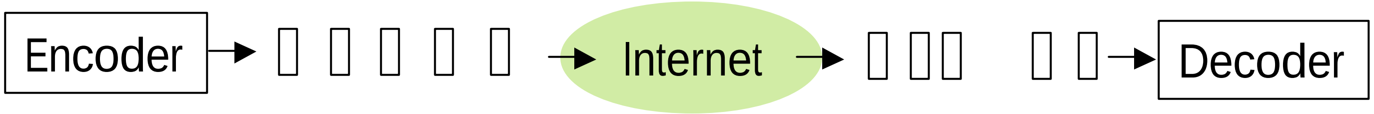
1. *Estimate the available bandwidth*
   - Probe-based or model-based methods
2. *Match video rate to available bandwidth*
   - Perform at sender or receiver

- Delay jitter
- Loss

Problems in Video Streaming over the Internet (cont.)

Problems to be addressed include *unknown* and *dynamic*:

- Bandwidth problem
- *Delay jitter ® Playout buffer*

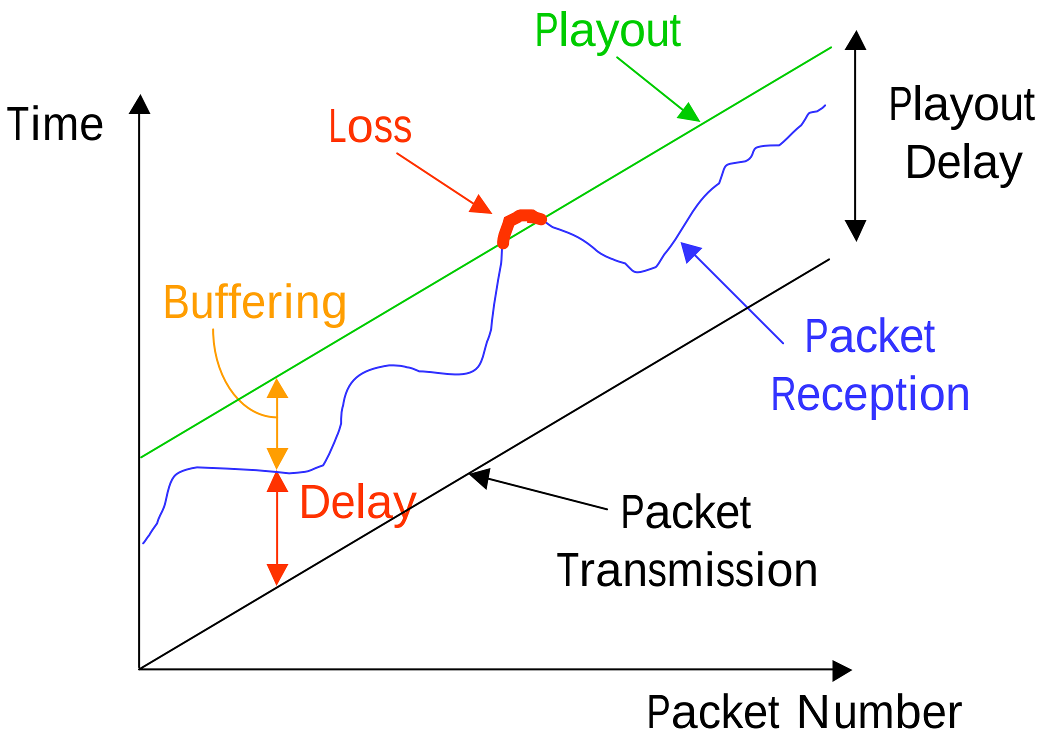


Encoder → Internet → Decoder

  - If video data arrives late it is useless
  - *Add playout buffer* to (partially) compensate for delay jitter
- Loss

Reliable Video

Overcoming Delay Jitter: Playout Buffer



Problems in Video Streaming over the Internet

Problems to be addressed include *unknown* and *dynamic*:

- Bandwidth problem
- Delay jitter
- *Loss ® Error control*
  - Forward Error Correction (FEC)
  - Retransmission
  - *Error concealment*
  - *Error-resilient coding*

Today's Lecture

Types of Errors

Application	Error Characteristics
Videophone over PSTN	Few bit errors and packet loss
Videoconferencing over ISDN	Practically error free (BER 10^{-8})
Digital Television	Almost error free
Video streaming over the Internet	Packet loss of 0-30%
Wireless video (cell phone)	BER up to 10^{-3}, Burst errors

(The above are rough numbers. BER = bit error rate)

Error Control

- Goal of error control:
  - Overcome the effect of errors, e.g. packet loss on a packet network or bit or burst errors on a wireless link
- Types of error control:

Channel Coding

  - Forward Error Correction (FEC)
  - Retransmission

Source Coding

  - Error concealment
  - Error-resilient video coding

Error Control (cont.)

- Last lecture:
  - – Examined mechanisms designed for *reliable data deliver*:
    1. Forward error correction (FEC)
    2. Retransmission
  - – Adapted these techniques to the unique properties of video, e.g. *unequal importance and delay constraints*
- Today: Examine *video-centric approaches*

  → Approaches applicable to video, but not to general data types

Error Control (cont.)

- *Error concealment*
  - Recover (conceal) any errors that may occur
- *Error-resilient video coding*
  - Design the compression algorithm and the compressed bitstream so that it is resilient to errors

Outline

- Review of video streaming from last lecture
- → Error concealment
- Error resilient video coding
  - Two basic error-induced problems:
    1. Bitstream synchronization
    2. Incorrect state and error propagation
  - Multiple description video coding
- Appendix: Protocols for media streaming over the Internet

Error Concealment

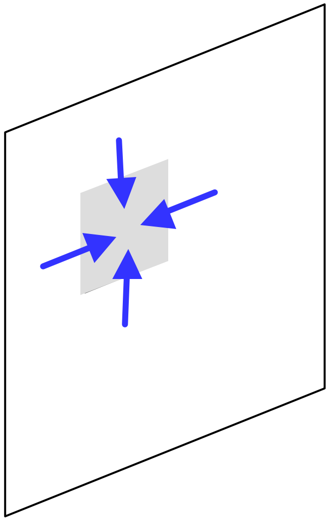
- Problem: Transmission errors may result in lost information
- Goal: Estimate the lost information in order to conceal the fact that an error has occurred
- Error concealment is performed at the decoder
- Observation: Video exhibits a significant amount of *correlation along the spatial and temporal dimensions*
- Basic approach: Perform some form of *spatial/temporal interpolation* to estimate the lost information from correctly received data

Error Concealment (cont.)

- Consider the case where a single macroblock (16x16 block of pixels) is lost
- Three examples of error concealment:
  1. Spatial interpolation
  2. Temporal interpolation (freeze frame)
  3. Motion-compensated temporal interpolation

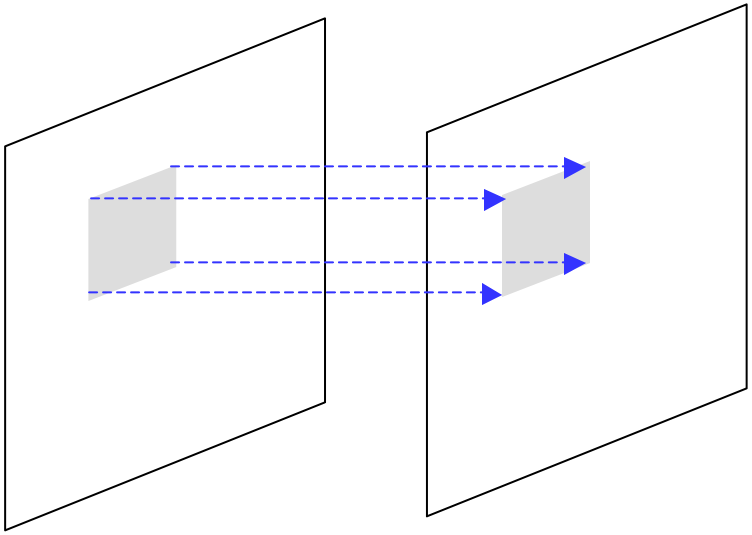
Error Concealment (cont.)

- Spatial interpolation:
  - Estimate missing pixels by smoothly extrapolating surrounding pixels
  - Correctly recovering missing pixels is extremely difficult, however even correctly estimating the DC (average) value is very helpful



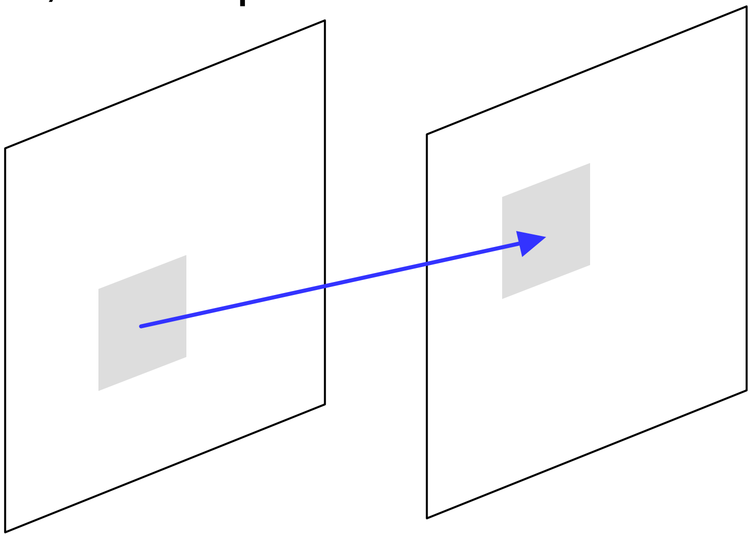
Error Concealment (cont.)

- Temporal interpolation (freeze frame):
  - – Copy the pixels at the same spatial location in the previous frame (freeze frame)
  - – Effective when there is no motion, potential problems when there is motion



Error Concealment (cont.)

- 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



Error Concealment (cont.)

Comments:

- Usually lose more than a single macroblock
  - E.g. lose a row of macroblocks or an entire frame
- Apply combination of spatial & temporal interpolation
- Motion-compensated temporal interpolation usually provides the best concealment (assuming accurate MV)
- Many sophisticated algorithms exist
  - Can be formulated as a signal recovery / inverse problem
- Error concealment is performed at the decoder
  - New error concealment algorithms can be incorporated as *standard-compatible enhancements* to conventional decoders

Outline

- Review of video streaming from last lecture
- Error concealment
- → Error resilient video coding
  - – Two basic error-induced problems:
    1. Bitstream synchronization
    2. Incorrect state and error propagation
  - – Multiple description video coding
- Appendix: Protocols for media streaming over the Internet

Error-Resilient Video Coding

- Goal: Design the video compression algorithm and the compressed bitstream so that it is resilient to errors
- Why: Compressed video is highly vulnerable to errors
- Examples:
  - Error in VLC
  - Error in prediction
- Next: Overview of error-resilient video compression
  - Basic problems introduced by errors
  - Methods to overcome these problems

Basic Error-Induced Problems

Assuming conventional MPEG-like system: MC-prediction, Block-DCT, runlength and Huffman coding

Two basic classes of problems:

- Bitstream synchronization: Decoder does not know what bits correspond to what parameters
  - E.g. error in Huffman codeword
- Incorrect state and 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

Outline

- Review of video streaming from last lecture
- Error concealment
- Error resilient video coding
  - Two basic error-induced problems:
    1. Bitstream synchronization
    2. Incorrect state and error propagation
  - Multiple description video coding
- Appendix: Protocols for media streaming over the Internet

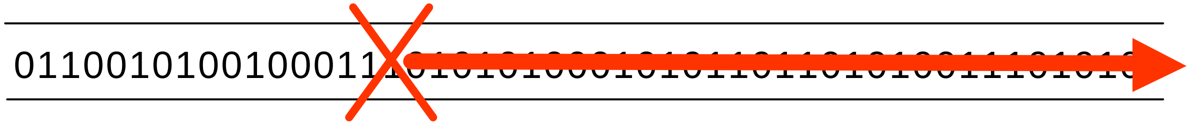
Basic Error-Induced Problems: Bitstream Synchronization

Bitstream synchronization: Decoder does not know what bits correspond to what parameters

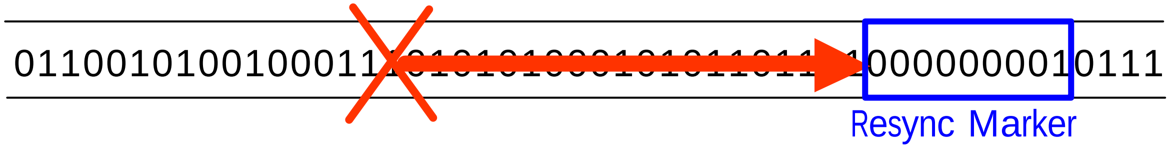
- Example: Error in Huffman codeword or other variable length code (VLC)
  - Single bit error can lead to *significant subsequent loss*
- Notes:
  - Fixed length codes (FLC) do not have this problem since the beginning and end of codeword is know, I.e. losses are limited to a single codeword
  - However, FLC's do not provide good compression efficiency
  - VLC's and other forms of entropy coding are widely used

Overcoming Problem: Bitstream Synchronization

- Bitstream sync problem: Decoder does not know what bits correspond to what parameters



- Goal: *Mechanisms to enable resynchronization*
- Approaches:
  1) *Use resync markers*
     a) Marker distinct from all codewords, concatenations of codewords, and minor perturbations of concatenated codewords
     b) Include information after marker to restart decoding



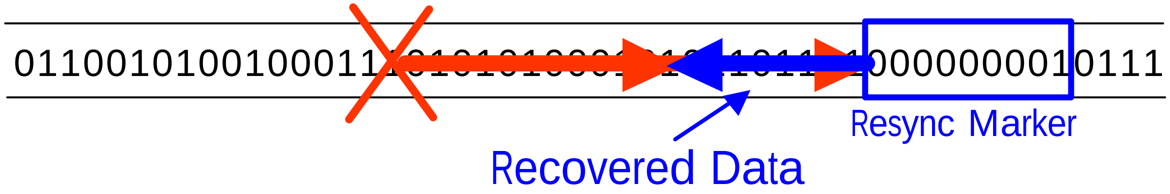
Overcoming Problem: Bitstream Synchronization (cont.)

- Use resync markers (cont.)
  - Place resync markers at *strategic locations* in bitstream, e.g. beginning of frame, slice, etc. (MPEG-1/2, H.261/3)
    - Resync every fixed # of blocks, variable # of bits
    - Active areas are more likely to be corrupted (bad!)
  - Place resync markers *periodically* (MPEG-4)
    - Resync every fixed # of bits, variable # of blocks
    - Advantages:
      - Active areas less likely to be corrupted
      - Simplifies search for resync markers
      - Supports network packetization

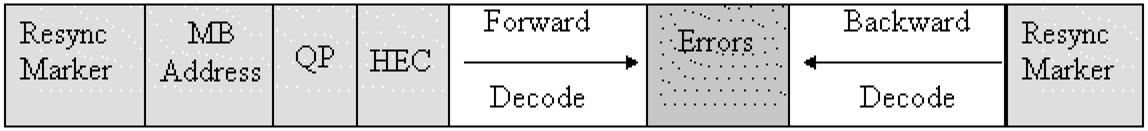
Overcoming Problem:
Bitstream Synchronization (cont.)

Approaches (cont):

- *Reversible variable length codes (RVLC):*
  - – Conventional VLC's are uniquely decodable only in forward direction
  - – RVLC's also have the property that they can be uniquely decodable in the *backward direction*
  - – Use: If an error is detected, jump to the next resync and start decoding backwards, enabling partial recovery of data (otherwise would be discarded)



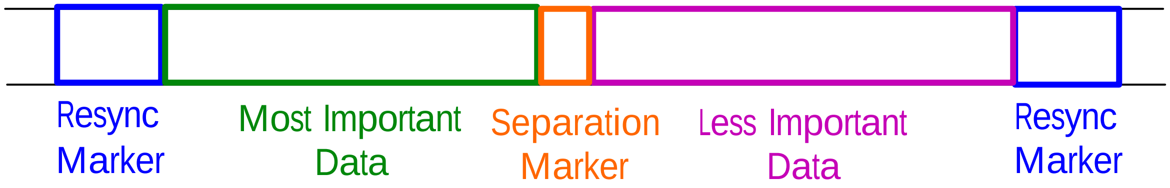
MPEG-4 Syntax



Overcoming Problem: Bitstream Synchronization (cont.)

Approaches:

- *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



Overcoming Bitstream Synchronization Problem: Summary

- Approaches discussed:
  - Resync markers at strategic locations
  - Resync markers at periodic locations
  - Reversible VLC’s (RVLC’s)
  - Data partitioning
- Basic ideas:
  - *Isolate (localize) corrupted information*
  - *Enable fast resynchronization*

Application-Aware Packetization: Application Level Framing (ALF)

- Goal: To combat packet loss, should *design the packet payload to minimize the effect of loss*
- Application knows best how to do this [Clark, Tennenhouse]
- For example:
  - Video encoder may know the packet size
  - Knows the boundaries of possible lost data
  - Design payload to provide bitstream resynchronization at the beginning of each packet
- *RTP and MPEG-4 and H.263+ support these features*
  - Bitstream synchronization problem can often be overcome for video streaming over the Internet
  - Error propagation is the major obstacle

Outline

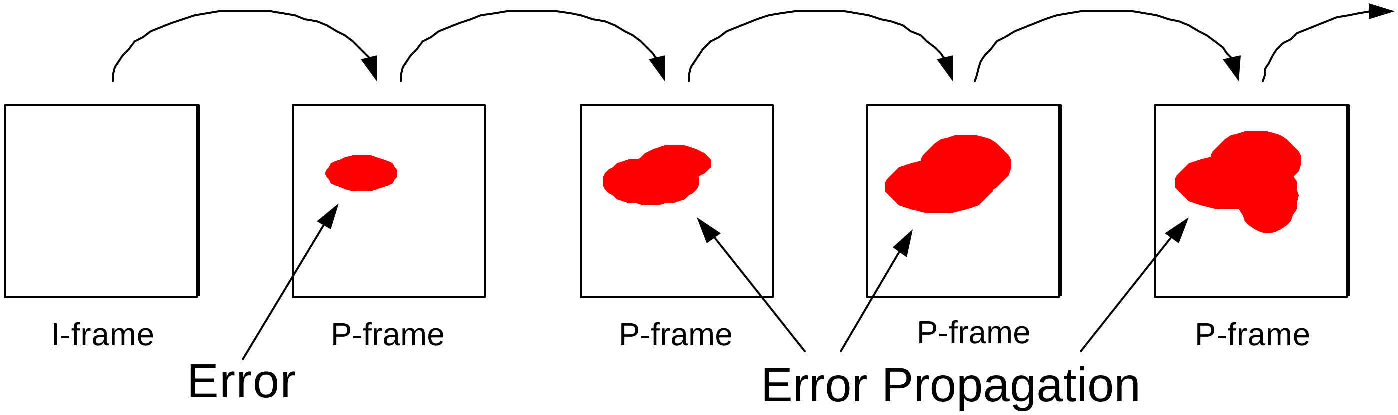
- Review of video streaming from last lecture
- Error concealment
- Error resilient video coding
  - Two basic error-induced problems:
    1. Bitstream synchronization
    2. Incorrect state and error propagation
  - Multiple description video coding
- Appendix: Protocols for media streaming over the Internet

Reliable Video

Basic Error-Induced Problems: Incorrect State and Error Propagation

Incorrect state and error propagation at decoder:

- An error causes the reconstructed frame (state) at the decoder to be incorrect
- Decoder's state is different from encoder's, leading to incorrect (mismatched) predictions and often significant error propagation



Overcoming Incorrect State and Error Propagation Problem

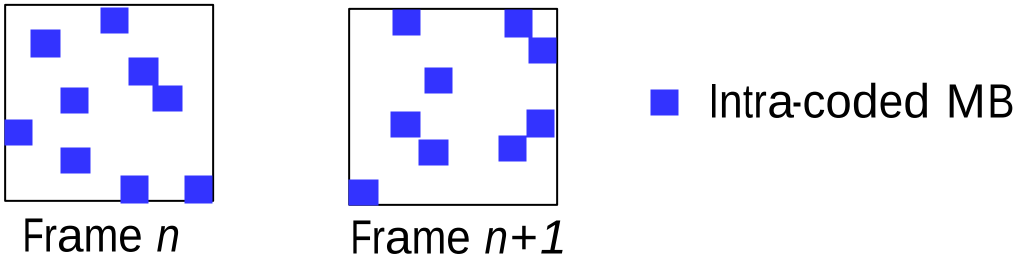
- An error causes the reconstructed frame at the decoder to be incorrect, and often leads to significant error propagation
- Goal: *Limit effect of error propagation*

  (e.g. reinitialize prediction loop)
- Approaches:
  - *All I-frame*
    - Eliminates error propagation, but poor compression
  - *Periodic I-frames*, e.g. MPEG GOP
    - Example: I-frame every 15 frames
    - Limits error propagation to one GOP
    - Compression is still relatively poor (inappropriate for very low bit rate video, e.g. wireless video or video over the Internet)

Reliable Video

Overcoming Incorrect State and Error Propagation Problem (cont.)

- Approaches (cont.):
  - *Partial intra-coding of each frame*
    - Partial: Individual macroblocks (MBs) are intra-coding
  1. *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



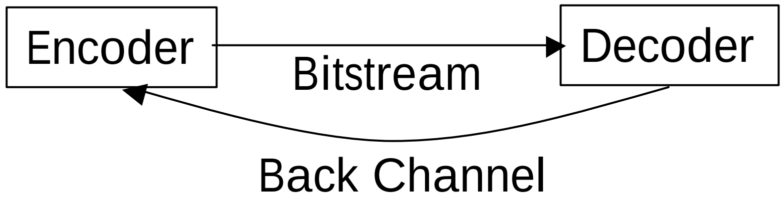
  2. *Preemptive intra-coding* based on estimated vulnerability to errors [Hinds, Pappas, Lim]
     - Can be used to optimize intra/inter mode decisions based on a channel loss model

Overcoming Incorrect State and Error Propagation Problem (cont.)

- Advantages of intra-coding approaches:
  - Limits the effect of error propagation
  - Sophistication at encoder, decoder is simple
- Disadvantages:
  - High bit rate required for intra coding limits its use
  - Optimal use depends on accurate knowledge of channel characteristics

Overcoming Error Propagation: Point-to-point with back-channel

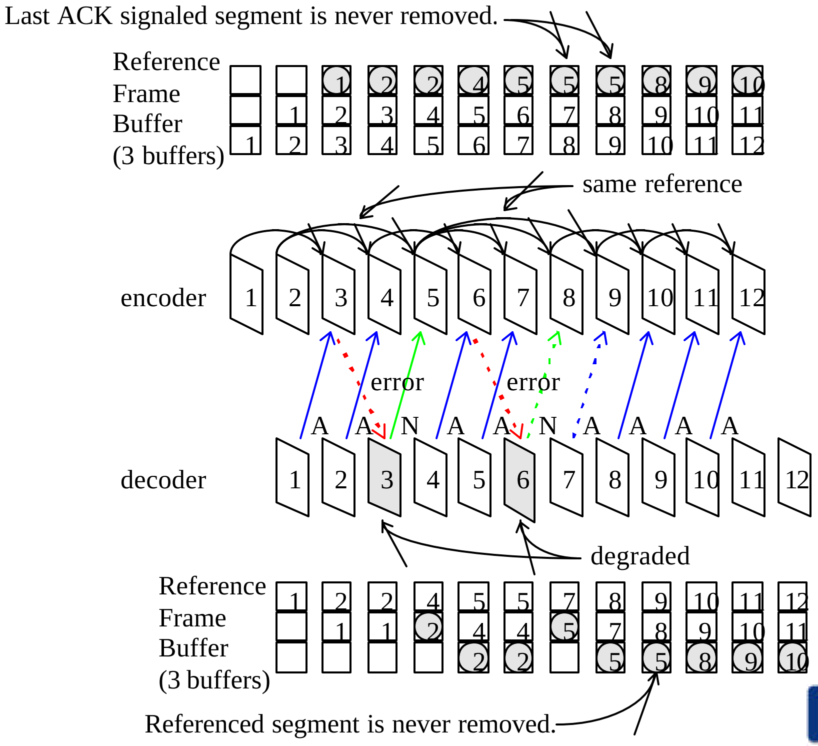
- *Special case:* Point-to-point communication with a back-channel
  - E.g. video phone, video conferencing



- *Decoder detects error and can tell the encoder:*
  - *Reinitialize prediction* (use I-frame)
    - Simple, straightforward
    - However, requires higher bit rate for intra coding
  - *Which frame to use as reference* for next prediction
    - Use inter coding with previous clean frames → better compression
    - 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)
    - Two modes of operations: ACK, NACK

Overcoming Error Propagation: Point-to-point with back-channel (cont.)

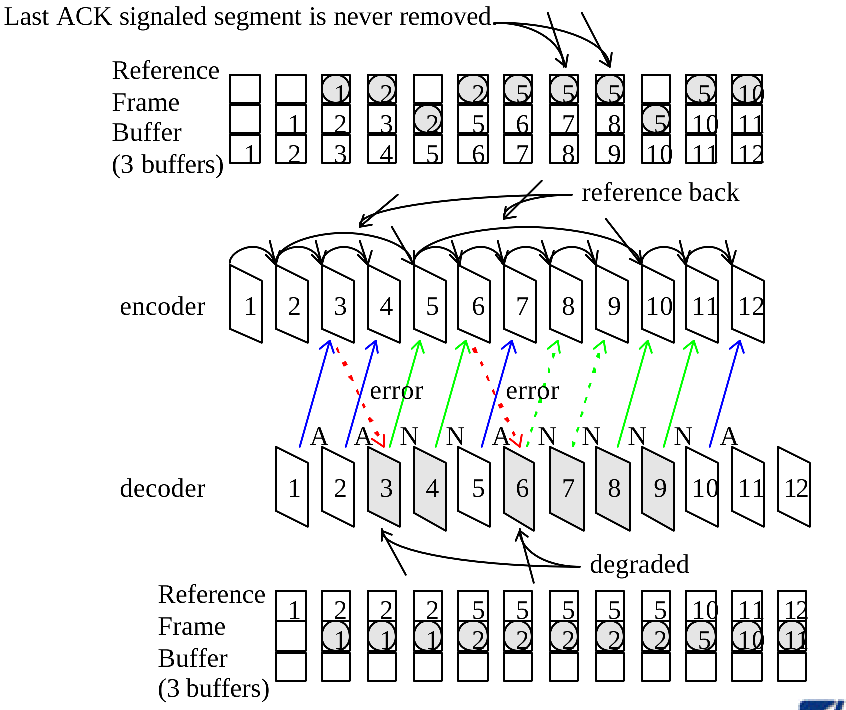
- *ACK mode:* Encoder only uses acknowledged (correctly received) frames as reference frames for prediction
  - Advantages: Minimizes error propagation
  - Disadvantages: May use “old” frames → poor compression



[NewPRED Description in MPEG-4 Spec]

Overcoming Error Propagation: Point-to-point with back-channel (cont.)

- *NACK mode:* Encoder uses last coded frame as reference for prediction, unless negative acknowledgement is received (an error has occurred)
  - Advantages: Usually uses most recent coded frame
  - Disadvantages: More error propagation



[NewPRED Description in MPEG-4 Spec]

Overcoming Error Propagation: Point-to-point with back-channel (cont.)

- Special case: Point-to-point communication with back channel
- Summary for this special case:
  - *Availability of feedback can have significant impact*

    [NewPred in MPEG-4, Reference Picture Selection in H.263]
  - *However, requires reliable back channel with a short round trip delay (RTD)*
    - Effectiveness decreases as RTD increases (measured in terms of frame intervals)
    - Not applicable for broadcast, multicast, point-to-point w/o back channel, as well as pre-encoded video

Reviewing Scalable Video Coding

- *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
- Scalable coding is a nice match for networks which support different qualities of service

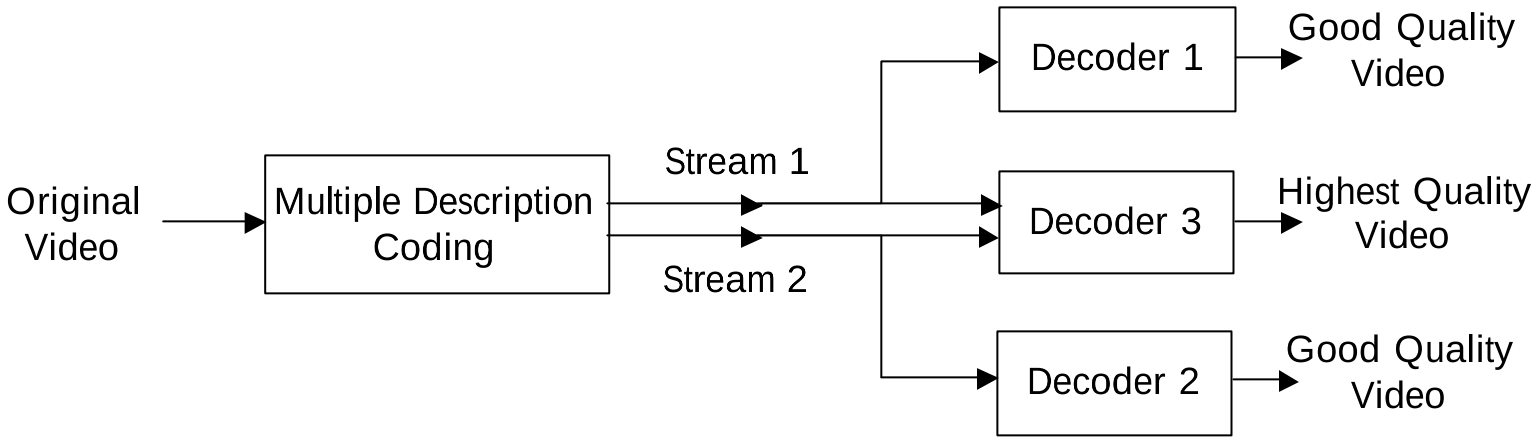
Reviewing Scalable Video Coding (cont.)

- Current *Internet is best-effort:*
  - Does not support QoS
  - All packets are *equally likely to be lost*
- Furthermore, base layer is critical
  - Video can be *completely lost* if there is an error in the base layer

Outline

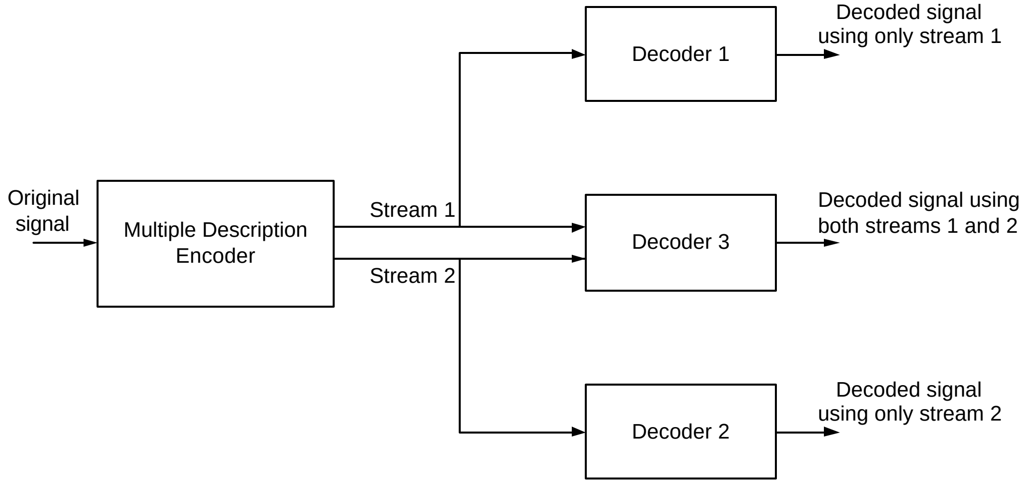
- Review of video streaming from last lecture
- Error concealment
- Error resilient video coding
  - Two basic error-induced problems:
    1. Bitstream synchronization
    2. Incorrect state and error propagation
  - → Multiple description video coding
- Appendix: Protocols for media streaming over the Internet

Multiple Description Video Coding



- Multiple description (MD) video coding:
  - Code video into a number of descriptions, each of *roughly equal importance*
- Properties of an MD coder:
  - Receiving *either bitstream* leads to good quality video
  - Receiving *both bitstreams* leads to highest quality video

Multiple Description Video Coding (cont.)

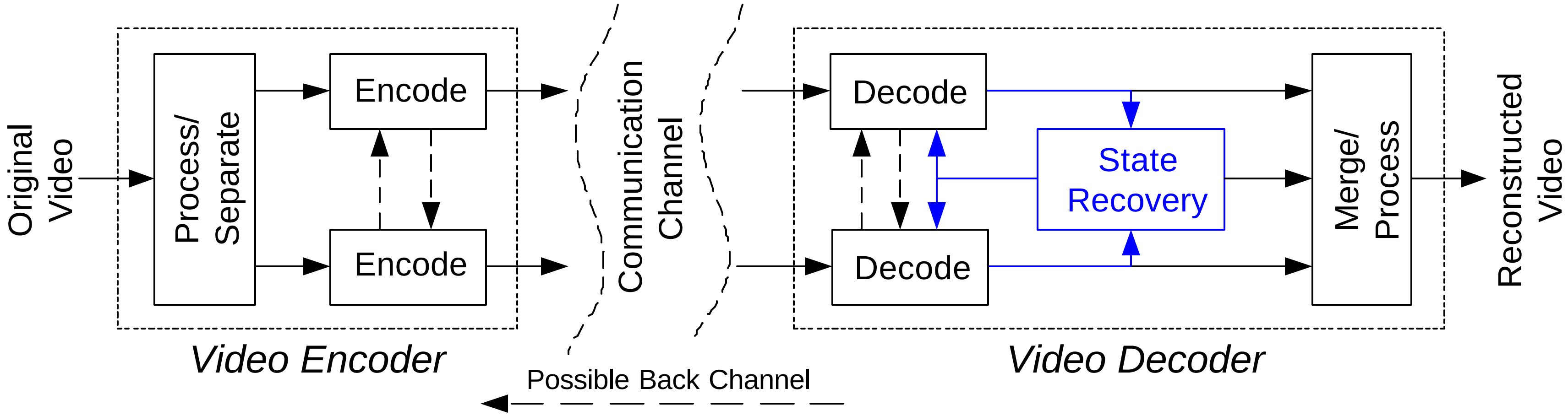


- MD video coding approaches:
  - Multiple threads with resync frames [Wenger]
  - Predictive MD quantizer [Vaishampayan, John]
  - MD transform coding [Reibman, Jafarkhani, Wang, Orchard, Puri]
  - Multiple states [Apostolopoulos]

Multiple State Video Coding

- Motivation:
  - – Conventional video coders have a single-state (previous coded frame) which if lost leads to error propagation
- Idea: Design a coder with multiple states, such that if one is corrupted, the other remains accurate, and can be used to estimate the corrupted state → state recovery
- Proposed video coder:
  - – Code video into multiple independently decodable streams, each with each own prediction process and state information
  - – Enable state recovery at the decoder
  - → One form of multiple description video coding

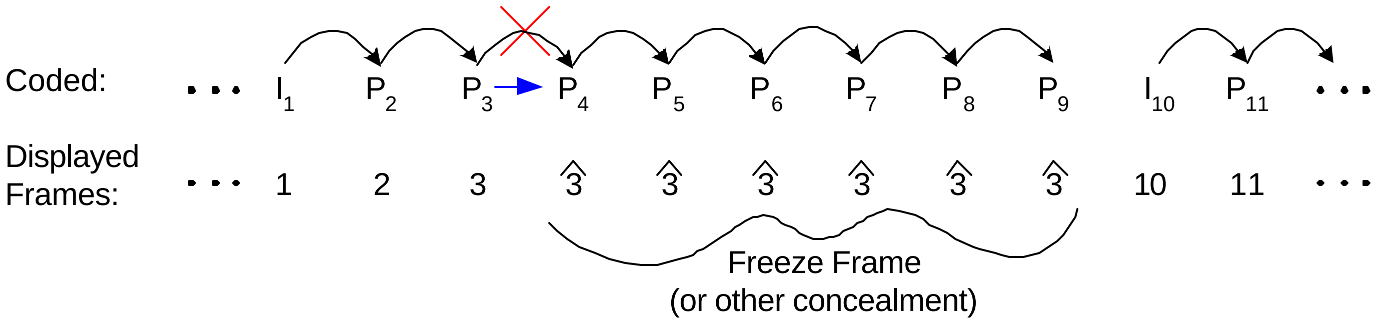
Multiple State Video Coding (cont.)



- Split video into even and odd frames:
  - – Two separate encoders, or *one encoder that stores last two previously coded frames*
  - → *Standard-compliant with MPEG-4 V2 and H.263 V2*
- Reduced compression efficiency since frames are spaced farther apart in time
- Improved error resilience because of *state recovery*

Error Effects: Conventional vs Multiple-State Approach

Conventional Approach

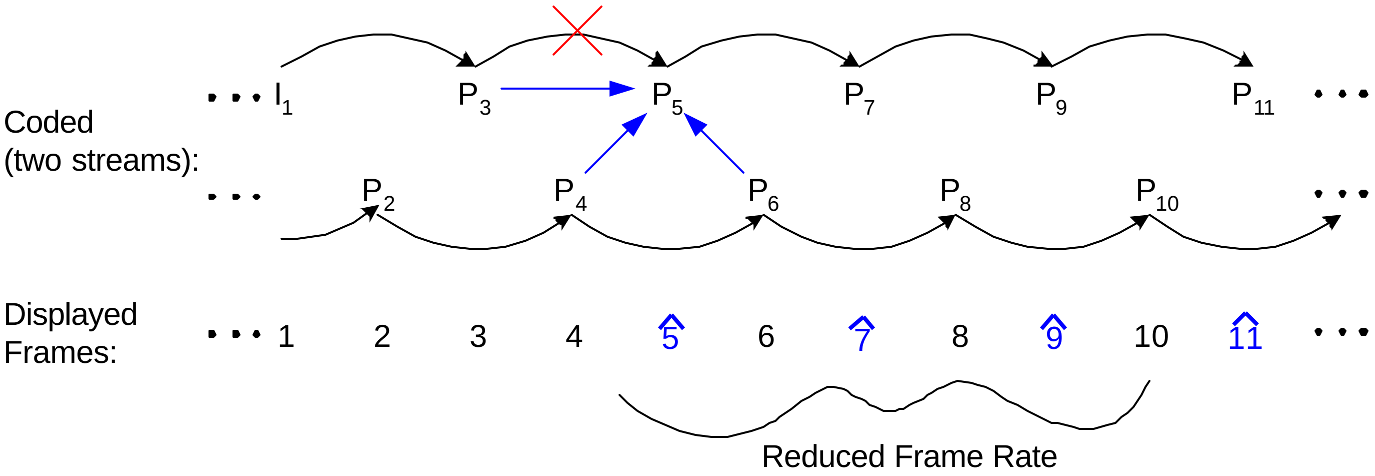


Coded: . . . I_1 P_2 P_3 P_4 P_5 P_6 P_7 P_8 P_9 I_{10} P_{11} . . .

Displayed Frames: . . . 1 2 3 $\hat{3}$ $\hat{3}$ $\hat{3}$ $\hat{3}$ $\hat{3}$ $\hat{3}$ 10 11 . . .

Freeze Frame
(or other concealment)

Multiple-State Approach



Coded (two streams):

. . . I_1 P_3 P_5 P_7 P_9 P_{11} . . .

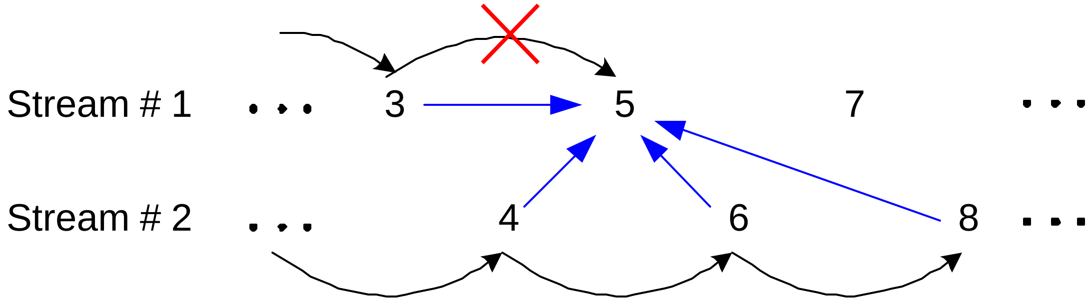
. . . P_2 P_4 P_6 P_8 P_{10} . . .

Displayed Frames: . . . 1 2 3 4 $\hat{5}$ 6 $\hat{7}$ 8 $\hat{9}$ 10 $\hat{11}$ . . .

Reduced Frame Rate

State Recovery

State Recovery:



Stream # 1 . . . 3 5 7 . . .

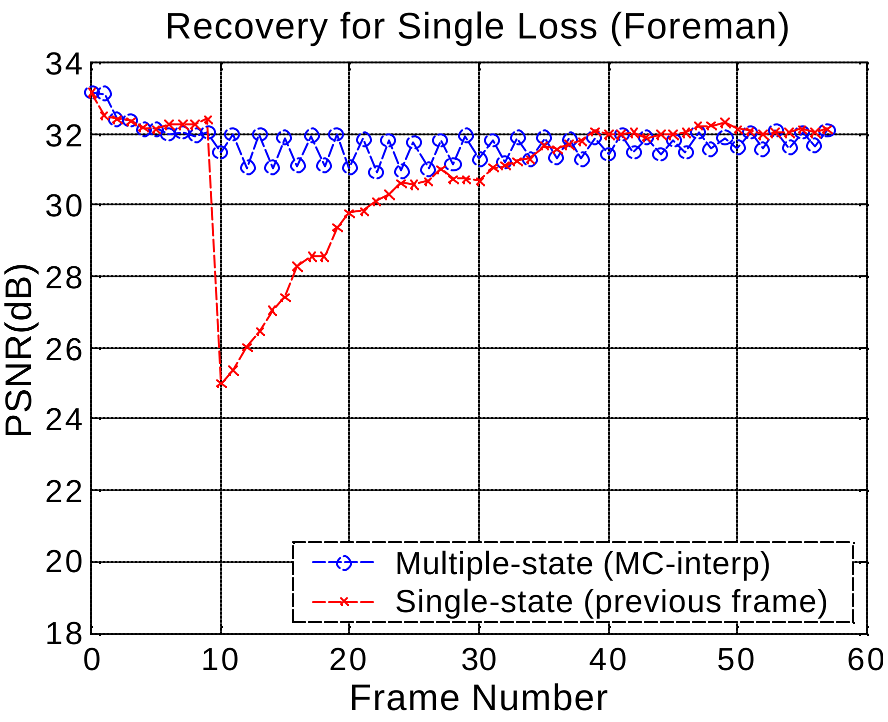
Stream # 2 . . . 4 6 8 . . .

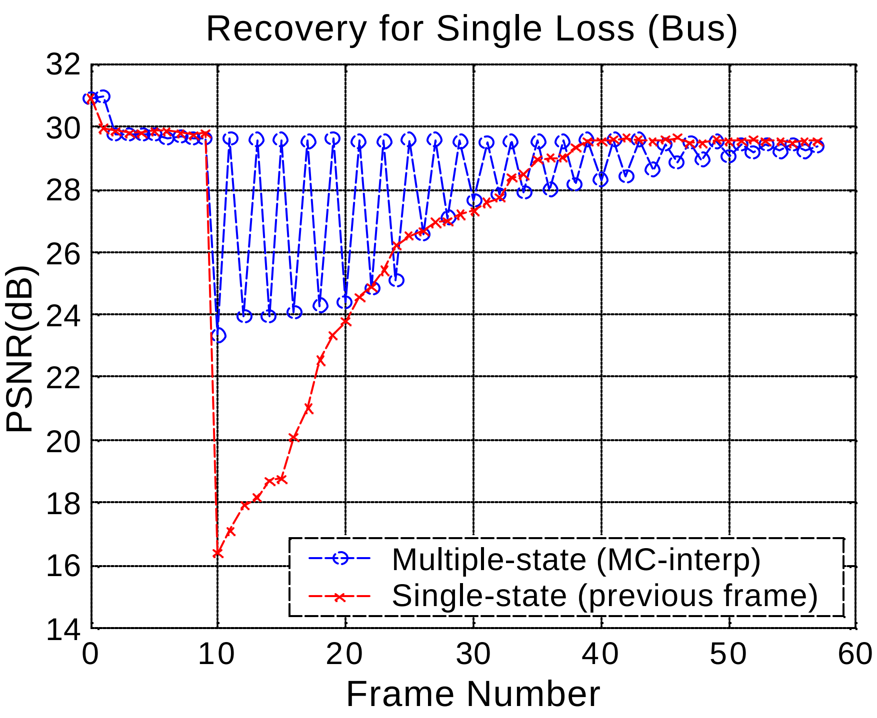
- *Novelty: Improved recovery from errors*
  - – Conventional (single-state) only has access to *previous frames*
  - – Multiple-state has access to *previous and future frames*
  - → Availability and careful usage of *bi-direction information* can greatly assist in recovering the corrupted stream
- State recovery is similar to MC-interpolation
  - – Can directly leverage prior work on MC-interpolation

Experimental Setup

- Test setup: Bus (CIF, 30 f/s), Foreman (QCIF, 30 f/s) coded at constant quality (PSNR) of 29.5 and 31.9 dB
- Multiple state encoding with two streams
  - – 53 and 4.45 kb/P-frame
  - – 12 % and 20 % larger than conventional approach designed for error-free channel
- Coded both conventional and MS at the same bit rate (57.8 and 4.72 kb/P-frame) with extra bits for intra coding
- Assumption: Any loss leads to the loss of one entire frame
- Assumption: Independent losses of each stream [Apostolopoulos]
- Error recovery:
  - – MS: Apply MC-Interpolation to estimate lost frame
  - – Conventional: Estimate lost frame as last correctly decoded frame

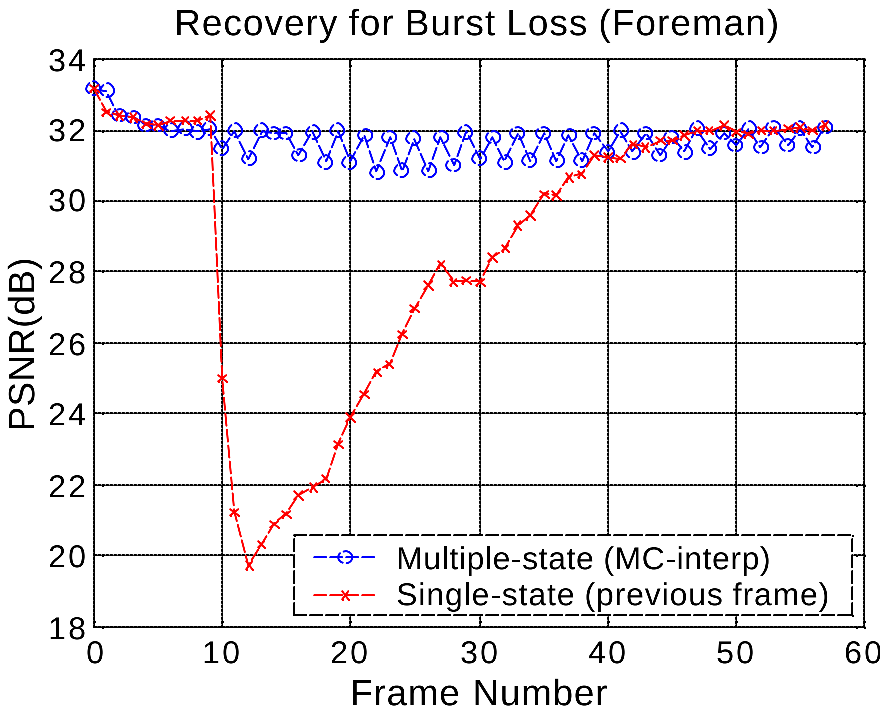
Experimental Results: Single Packet Loss

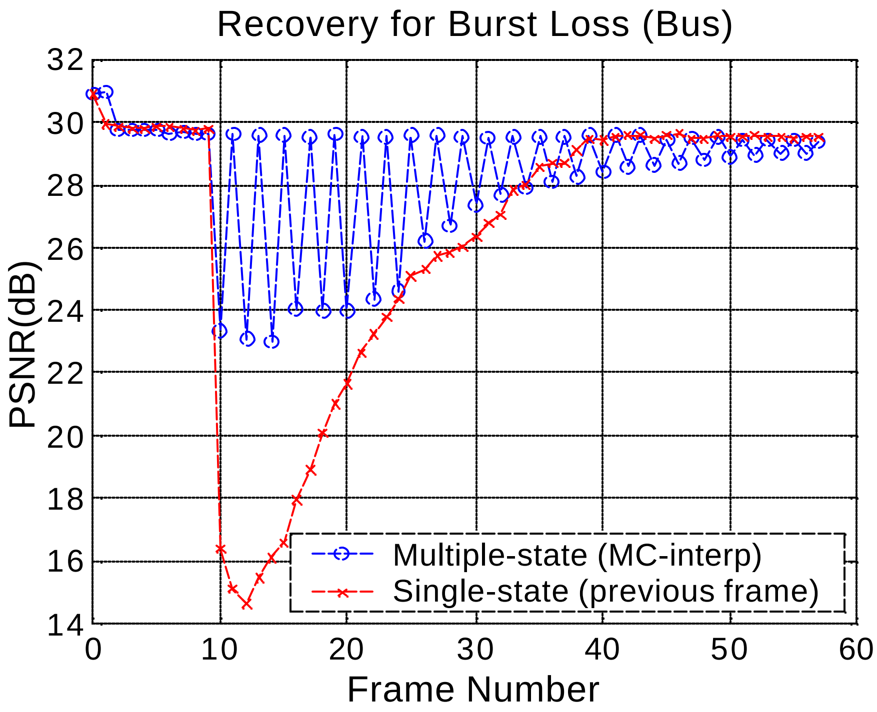

Recovery for Single Loss (Foreman)


Recovery for Single Loss (Bus)

- Single-state has more intra coding, converges slightly faster to point of complete recovery, however significantly more vulnerable to losses
- Multiple-state uses bi-directional information to more effectively recover from a loss (state recovery)

Experimental Results:
Burst loss of 100 ms (loss of 3 frames)

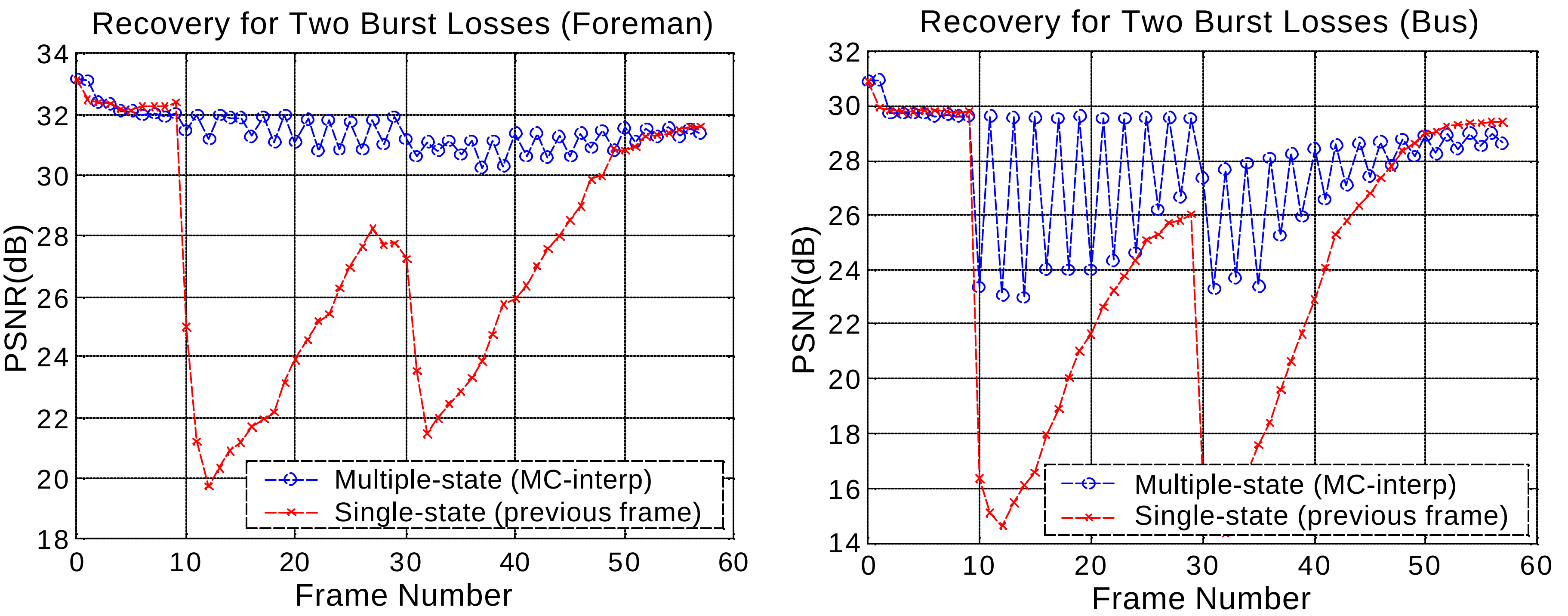




- SS: Longer durations of loss lead to greater reductions in quality
- MS: Largely immune to the duration of the loss (as long as the other stream is correctly received)

Reliable Video

Experimental Results: Two burst losses of 100 ms duration, spaced apart by 2/3 sec



- Illustrates the effect of a significant loss while the system is still recovering from a previous loss
- MS: Separate states can bootstrap off of each other, as long as they are not both simultaneously corrupted

Experimental Results

Conventional Approach

Multiple Description

- Foreman (QCIF) and Bus (CIF), 30 f/s, coded at constant quality (PSNR), 4.72 and 57.8 kb/P-frame, with extra bits for intra coding
- Burst loss (congestion of 100 ms duration, 3 frames lost)
- Standard-compatible enhancement to MPEG-4 V2 and H.263 V2

Summary of Today’s Lecture

- Review of video streaming from last lecture
- Error concealment
- Error resilient video coding
  - Two basic error-induced problems:
    1. Bitstream synchronization
    2. Incorrect state and error propagation
  - Multiple description video coding
- Appendix: Protocols for media streaming over the Internet

Near Future: Wireless Video

- Wireless video, e.g. videophone on a cell phone
  - Third generation (3G) cellular
  - Data rates: 64, 384 kb/s
- Challenges:
  - Limited bandwidth
  - Low delay (interactive)
  - Mobility (handoffs)
  - High error rate resulting from multipath fading (bit errors, burst errors, packet loss)
  - *Highly dynamic* channel characteristics
- Requires highly efficient, adaptive, and error resilient video compression

Appendix:

Protocols for Media Streaming over the Internet

Protocols for Video Streaming over the Internet

- Goal: Briefly highlight the communication requirements and network protocols for video streaming
- Brief review of Internet Protocols
  - IP, TCP, UDP
- Media delivery and control protocols:
  - Media delivery
    - RTP, RTCP
  - Media control
    - RTSP, SIP
  - Media description and announcement
    - SDP, SAP

Review of Internet Protocols

- Internet Protocol (IP): Addressing
- Transmission Control Protocol (TCP): Provides reliable stream services. Guarantees delivery via retransmissions and acknowledgements.
- User Datagram Protocol (UDP): Provides unreliable, connectionless, packet delivery service

- Web & data traffic: TCP/IP
  - Note: Data traffic does not have delay constraints, video streaming does
- Video streaming:
  - Data via UDP/IP
  - Control via TCP/IP

Real-time Transport Protocol (RTP) and Real-time Control Protocol (RTCP)

- *RTP and RTCP:* IETF protocols designed to *support streaming media*
- RTP for data transfer, RTCP for control
- Note: These protocols do NOT enable "real-time" services, only the underlying network can do this, however they provide functionalities that support "real-time" services
- RTP does not guarantee QoS or reliable delivery, but provides support for applications with time-constraints
  - Time stamps
  - Sequence numbering
  - Payload type
- RTP enables detection of lost packets

RTP & RTCP (cont.)

- *RTCP provides feedback on quality of data delivery*
  - QoS feedback: # of lost packets, inter-arrival jitter, delay
  - Periodic feedback packets, no more than 5% of total session BW, at least one every 5 sec
  - Sender can use feedback to adjust its operation, e.g. adapt bit rate
- Conventional approach: RTP/UDP and RTCP/TCP or UDP

[Henning Schulzrinne, www.cs.columbia.edu/~hgs/rtp]

Session Control Protocols: RTSP and SIP

- Real-Time Streaming Protocol (RTSP)
  - Establishes a session
  - Supports VCR functionalities
    - E.g. Start / stop, pause / resume, fast forward / reverse
- Session Initiation Protocol (SIP)
  - Commonly used in VoIP
  - Similar to RTSP
  - In addition, can support user mobility

Additional Protocols

- Session Description Protocol (SDP)
  - – Provides information describing the session
  - – For example: Video or audio, codec, etc.
- Session Announcement Protocol (SAP)
  - – To announce the availability of a multicast session

Summary of Protocols for Streaming Media

- Media encoding
  - – MPEG-4 video and audio, H.263 video
- Media transport
  - – RTP (data) & RTCP (control – QoS feedback), usually over UDP/IP
- Media control
  - – RTSP, maybe SIP
- Media description and announcement
  - – SDP, SAP