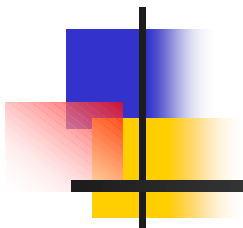


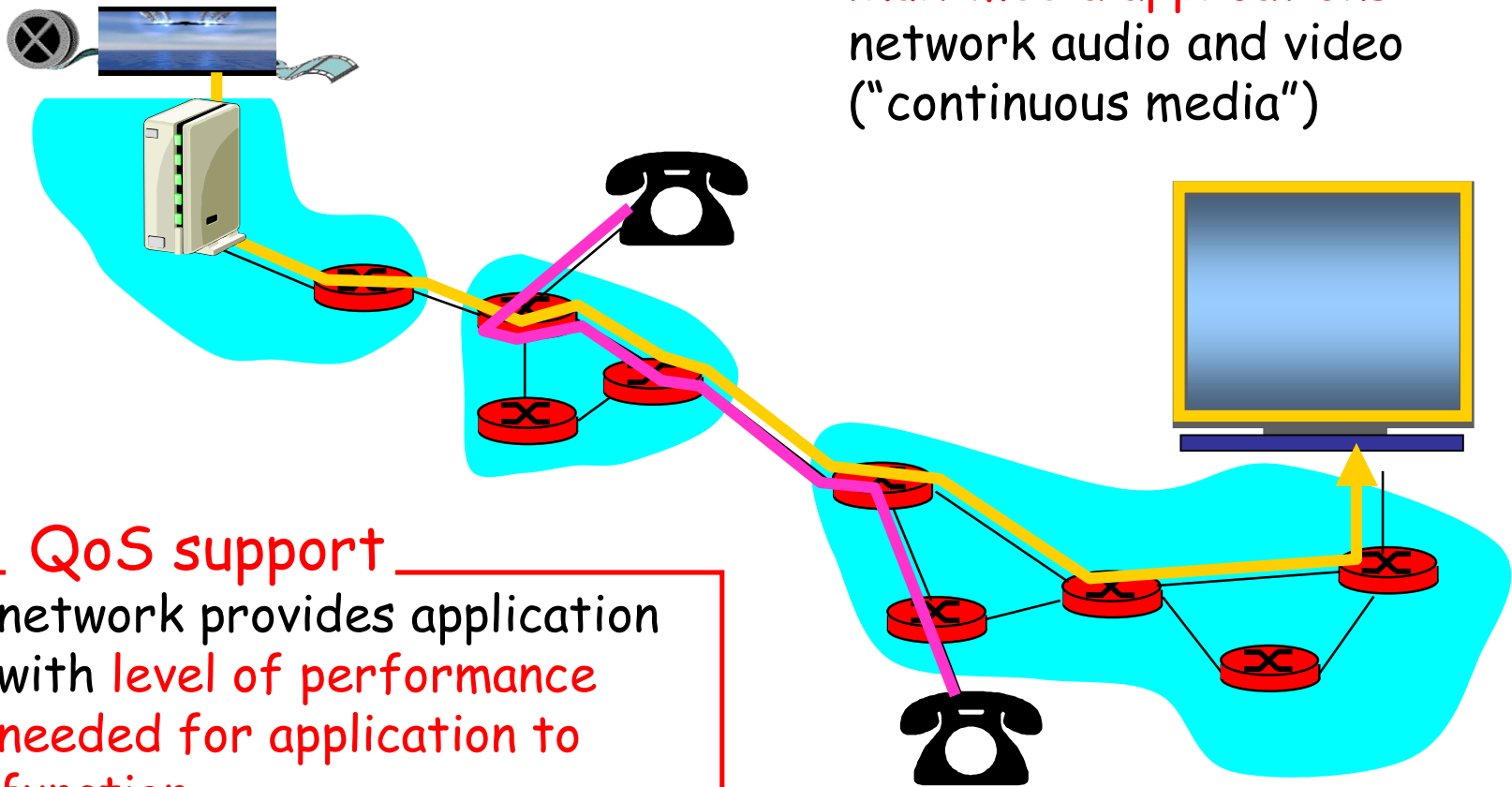
Multimedia Networking



Dept. of Computer Science, University of Rochester

Multimedia and Quality of Service

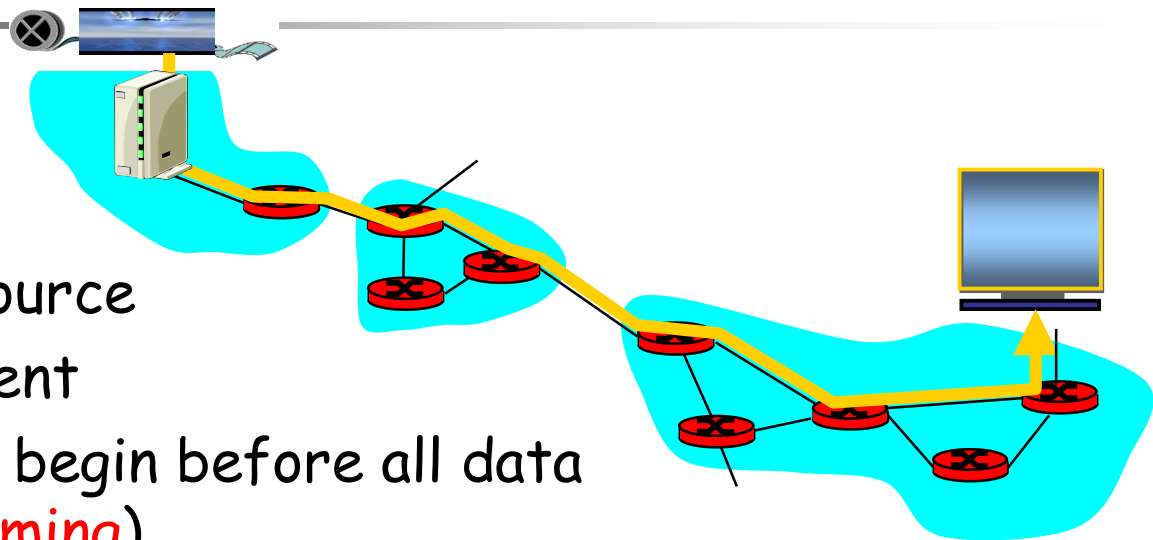
Multimedia applications:
network audio and video
("continuous media")



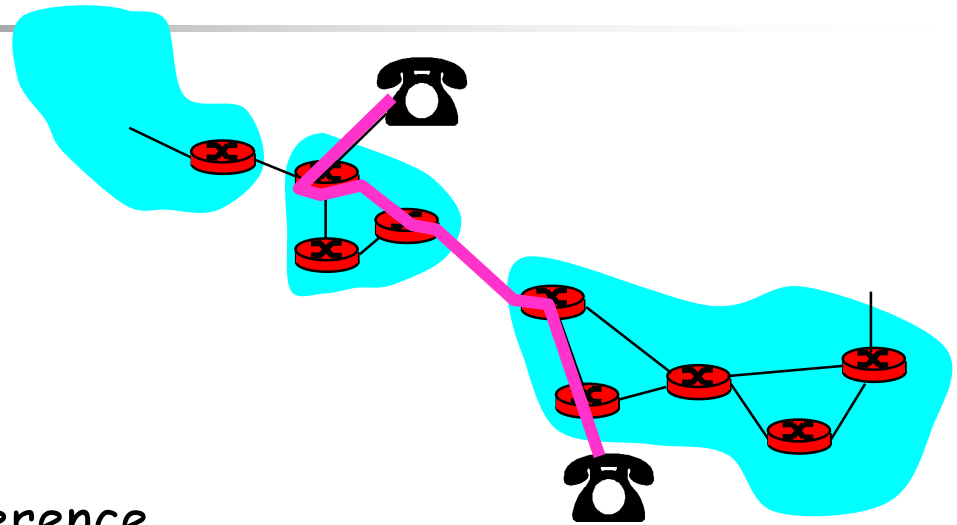
QoS support
network provides application
with level of performance
needed for application to
function.

Stored Multimedia

- media stored at source
- transmitted to client
- client playout may begin before all data has arrived (**streaming**)
 - timing constraint for still-to-be transmitted data: in time for playout
 - sensitive to delay jitter: variability of packet delays within the same packet stream



Real-Time Interactive Multimedia



- **applications:**
 - IP telephony, video conference
- sensitive to both end-to-end delay and jitters
- **end-end delay requirements:**
 - audio: < 150msec good, < 400msec OK
 - higher delays noticeable, impair interactivity
 - includes application-level (compression/decompression) and network delays



Multimedia Networking

Classes of MM applications:

- Stored audio and video
- Real-time interactive audio and video

Fundamental characteristics:

- Typically **delay sensitive**
 - delay jitter
 - end-to-end delay
- But **loss tolerant**: infrequent losses cause minor glitches

QoS support for multimedia networking:

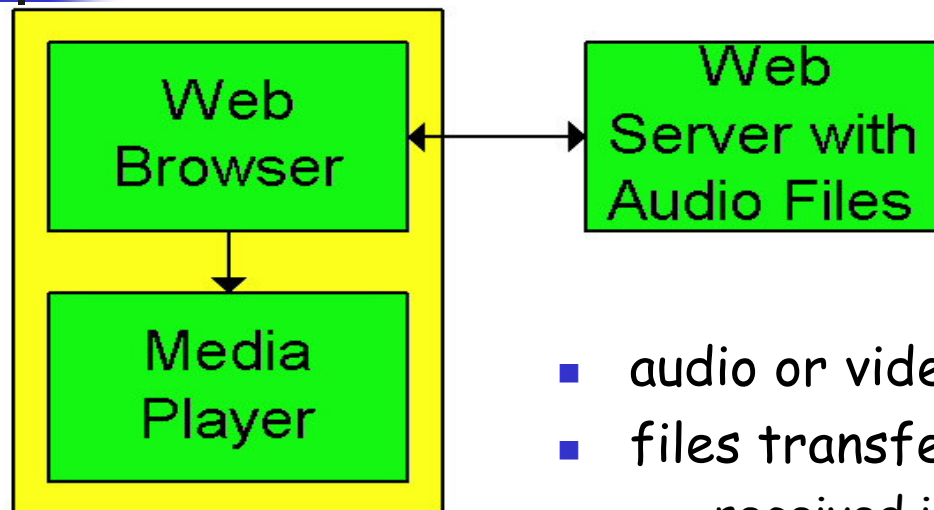
- No direct support from the Internet (best effort, no delay guarantee)
- Employ application techniques to mitigate problems



Outline

- Multimedia applications
- Stored multimedia
- Real-time interactive multimedia
- RTP (Real-time Transport Protocol)

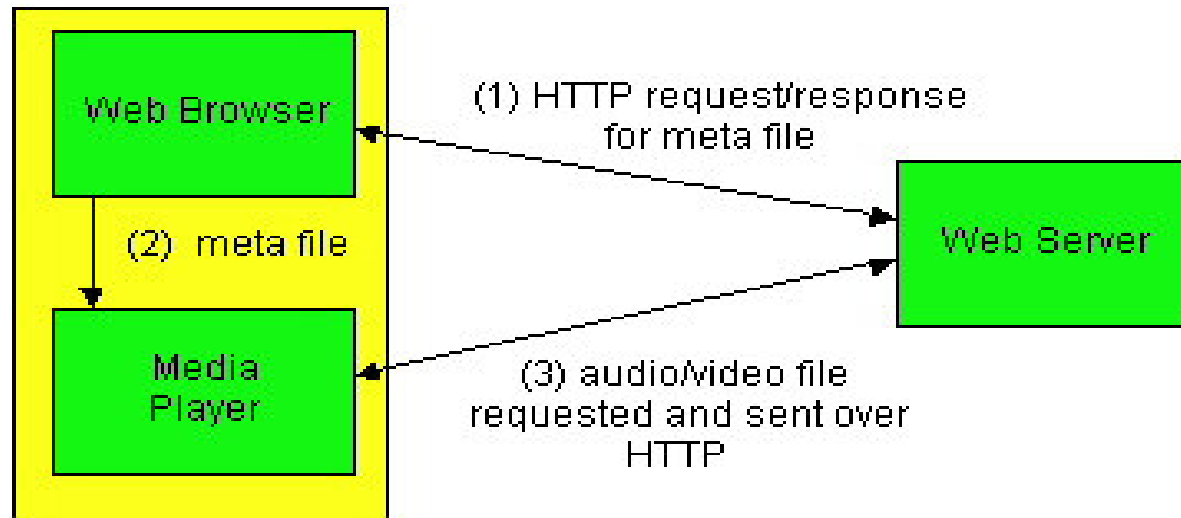
Stored Multimedia over the Internet: the basic approach



client

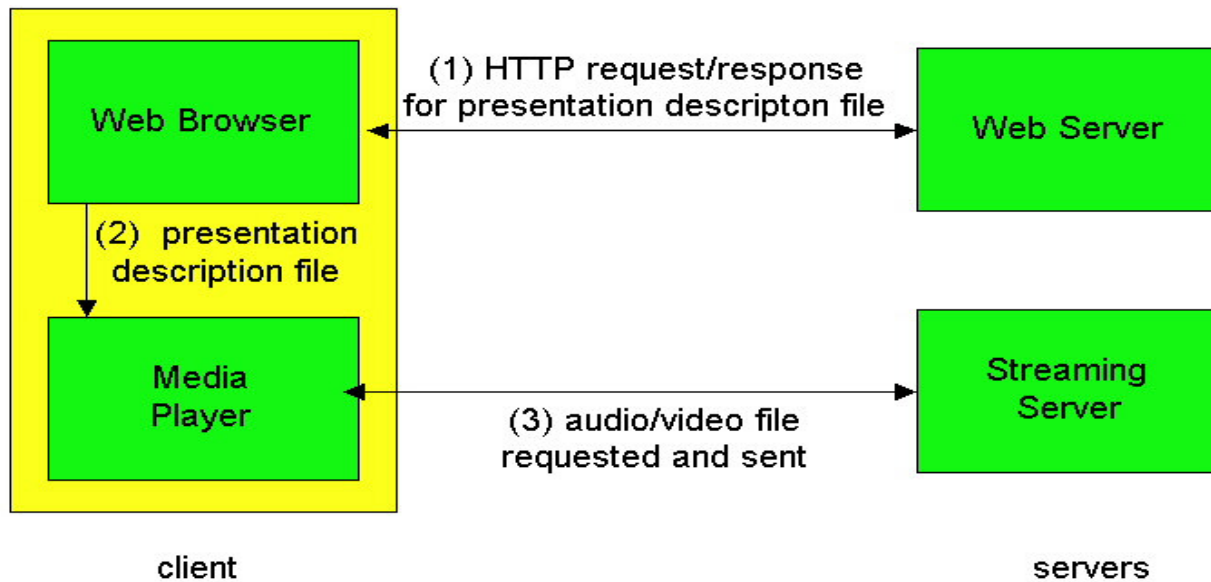
- audio or video stored in files
- files transferred as HTTP object
 - received in entirety at client
 - then passed to player
- media player
 - decompression depending on media format
 - user interface allow interactive control
- long delays until playout!

Stored Multimedia over the Internet: streaming approach



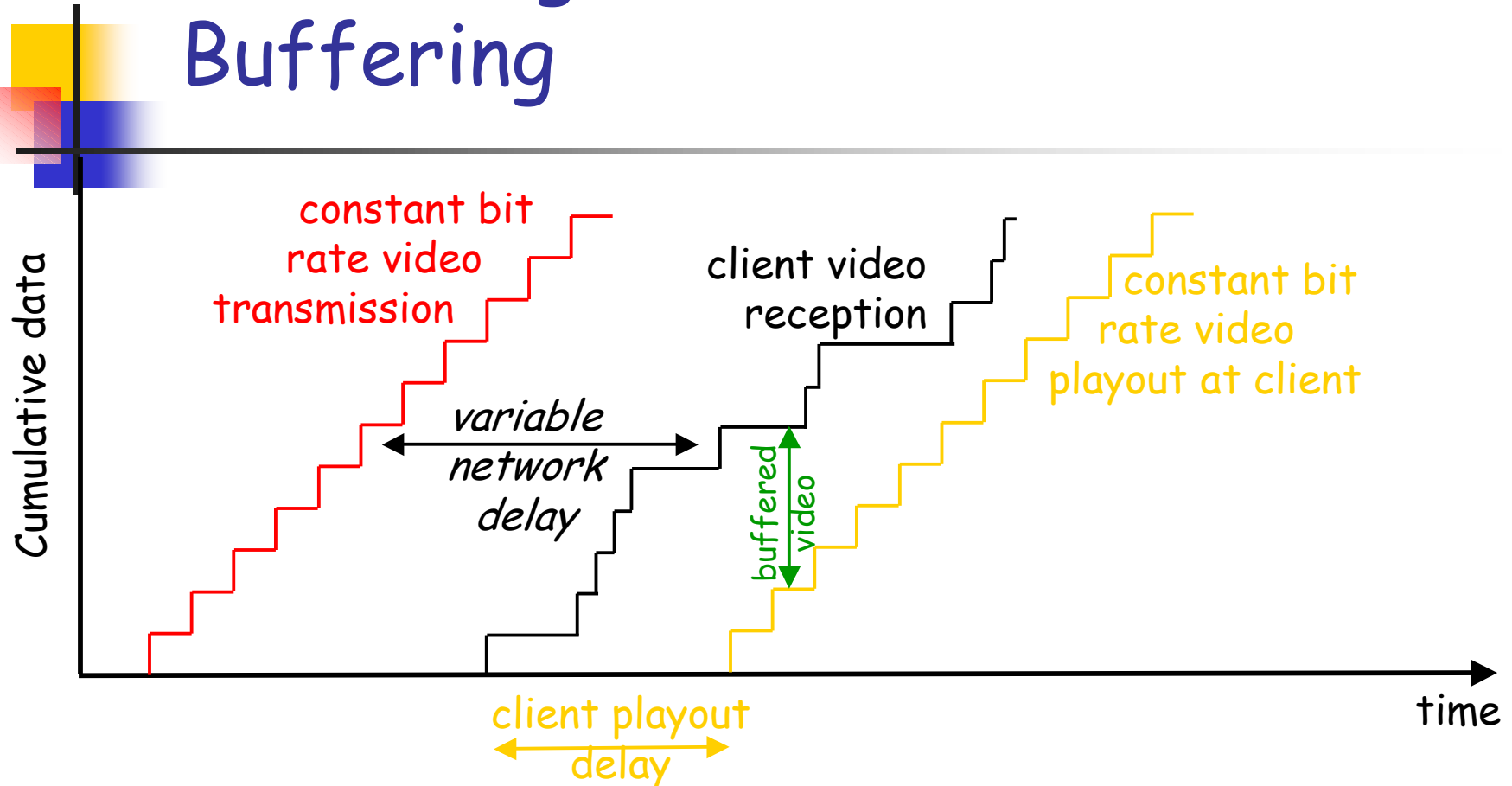
- browser GETs **metafile**
 - location of the media file, type of encoding, ...
- browser launches the media player, passing metafile
- the media player contacts server and the server **streams** audio/video to player
 - pipelining: client playout may begin before all data has arrived

Streaming from a Streaming Server



- This architecture allows for non-HTTP protocol between server and media player
- Can also use UDP instead of TCP.

Streaming Multimedia: Client Buffering



- Client-side buffering, playout delay compensate for network-added delay, delay jitter

Streaming Multimedia: TCP or UDP?



TCP

- send at maximum possible rate under TCP
 - rate fluctuates due to TCP congestion control
 - hard to control

UDP

- server sends at rate appropriate for client (oblivious to network congestion!)
- retransmission of lost packets (at the application-layer) when timing constraints permit
- network unfriendly, unfair to other TCP streams employing congestion control



Outline

- Multimedia applications
- Stored multimedia
- Real-time interactive multimedia
- RTP (Real-time Transport Protocol)



Real-time Interactive Multimedia: Internet Phone

- speaker's audio: alternating talk spurts, silent periods.
 - 64 kbps during talk spurt
- packets generated only during talk spurts
 - 20 msec chunks at 8 Kbytes/sec: 160 bytes data
- application-layer header added to each chunk.
- chunk+header encapsulated into UDP segment.
- application sends UDP segment into socket every 20 msec during talk spurt.



Internet Phone: Playout Delay

- Receiver attempts to playout each chunk exactly q msec after chunk was generated.
 - chunk has time stamp t : play out chunk at $t+q$.
 - chunk arrives after $t+q$: data arrives too late for playout \Rightarrow delay loss
- Tradeoff for q :
 - large q : less packet delay loss
 - small q : better interactive experience
 - typical maximum tolerable delay: 400 ms



Internet Phone: Packet Loss

- **network loss:**
 - IP packet lost due to network congestion (router buffer overflow)
- loss tolerance: depending on voice encoding, packet loss rates between 1% and 10% can be tolerated
- beyond that?
 - small tolerable playout delay prohibits retransmission
 - we need loss concealments - recover from packet loss without retransmission



Recovery from Packet Loss (1)

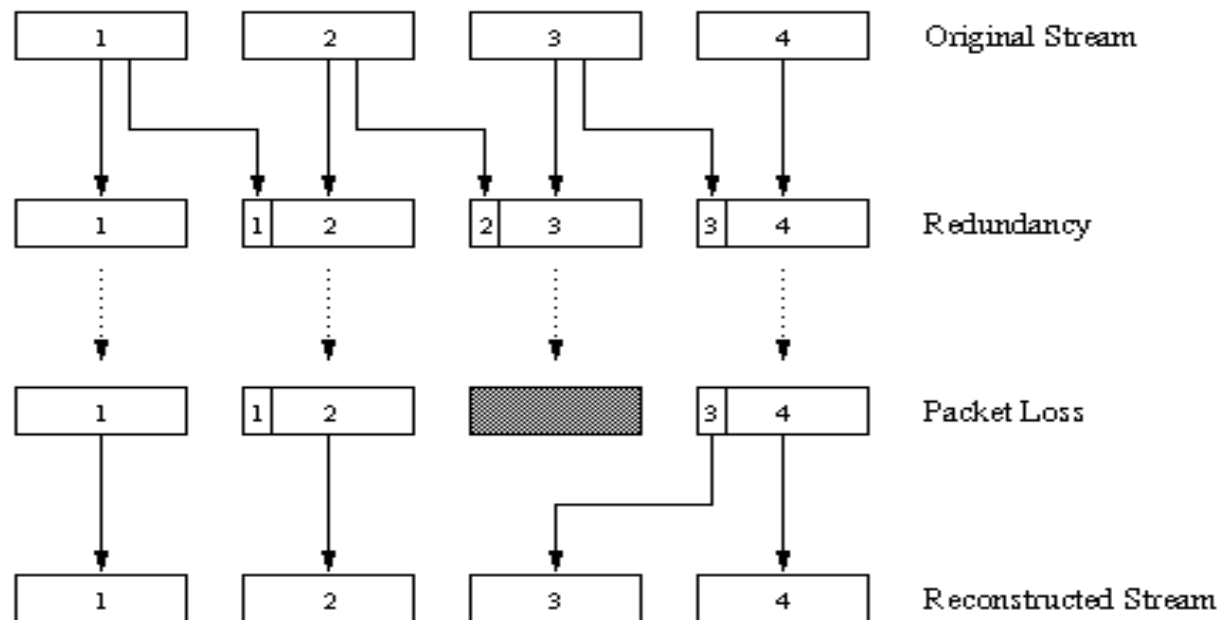
Forward Error Correction (FEC):

- for every group of n chunks create a redundant chunk by exclusive OR-ing the n original chunks
- send out $n+1$ chunks, increasing the bandwidth need by $1/n$
- can reconstruct the original n chunks if there is at most one lost chunk from the $n+1$ chunks
- playout delay needs to be long enough for the client receiving all $n+1$ packets

Impact of a larger n :

- less bandwidth waste
- longer playout delay
- higher probability that 2 or more chunks will be lost

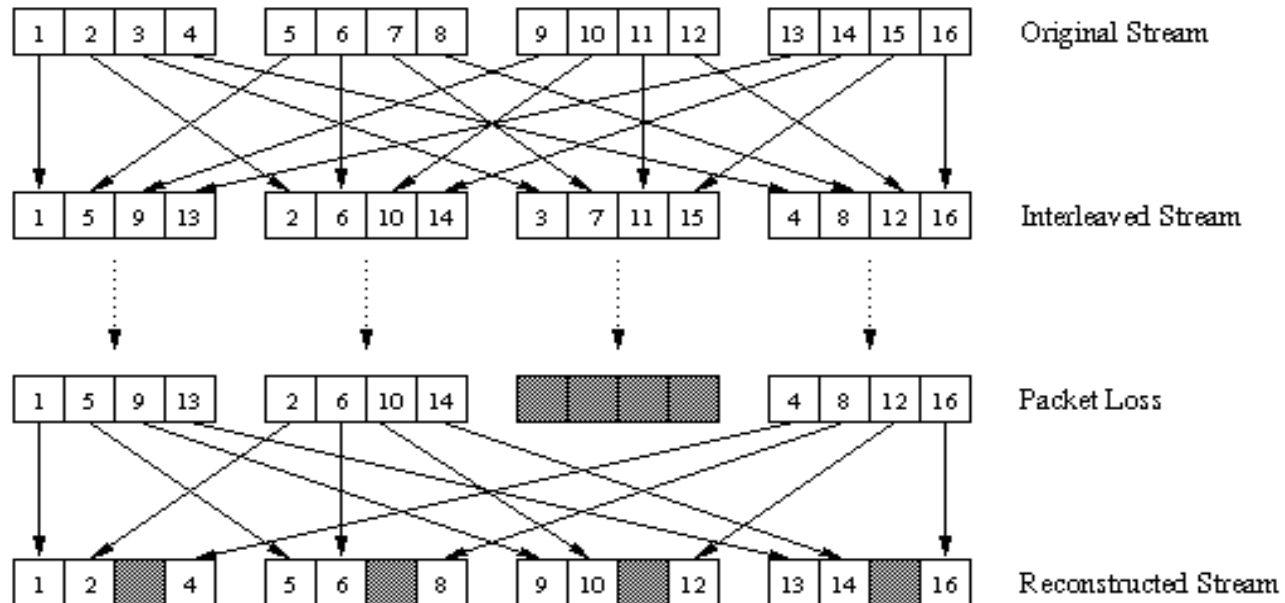
Recovery from Packet Loss (2)



piggyback lower quality stream:

- send lower resolution audio stream as the redundant information
- the receiver can conceal non-consecutive loss
- What is the playout delay?

Recovery from Packet Loss (3)



Interleaving

- chunks are broken up into smaller units (e.g. four 5 msec units per chunk)
- Packet contains small units from different chunks
- if packet is lost, still have most of every chunk
- has no redundancy overhead, but adds to playout delay



Streaming Multimedia on the Internet: Summary

- use UDP to avoid TCP congestion control (delays) for time-sensitive traffic
- compensate delays and jitters:
 - client-side buffering
- error recovery (on top of UDP)
 - retransmissions, time permitting (for stored multimedia)
 - conceal errors: FEC, piggybacking, interleaving (for real-time interactive multimedia)

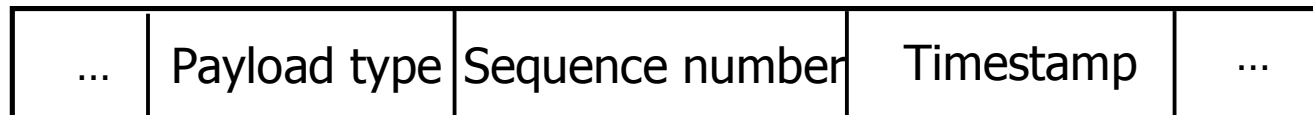


Real-time Transport Protocol (RTP)

- Real-time transport protocol
 - RTP specifies a packet structure for packets carrying audio and video data
 - RTP packets are encapsulated in UDP segments
 - RTP understood/supported in the end systems
- Interoperability:
 - If two Internet phone applications run RTP, then they can understand each other's data



RTP Header



- **Payload Type (7 bits):** Indicates type of encoding currently being used.
- **Sequence Number (16 bits):** Increments by one for each RTP packet sent, and may be used to detect packet loss and to restore packet sequence.
- **Timestamp field (32 bits):** Reflects the sampling instant of the first byte in the RTP data packet.



Disclaimer

- Parts of the lecture slides contain original work of James Kurose, Larry Peterson, and Keith Ross. The slides are intended for the sole purpose of instruction of computer networks at the University of Rochester. All copyrighted materials belong to their original owner(s).