ENSC 424 - Multimedia Communications Engineering

Error control in video communications

Thanks to Prof. Yao Wang of NYU for providing some of the PPT slides



Outline

- Error control challenges
- Overview of error control techniques
  - Error resilient encoding
  - Error concealment
  - Encoder/decoder interactive error control



Video communications challenges

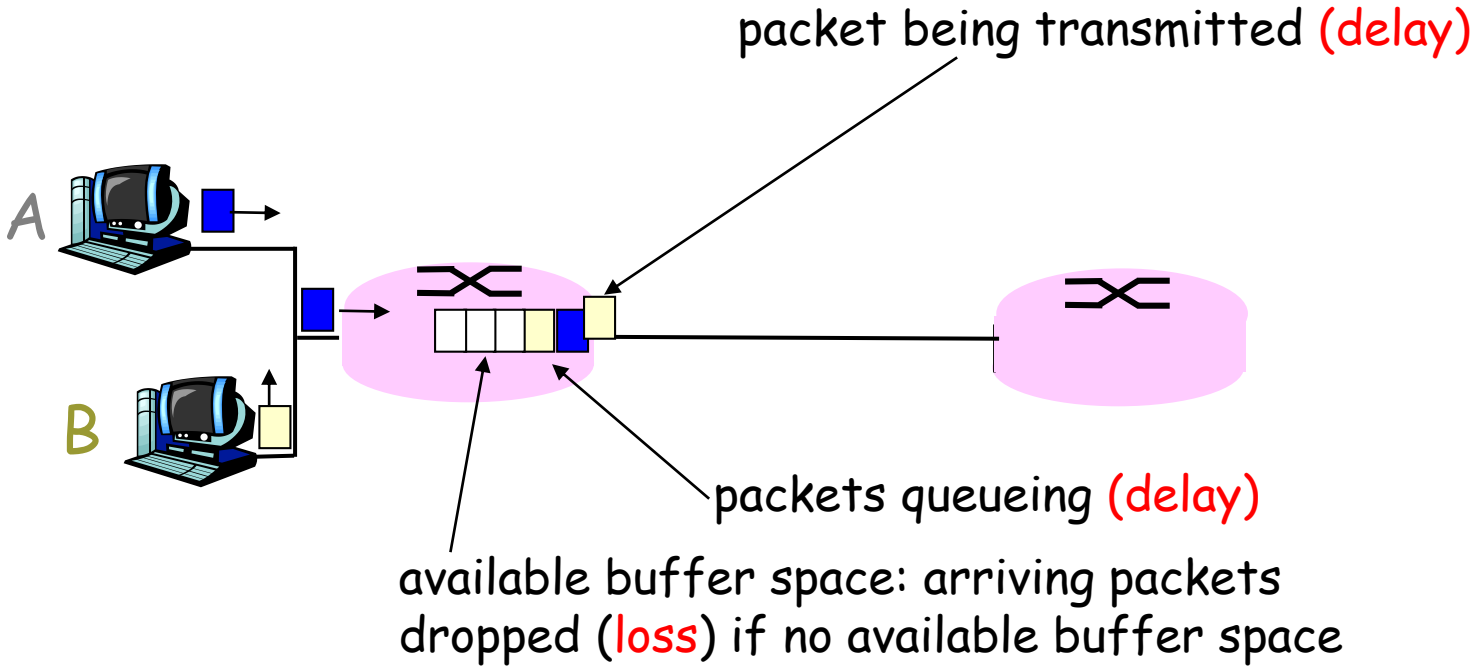
- Real networks are unreliable
  - Wireless networks: random bit errors, long burst errors, and possibly link outages
  - Internet: packet loss and variable delay due to network congestion
  - Excessive delay = loss for real-time applications
- Real networks are heterogeneous in bandwidth and reliability
- Video information is delay-sensitive
  - Retransmission sometimes infeasible



How do loss and delay occur?

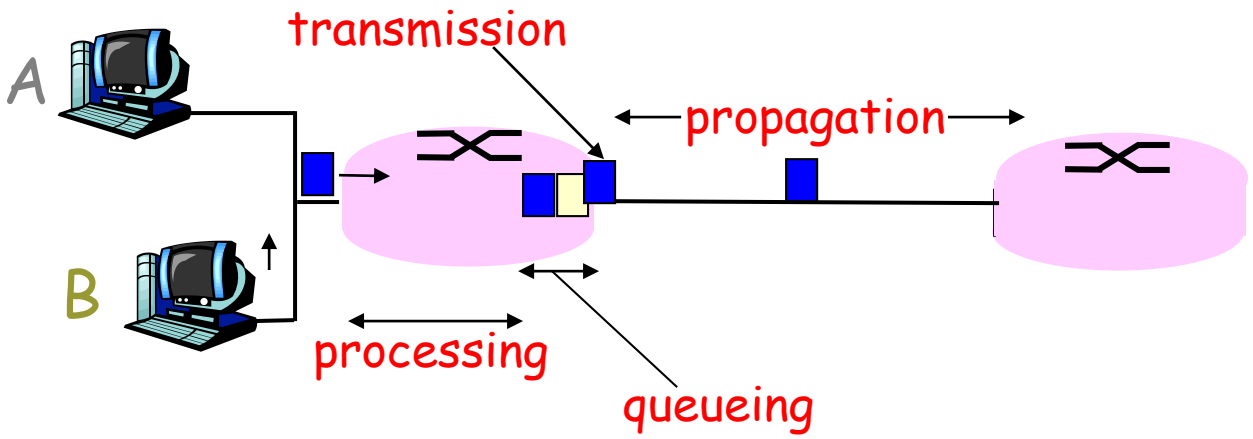
Packets *queue* in router buffers

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



Four sources of packet delay

- 1. Processing delay:
  - check bit errors
  - examine header
  - determine output link
- 2. Queueing delay:
  - time waiting for transmission
  - depends on congestion level of router



Delay in packet-switched networks

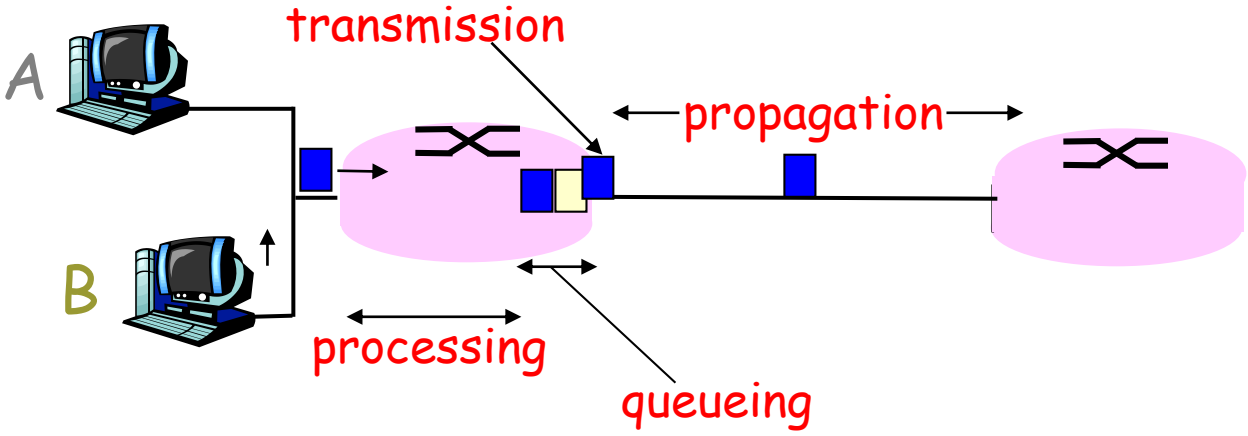
3. Transmission delay:

Store-and-forward delay

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

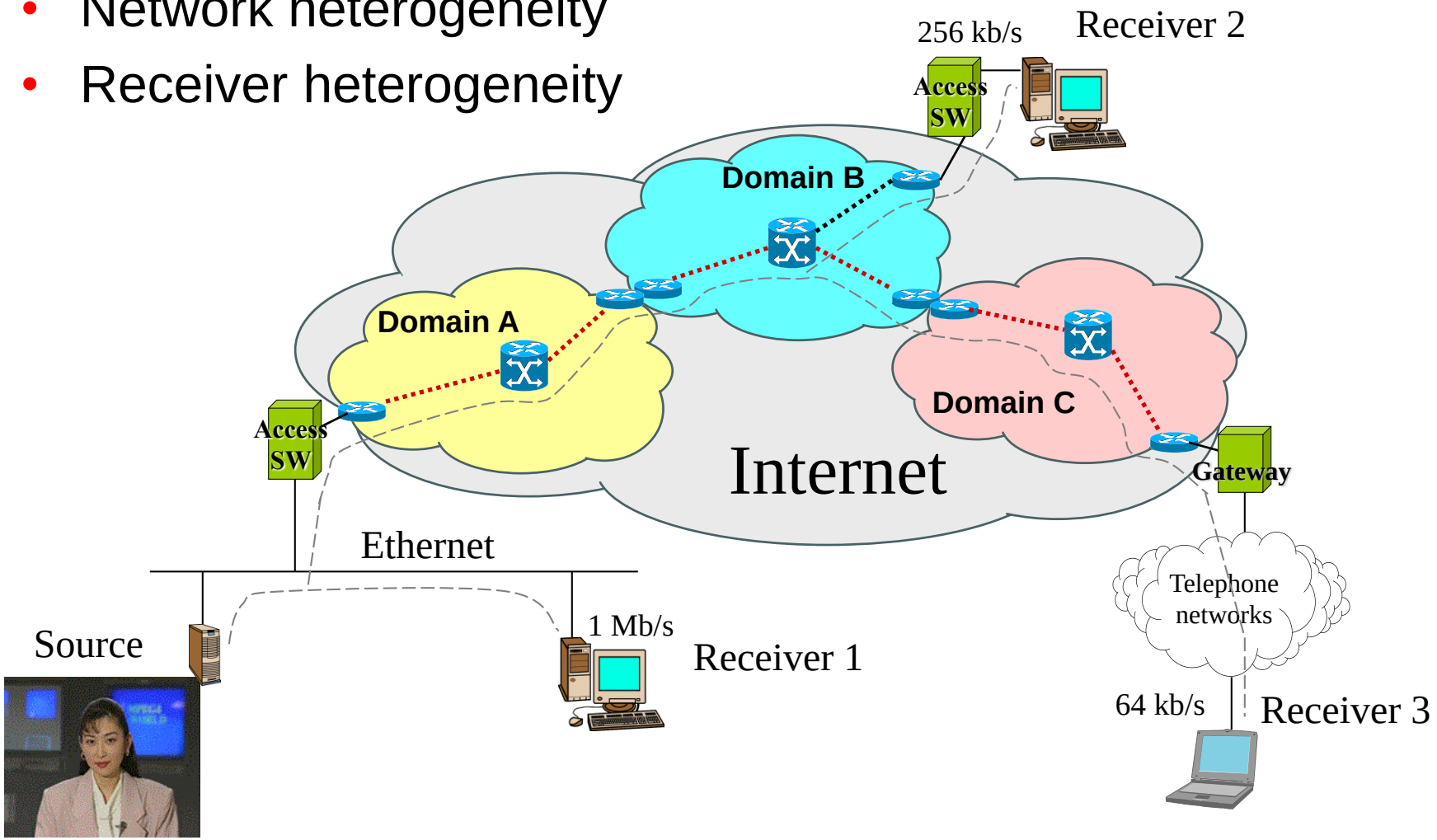
4. Propagation delay:

- d = length of physical link
- s = propagation speed in medium (~$2x10^8$ m/sec for optical cable)
- propagation delay = d/s



Heterogeneity

- Network heterogeneity
- Receiver heterogeneity

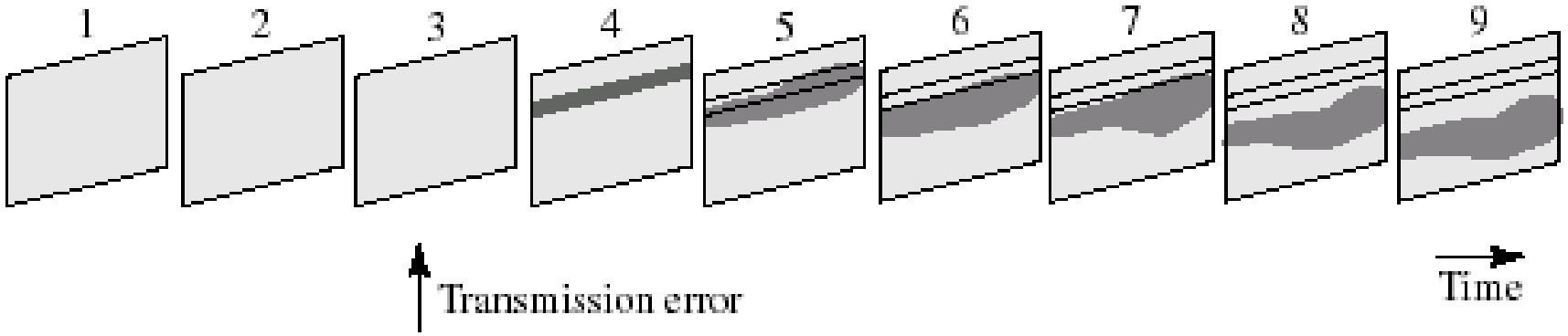


Compression versus robustness

- Compressed video data is very sensitive to transmission errors
  - Variable length (entropy) coding
  - Temporal predictive coding
  - Spatial predictive coding
  - All contribute to error propagation within the same frame as well as in the following frames
  - Even a single bit error or packet loss can render subsequent received data useless

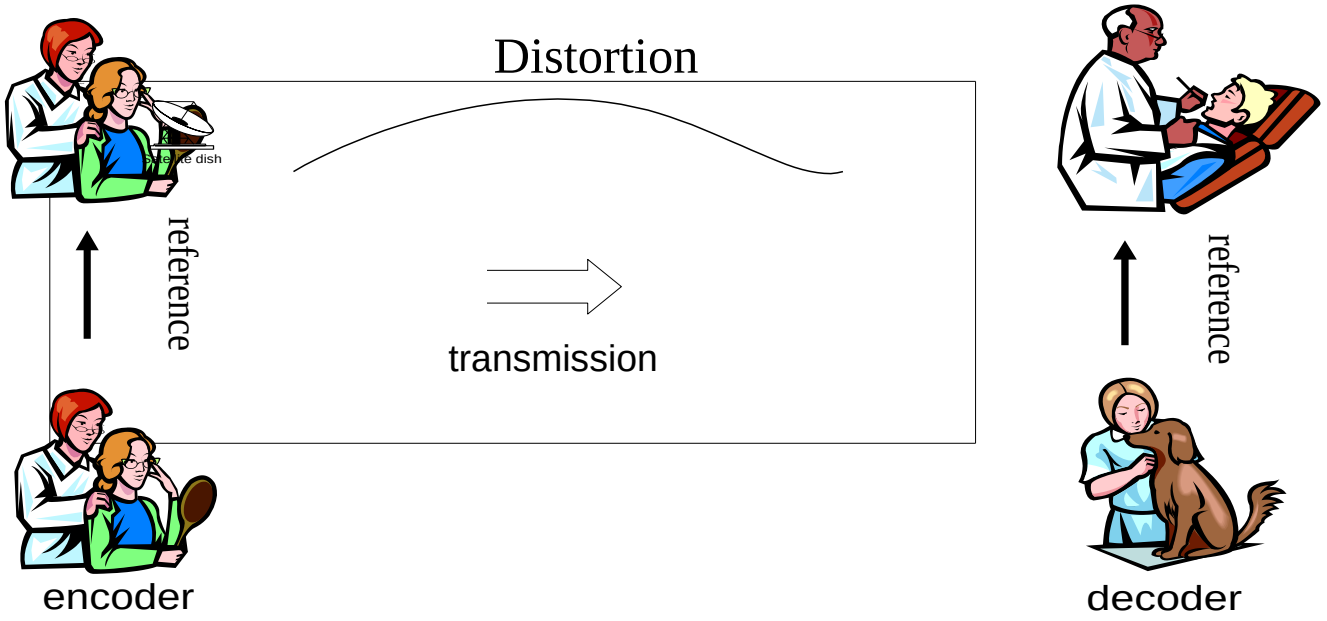


Spatial/temporal error propagation



Drift problem (reference mismatch)

- Loss in a previous frame can cause mismatch between the reference frame used in the encoder and that in the decoder
  - Encoder and decoder out of synchronization



Effects of transmission errors

Coded,
No loss



3%

5%

10%

Example reconstructed video frames from a H.263 coded sequence, subject to packet losses

Typical video communications applications

- Interactive two-way visual communications
- One-way video streaming
- One-way video downloading
  - No difference from file downloading



Interactive visual communications

- E.g. Teleconferencing, video telephony, virtual classroom
- Very stringent delay requirements
  - ≤ 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 done 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.
- Conversational video usually has limited motion → easier to code and conceal errors



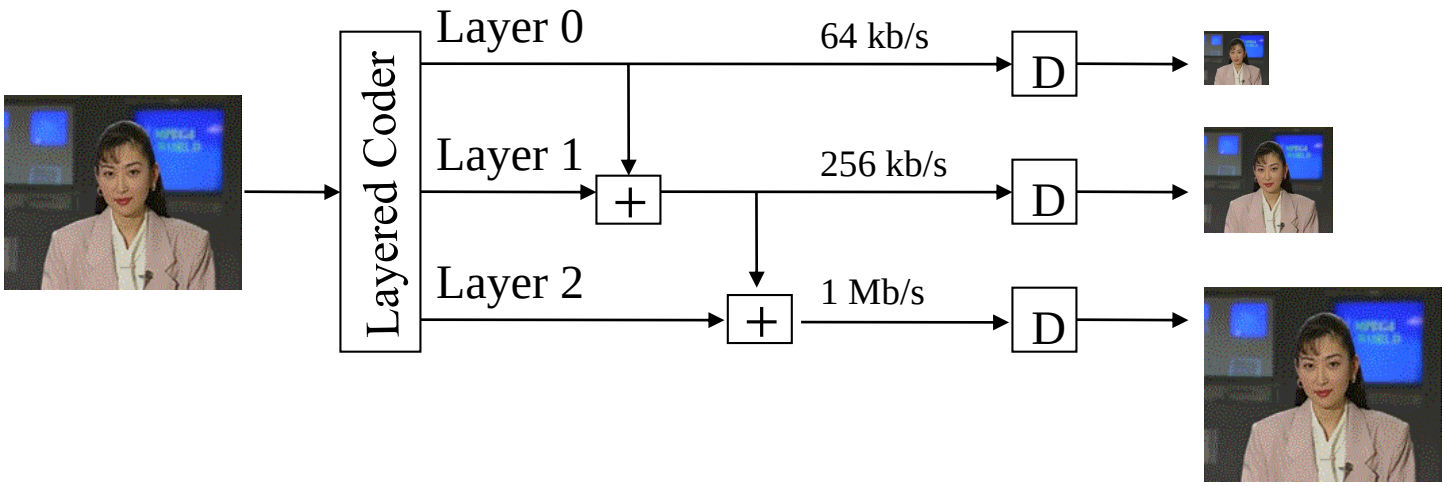
Video streaming

- E.g. TV broadcast, Multicast of a conference/event, Video on Demand
- 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 (from wireless modem to fast Ethernet) and the receiving terminal may have varying computing power (e.g. cell phone vs. laptop vs. desktop)
  - Scalable coding desired

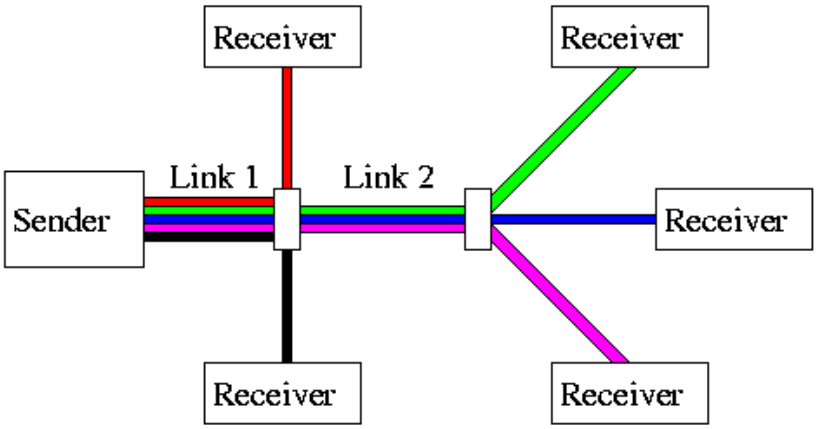


Scalable video coding

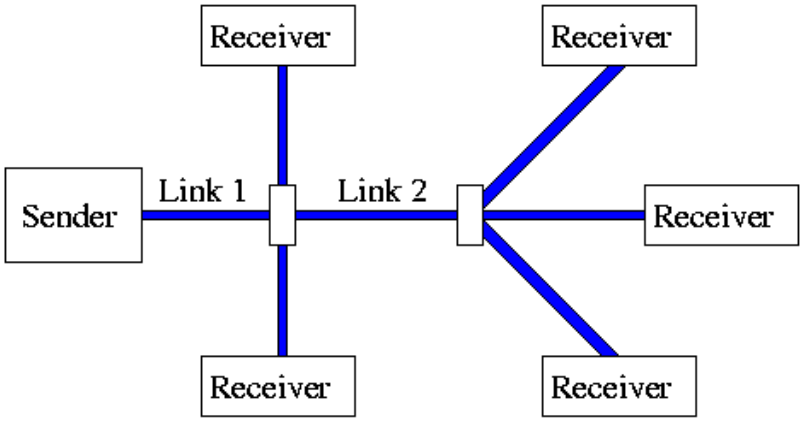
- Recall: typical forms of scalability are spatial (resolution), temporal (frame rate) and SNR (quality)



Simulcast versus scalable multicast

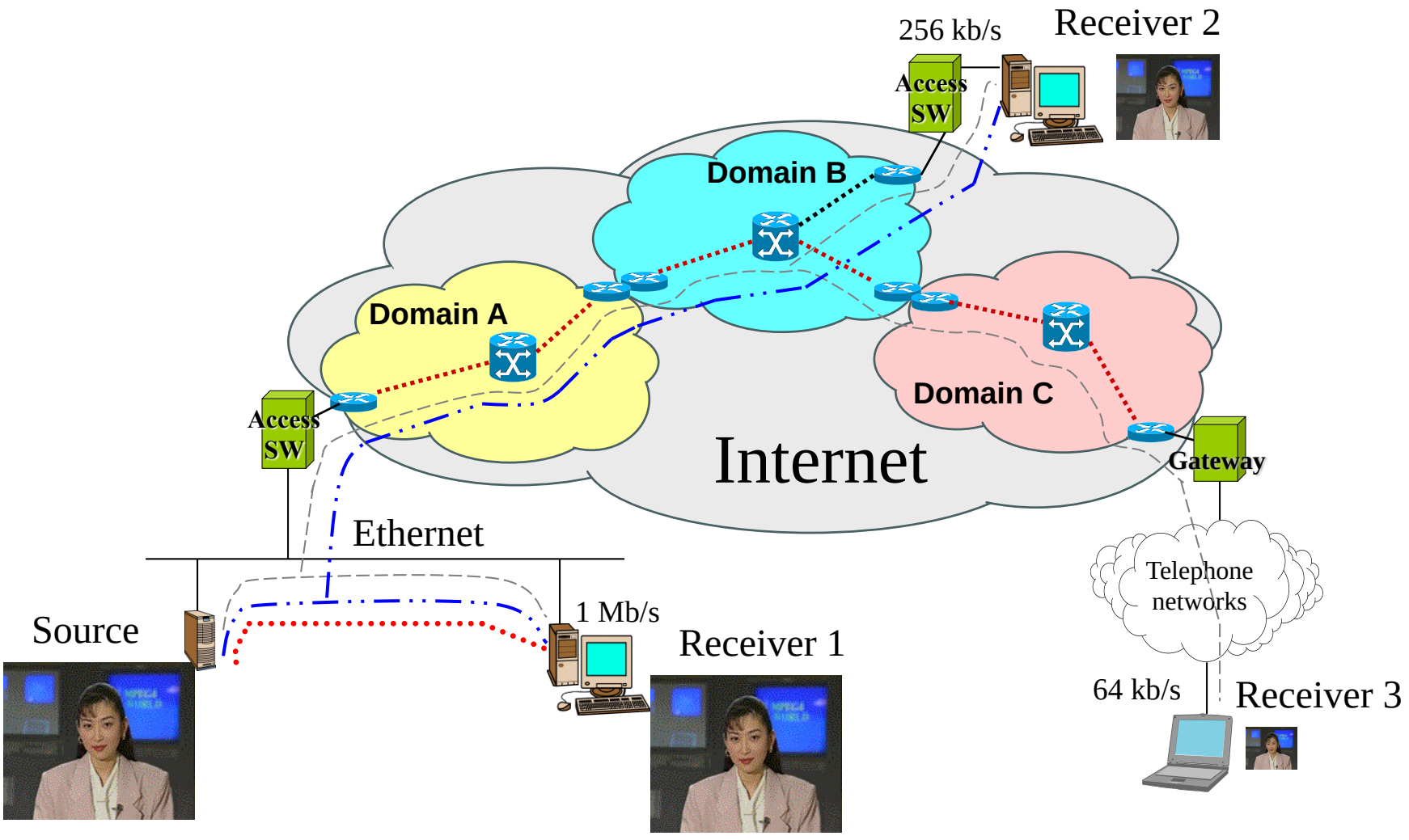


Simulcast



Scalable multicast

Scalable video multicast



Error control for video

- Transport-level error control
  - Error detection and correction through FEC
  - Retransmission of lost packets
- Error resilient encoding
  - Adding redundancy in the bit stream to help the decoder recover from transmission error
- Error concealment
  - Recover (interpolate/extrapolate) lost/damaged regions based on image/video characteristics and human visual system properties at the decoder
- Encoder-decoder-network interactive error control
  - Joint source/channel coding
    - E.g.: layered coding with unequal error propagation
  - Feedback-based adaptive encoding
    - E.g.: Reference picture selection, Selective intra update
  - Path diversity
    - Different bit streams sent through separate paths



Transport-level error control

- Channel coding
  - Forward Error Correction (FEC)
  - Forward error detection
- Retransmission
  - Automatic Retransmission reQuest (ARQ)



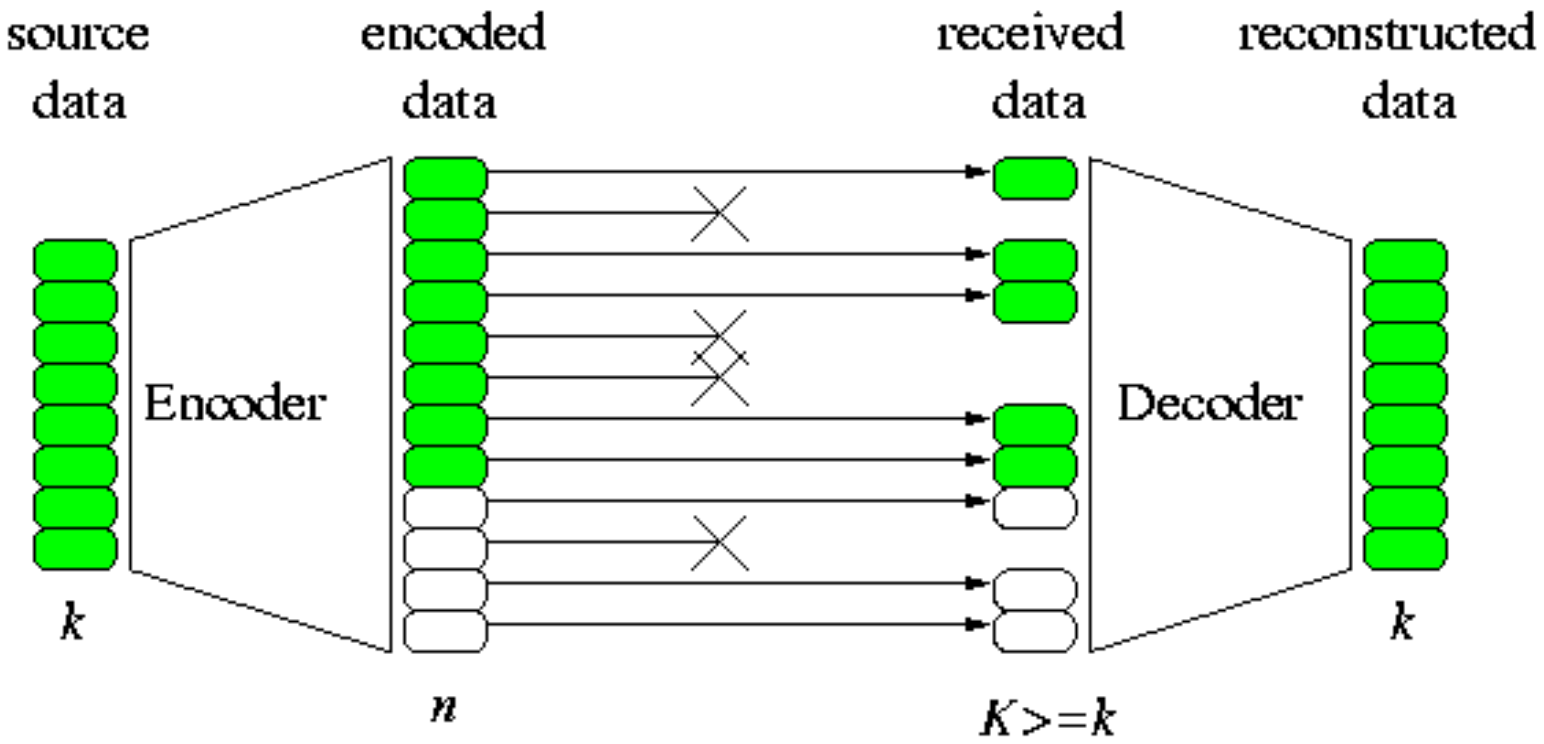
Channel coding basics

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



FEC for video communications

- FEC can be applied across packets to correct packet losses



- **Unequal Error Protection** (UEP): using stronger channel codes for more "important" parts of the bit stream (e.g., base layer).

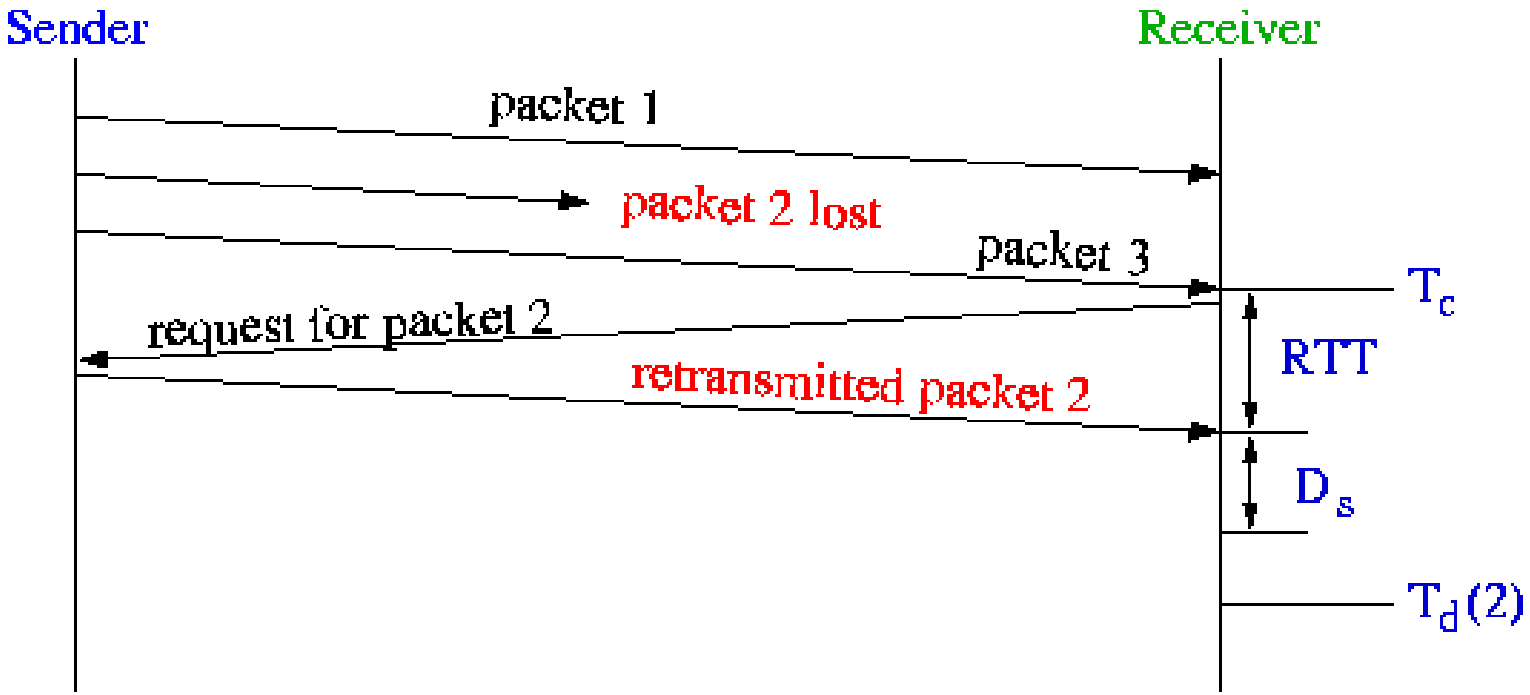
Delay-constrained ARQ

- 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)
  - Applying ARQ to “important” packets (base layer) is another way to achieve UEP
- For broadcast/multicast applications, ARQ is inappropriate in general, although it can be deployed at the link layer
  - But there are other forms, such as type 2 hybrid ARQ/FEC, that are applicable to multicast



Delay-constrained ARQ

- Retransmitted packet can be useful if it is received before the playout deadline



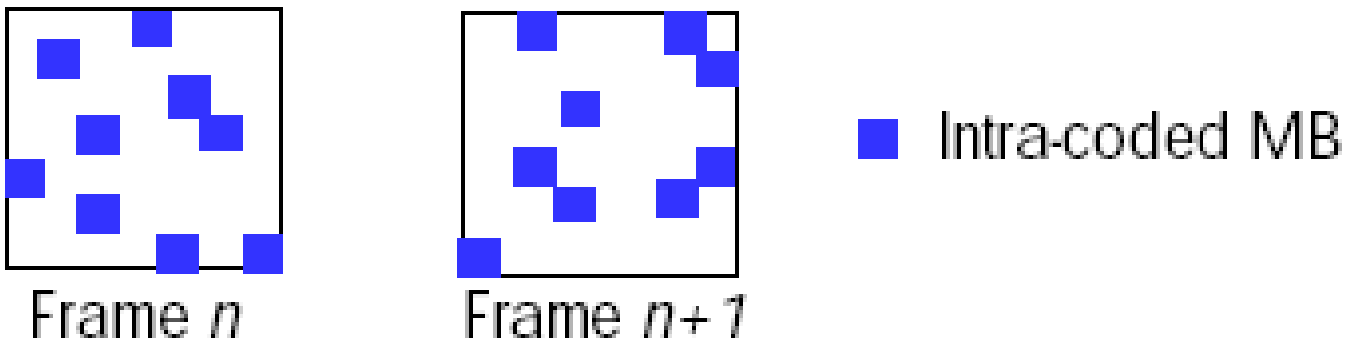
Error-resilient encoding

- 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 partitioning
- Robust binary encoding
  - Reversible VLC (RVLC) (part of H.263/MPEG4 standard)
- Error resilient prediction
  - Periodically insert intra-coded frames or blocks
  - Independent segment prediction (part of H.263/MPEG4 standard)
- Layered coding with unequal error protection
- Multiple description coding



Periodic intra-coding

- To limit effect of error propagation:
- Periodic I-frames:
  - Example: I-frame every 15 frames
- Partial intra-coding of each frame:
  - Individual macroblocks (MBs) are intra-coded
  - A fraction of the MBs in each frame are intra-coded in some predefined order



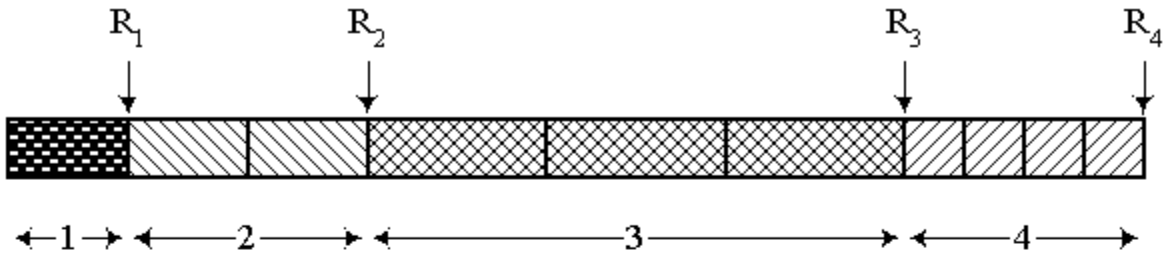
Layered/scalable coding + UEP

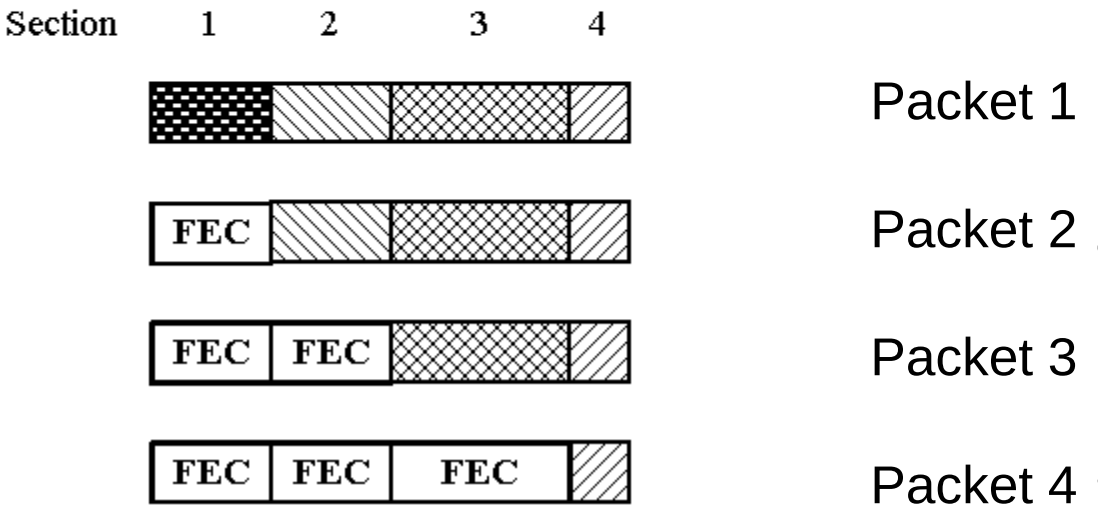
- SC+UEP is one of the promising approaches for combating channel errors
  - Base layer provides acceptable quality, enhancement layer refines the quality
  - Base layer stream is delivered through a reliable channel (by using ARQ and/or strong FEC)
  - Good for a network with differentiated services (do NOT exist today in the Internet, may become part of the 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



Scalable coding + UEP example

- Scalable bit stream split into sections, each protected using different amount of redundancy



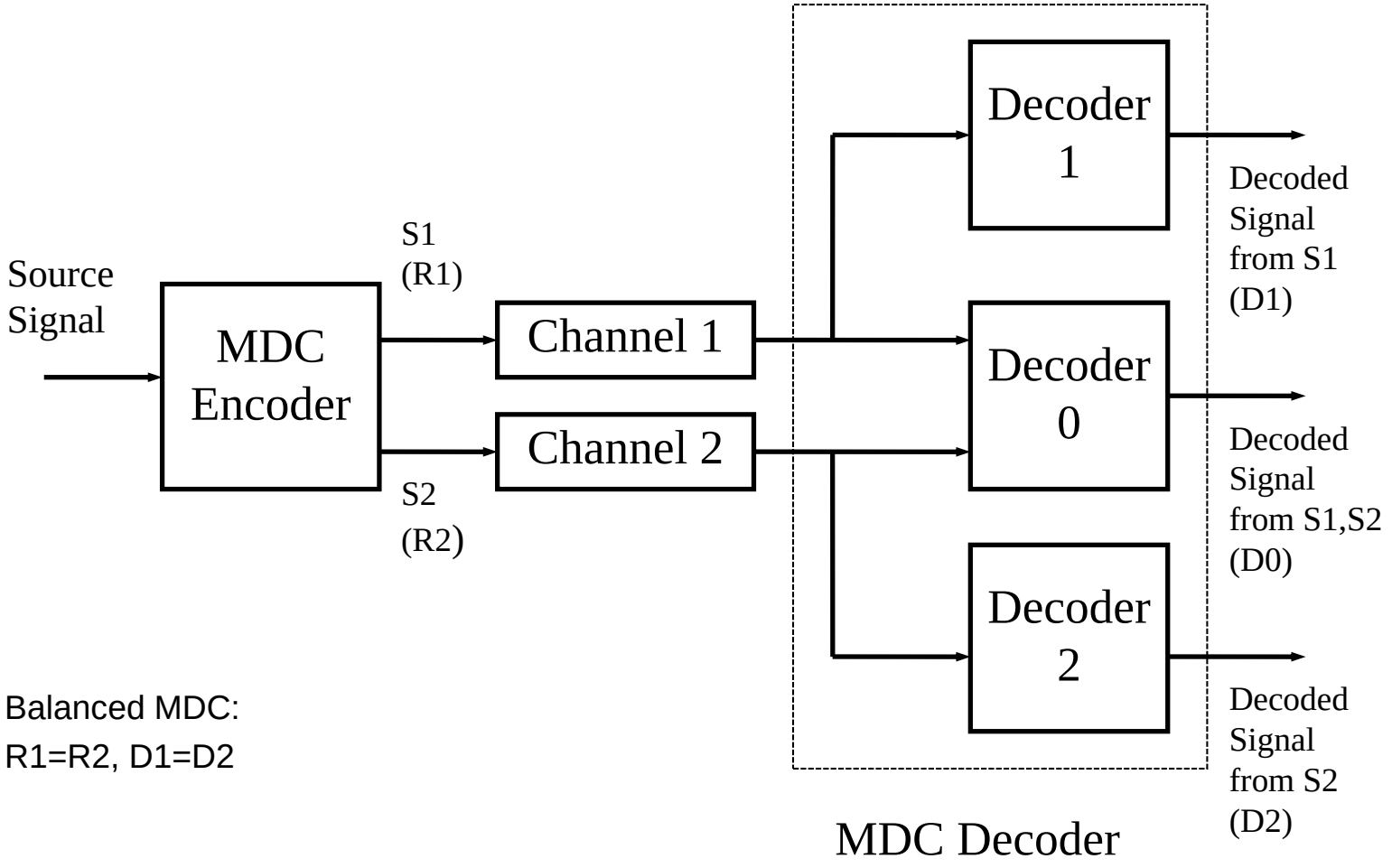


Multiple Description Coding (MDC)

- Assumptions:
  - multiple channels between source and destination
  - independent error and failure events
  - probability that all channels fail simultaneously is low
  - good model for the Internet and wireless networks when data are properly packetized and interleaved
- MDC: Generate multiple correlated descriptions encoded independently
  - any description provides low but acceptable quality
  - additional descriptions provide incremental improvements
  - No retransmission required → low delay ☺
  - However: correlated descriptions coded independently → low 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

Challenges in MD video coding

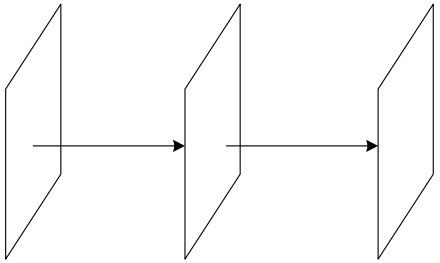
- To achieve high coding efficiency, the encoder should retain the temporal prediction loop
- Should the prediction be based on reconstruction from both descriptions or individual descriptions?
- Prediction based on two-description reconstruction
  - Higher prediction efficiency
  - Mismatch problem at the decoder
- Prediction based on single-description reconstruction
  - Lower prediction efficiency
  - No mismatch problem
- How to provide a proper trade-off between prediction efficiency and mismatch
  - Predict based on two-description reconstruction, but explicitly code the mismatch error



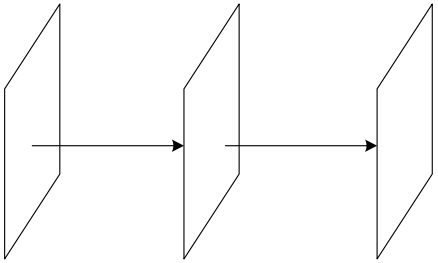
Video redundancy coding in H.263+

- 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

odd frames



even frames



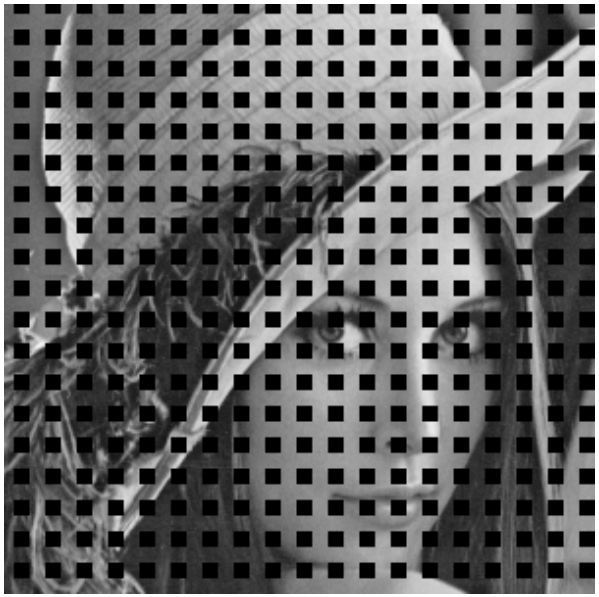
Error concealment

- With proper error-resilience tools, packet loss typically leads to the loss of an isolated segment of a frame – a group of macroblocks (MBs)
- 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 standards
- Decoders on the market differ in their error concealment capabilities



Error concealment techniques

- Basic idea: recover damaged regions by interpolating from surrounding regions in the same and/or nearby frames
- **Spatial interpolation**
  - Use the data in neighboring MBs to interpolate the missing MB



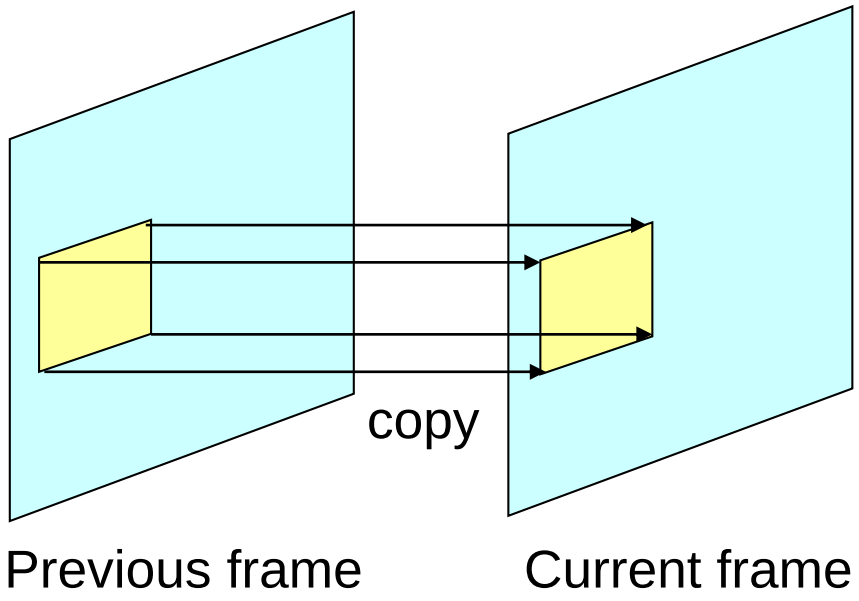
Received image with error



Concealed image (PSNR = 35.8dB)

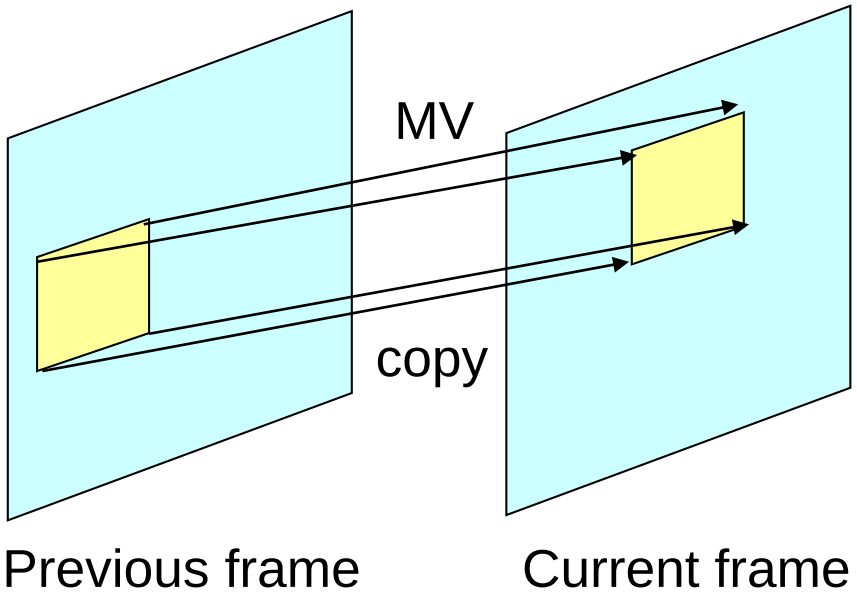
Temporal interpolation

- Use the co-located MB in the previous frame as the estimate of the missing MB



Motion-compensated temporal interpolation

- Replace a damaged MB by its reference MB in the reference frame
- If the MV is also lost, has to estimate the MV first, e.g. by
  - Copying the MV of the MB above
  - Finding the mean or median of available surrounding MVs
  - Re-estimating the MV using boundary matching



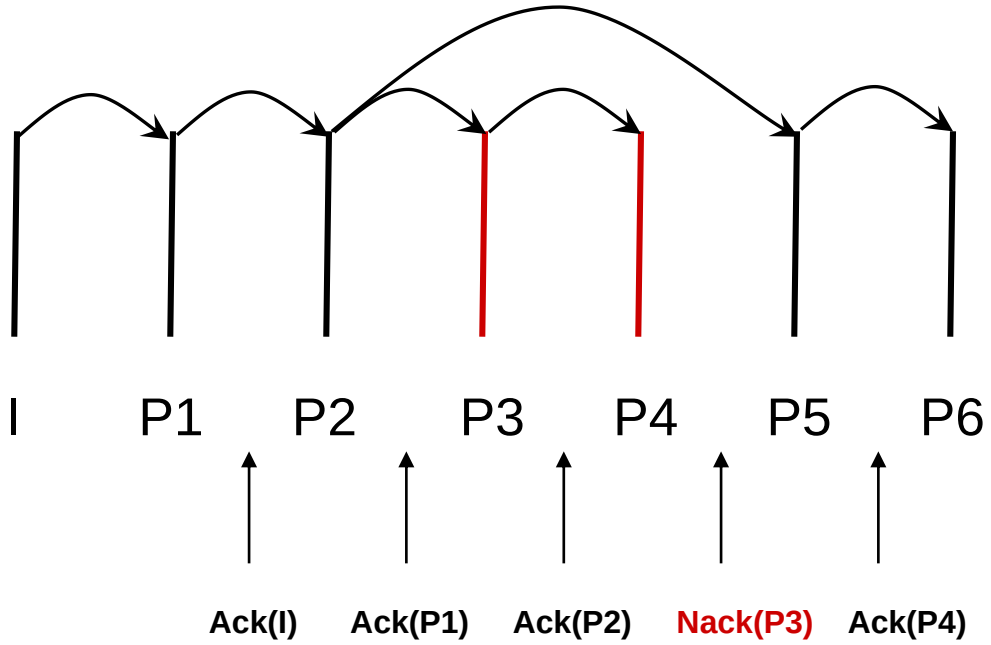
Encoder-decoder interactive error control

- Coding parameter adaptation based on channel conditions
  - E.g., change intra-coding rate based on loss rate
- 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
- Requires a feedback channel



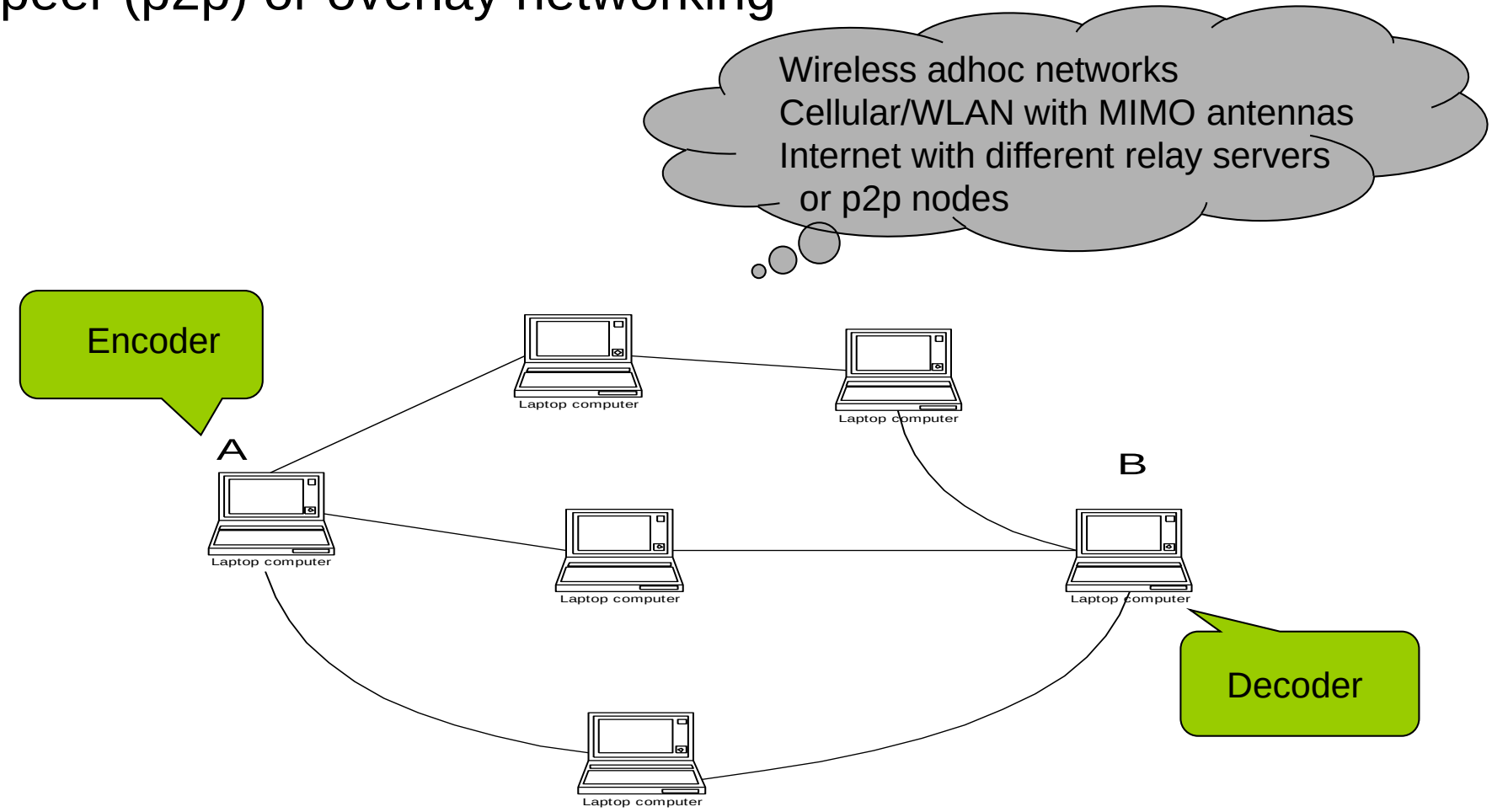
Reference Picture Selection

- Based on the feedback from the decoder, the encoder chooses the reference frame for motion-compensated prediction among the correctly received frames
- Prevents error propagation



Video communications using path diversity

In the Internet, path diversity can be provided via peer-to-peer (p2p) or overlay networking



Why multiple paths?

- Enable sending high bandwidth signal (video) that exceeds the capacity of a single path (e.g. low bandwidth wireless links)
- Can withstand individual path failures in wireless networks or excessive delay due to congestion on a particular path in the Internet
- Enable traffic dispersion and load balancing, which in turn help to reduce congestion and consequently packet losses
- If one path is known to be better than the others, or if it is feasible to set up such a path - use layered/scalable coding
- If all the paths have similar characteristics and no reliable path can be set up - use MDC

