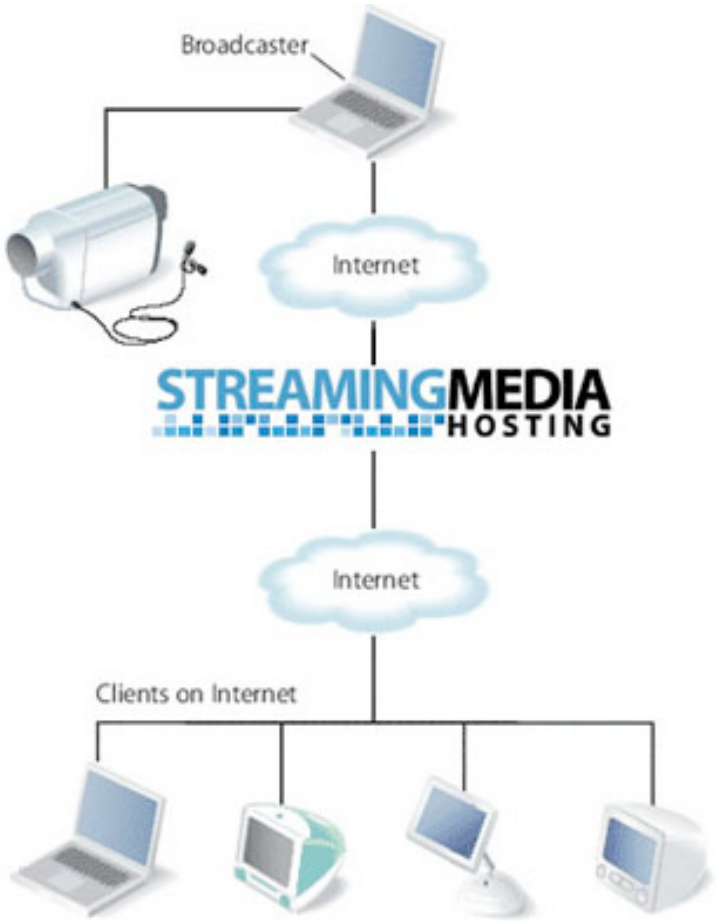
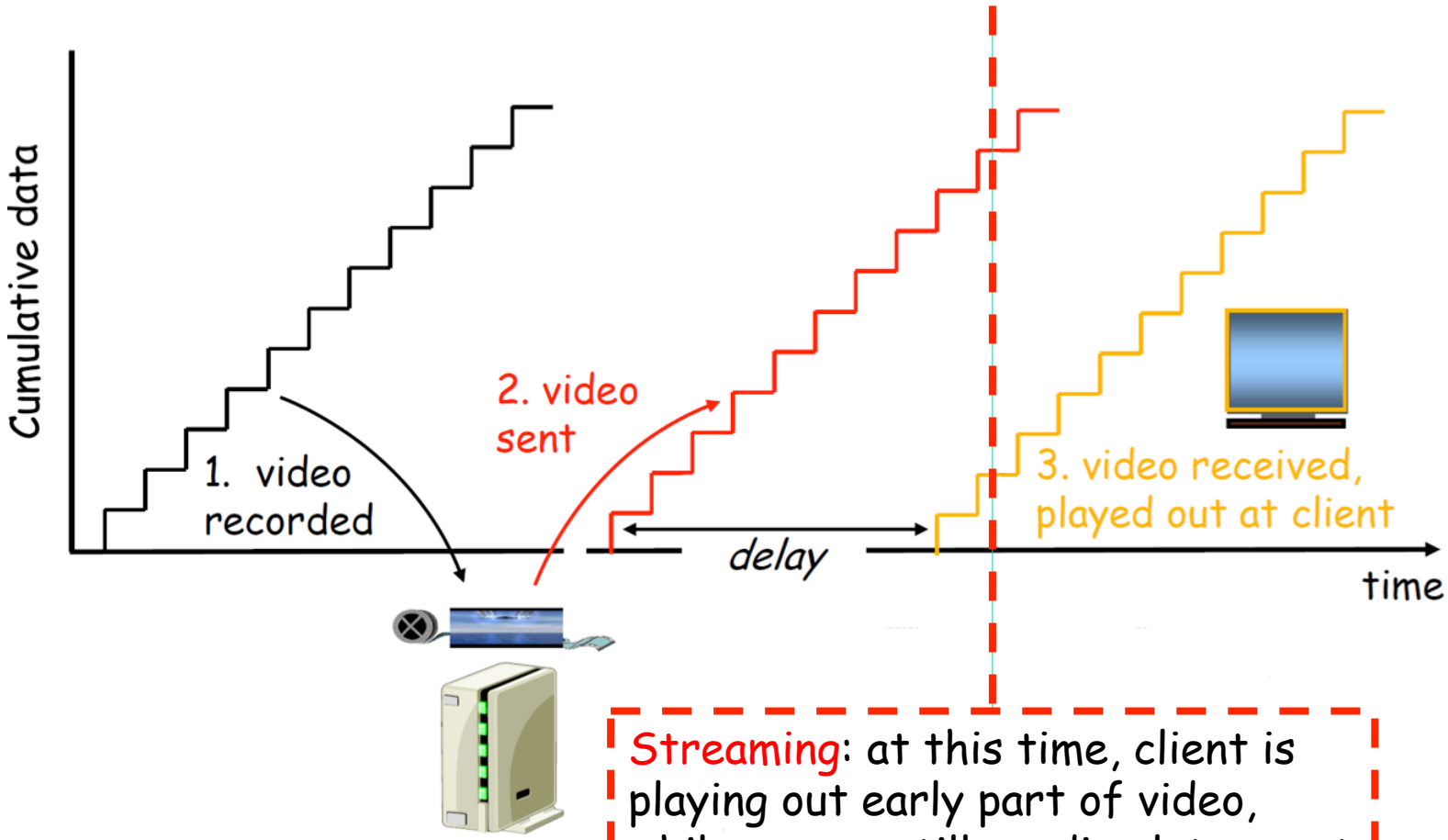
# Streaming media

# Multimedia applications

- Streaming stored audio and video
  - Jitter
- Streaming of live audio and video
- Interactive audio and video
- Delay sensitive and loss tolerant
- Best effort delivery



# Streaming stored multimedia




Streaming: at this time, client is playing out early part of video, while server still sending later part of video.

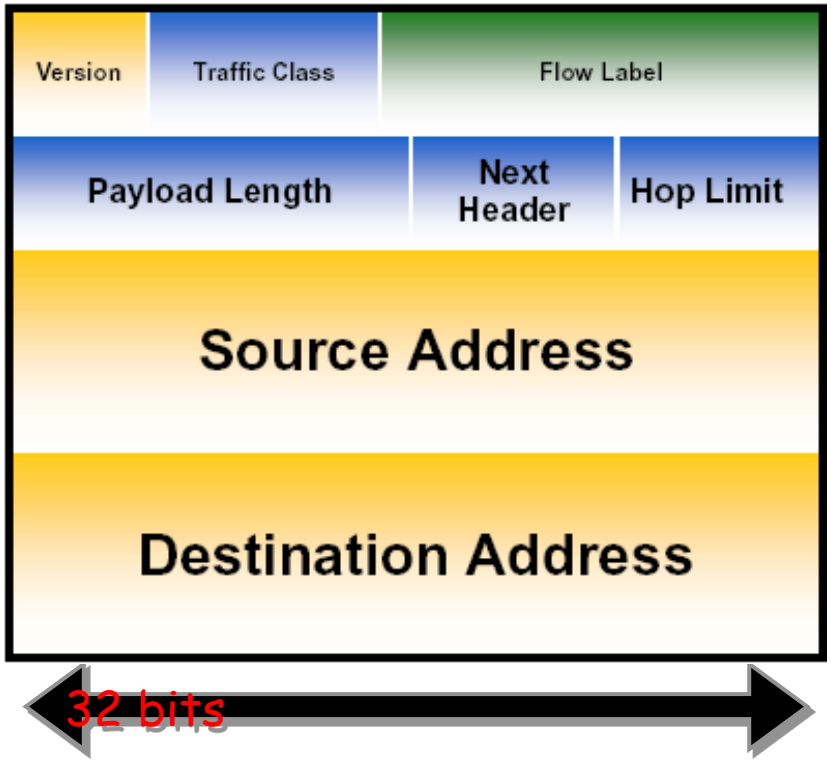
# How should the Internet better support multimedia?

- Laissez-faire
  - no major changes
  - more bandwidth from ISP when needed
  - content distribution (CDN), application-layer multicast
    - application layer
    - Overlay Networks
- Integrated services philosophy
  - Fundamental changes in Internet so that apps can reserve end-to-end bandwidth
  - Requires new, complex software in hosts & routers
- Differentiated services philosophy:
  - Fewer changes to Internet infrastructure, yet provide 1st and 2nd class service.

# IPv6 packet

*Traffic class:* allocate different priority to packets based on applications

*Flow Label:* identify datagrams in same "flow" (concept of "flow" not well defined)




Checksum has been removed entirely to reduce processing at each hop

# Quick exercise

- The lack of _____ is the most common factor that impairs streaming media quality.
  - A. Network delay
  - B. Network bandwidth
  - C. Network security
  - D. Network administration

# Implementing best-effort service

- Some problems to be solved.
  - Limited bandwidth compared to requirements.
    - **360p -> 720p -> 1080p**
  - Random network delays
  - Packet loss

# Compression - reduce bandwidth use

- Digitization -> conversion to bits.
  - = "Encoding"
- Consider problem of encoding an image:
  - 1024x1024 pixels in size.
  - 24 bits per pixel.
  - 3MB per image.
- Consider a video (frame = image):
  - 24 frames per second (at least).
  - 24 frames * 3 Mbytes = 72 MB per second.
  - 72 Mbytes per second = **576 Mbps**.
  - 802.11g gives you a maximum of **54 Mbps**
- **We need compression!!**

# Compression examples

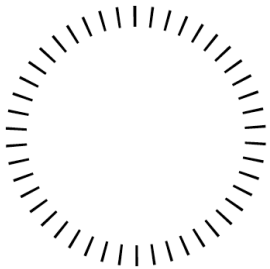
- Audio
  - Encoding techniques
    - Telephone (pulse code modulation) : 64 Kbps
    - CD (pulse code modulation) : 1.411 Mbps for stereo
  - Compression techniques
    - G.729 : 8 Kbps
    - MP3 : 112 or 128 Kbps for CD quality

- Video
  - MPEG2 : 3-6 Mbps for DVD quality
  - MPEG4: high-definition video media like Blu-ray Disc

http://www.dr-lex.be/info-stuff/videocalc.html

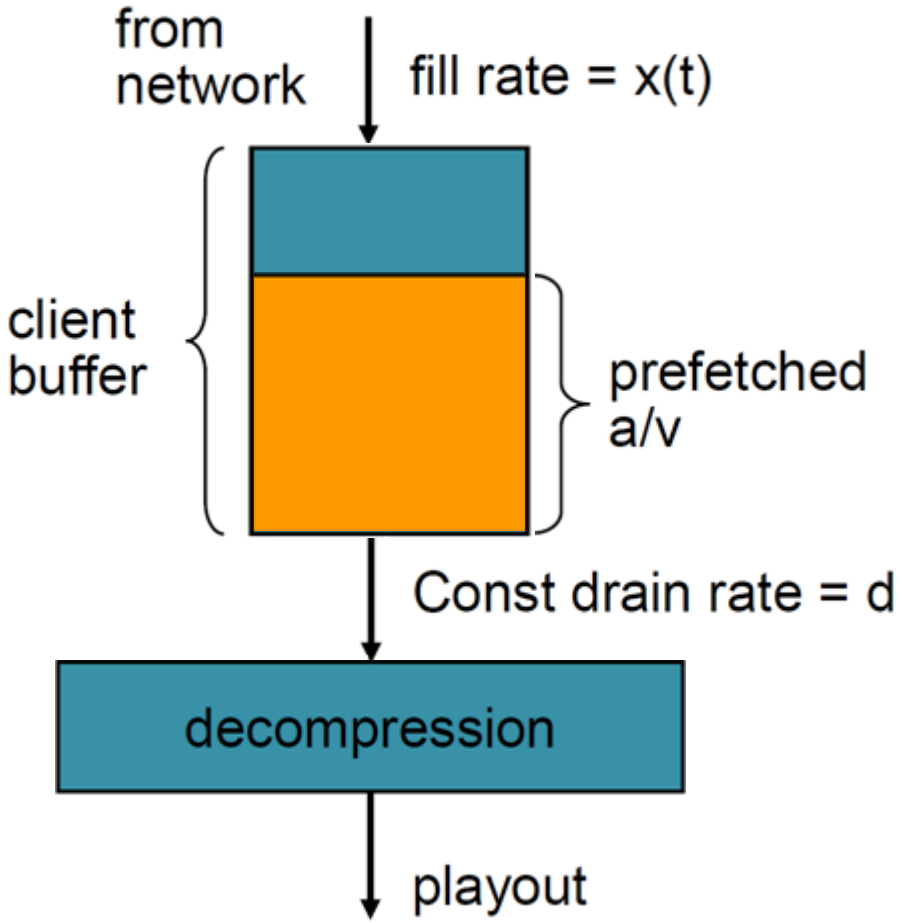
# Quick exercise

- Which of the following statements are correct?
  - A. MPEG4 supports lossless video compression
  - B. MPEG4 supports lossy video compression
  - C. MPEG4 supports lossless audio compression
  - D. MPEG4 supports lossy audio compression

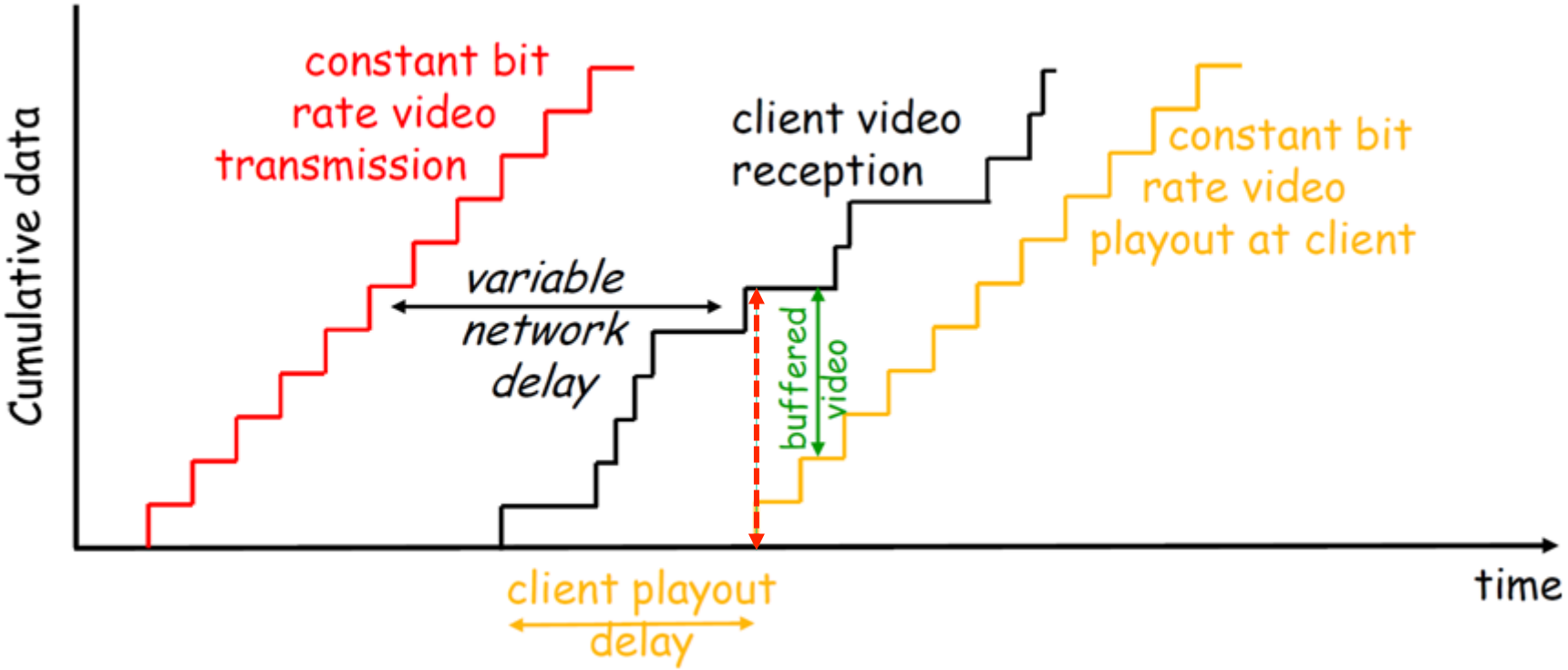


# Client Buffering

- Hide jitter by buffering arriving packets
- We talk about fill rates and playout rates.
- Also supports decompression, can process groups of packets.



# Effect of Client Buffering




- Client-side buffering, play-out delay compensate for network-added delay, delay jitter

# Quick exercise

- Which of the following activities do NOT rely on the playback buffer?
  - A. Reordering incoming chunks of multimedia content
  - B. Decompression
  - C. Cancel the playback process
  - D. Temporarily halt the playback of audio or video clip

# Causes of lost packets

- Use of UDP (best-effort).
- TCP would avoid this, but too slow
- Besides humans can tolerate 1-10% loss!
- So aim is not no loss but to cope with some loss.
  - ***Conceal packet loss***

# Concealing packet loss

- ***Forward error correction***

- Principle:

  *Add redundant information (but not too much)*

- Simple XOR FEC scheme (1990):
  - Break transmission into groups of n packets
  - Every n-th packet, send a special n+1 packet
  - n+1 packet = previous packet XORed together
  - Should receiver lose any of the packets in the group, receiver can recover the missing packet by XORing the special packet with what has been received

# Concealing Packet Loss

***What is sent***

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Packet 1 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 |
| Packet 2 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| XOR packet | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 |

***What is received***

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Packet 1 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 |
| Packet 2 | **LOST!!!** | | | | | | | |
| XOR packet | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 |

***Reconstructing the lost packet***

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Packet 1 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 |
| XOR packet | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 |
| Recovered packet 2 | **0** | **0** | **1** | **1** | **0** | **0** | **0** | **0** |

# Concealing Packet Loss

***What is sent***

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Packet 1 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 |
| Packet 2 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| X | | | | | | | | |
| P | | | | | | | | |
| P | | | | | | | | |
| XOR packet | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 |

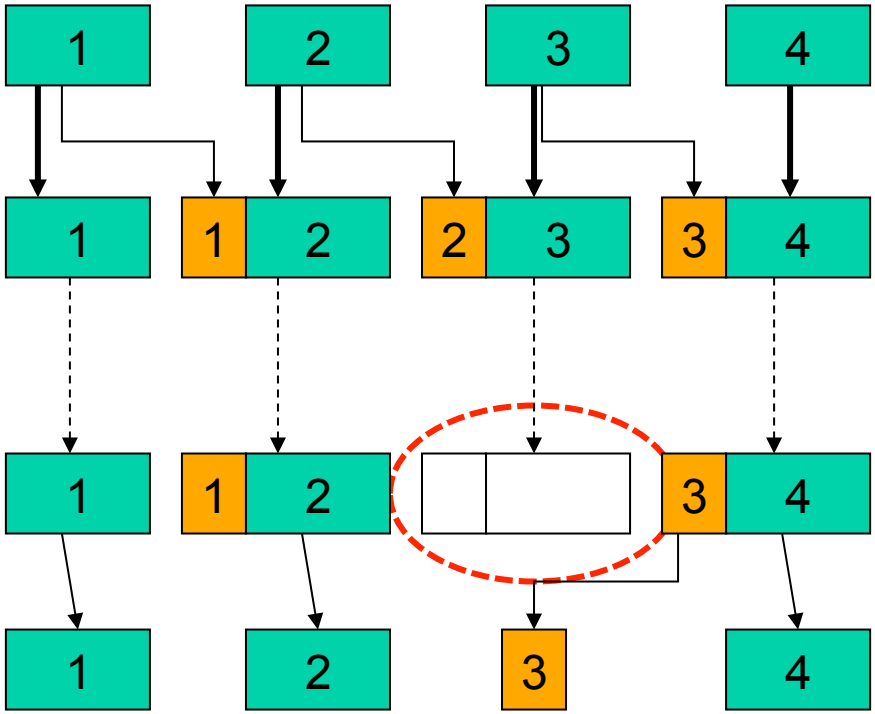
Simple scheme but problems:
(1) What if lose more than one packet per group
(2) What is the effect on playback delay?

***Reconstructing the lost packet***

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Packet 1 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 |
| XOR packet | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 |
| Recovered packet 2 | **0** | **0** | **1** | **1** | **0** | **0** | **0** | **0** |

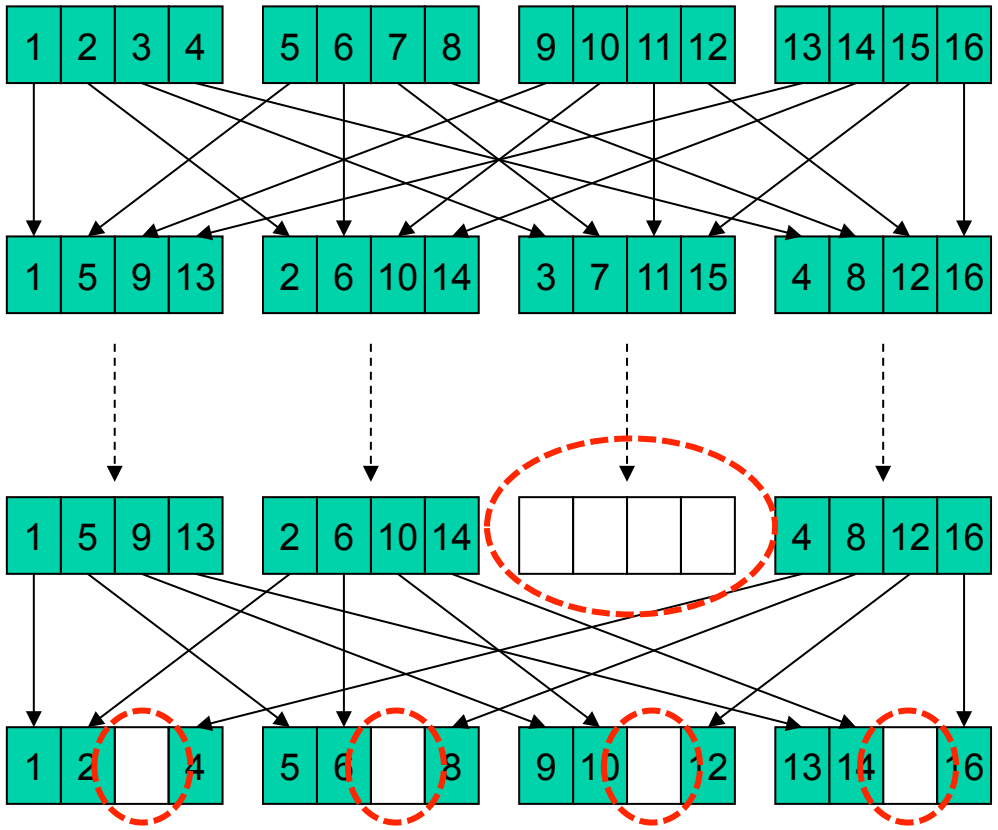
# Concealing packet loss

- **Alternative scheme**
  - High resolution stream
  - Low resolution stream
- Carry redundant data with **every packet**.
- Redundant data is lower quality version of previous packet (for example, lower quality audio).



# Interleaving

- Interleave so that if do lose packets, do not lose adjacent packets.
- **Good**: no extra data is transmitted.
- **Bad**: playback is delayed.



# Summary: bag of tricks for media

- UDP to avoid TCP congestion control (delays) for time-sensitive traffic
- Client-side adaptive playout delay: to compensate for network delay
- Error recovery (on top of UDP)
  - FEC, interleaving
  - retransmissions, time permitting
  - conceal errors: repeat nearby data