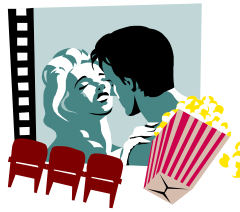
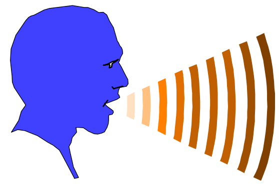
   

Multimedia Communication Systems II

Multimedia Networking

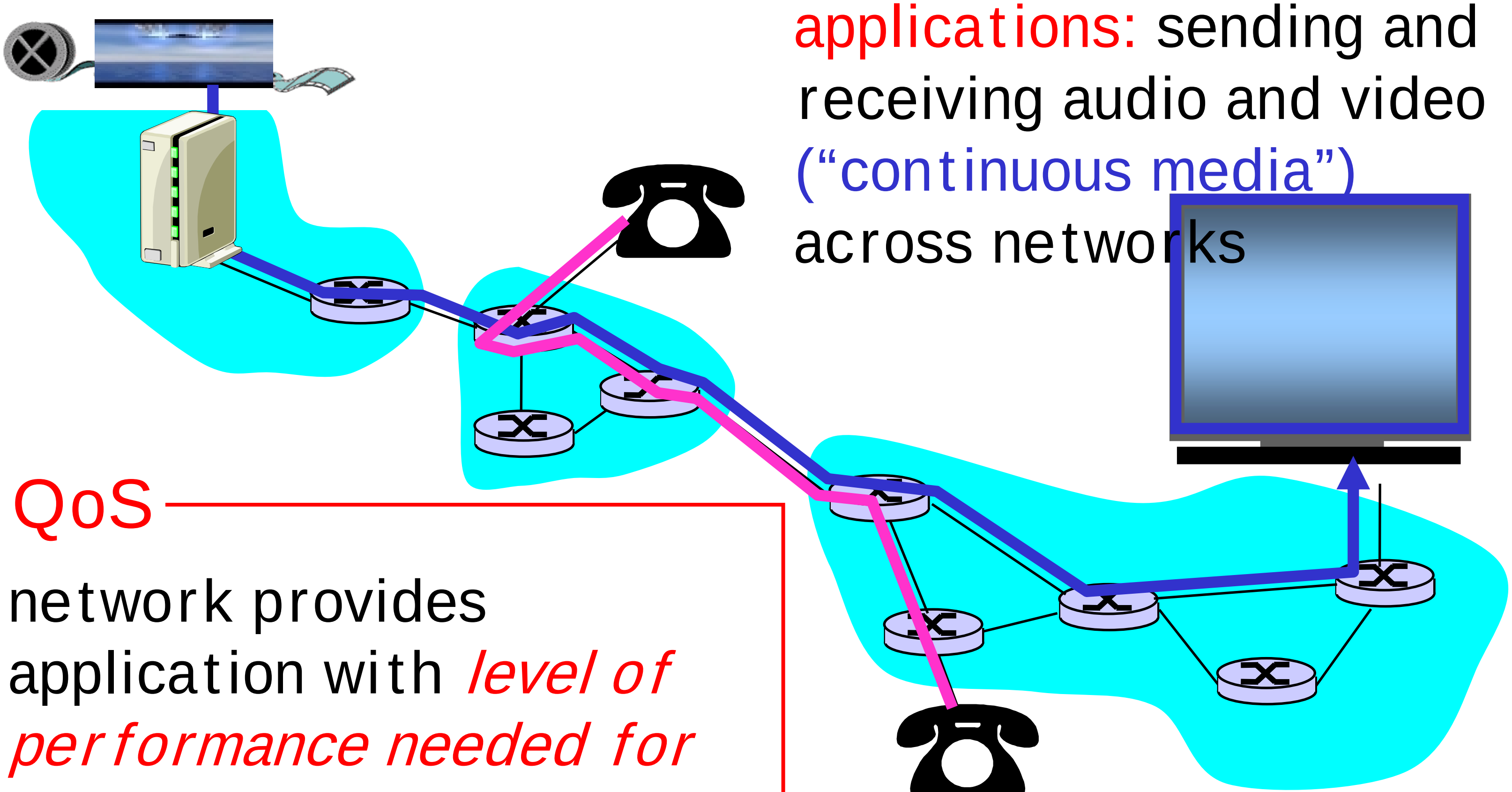
Yao Wang
Polytechnic University, Brooklyn, NY11201
yao@vision.poly.edu

These slides are adapted from the slides made by authors of the book (J. F. Kurose and K. Ross), available from the publisher site for instructors. We would like to thank the authors for the excellent book and the slides.

Slides based on Computer Networking: A Top Down Approach Featuring the Internet,
2nd edition.
Jim Kurose, Keith Ross
Addison-Wesley, July 2002.
Chapter 6

Multimedia, Quality of Service: What is it?

Network Multimedia applications: sending and receiving audio and video ("continuous media") across networks



QoS

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

Roadmap

- Multimedia Networking Applications
- RTP and RTCP
- Streaming stored audio and video
  - RTSP
- Internet video phone
  - SIP
  - SIP vs. H.323
- Recovery from loss
- Beyond Best Effort

MM Networking Applications

Classes of MM applications:

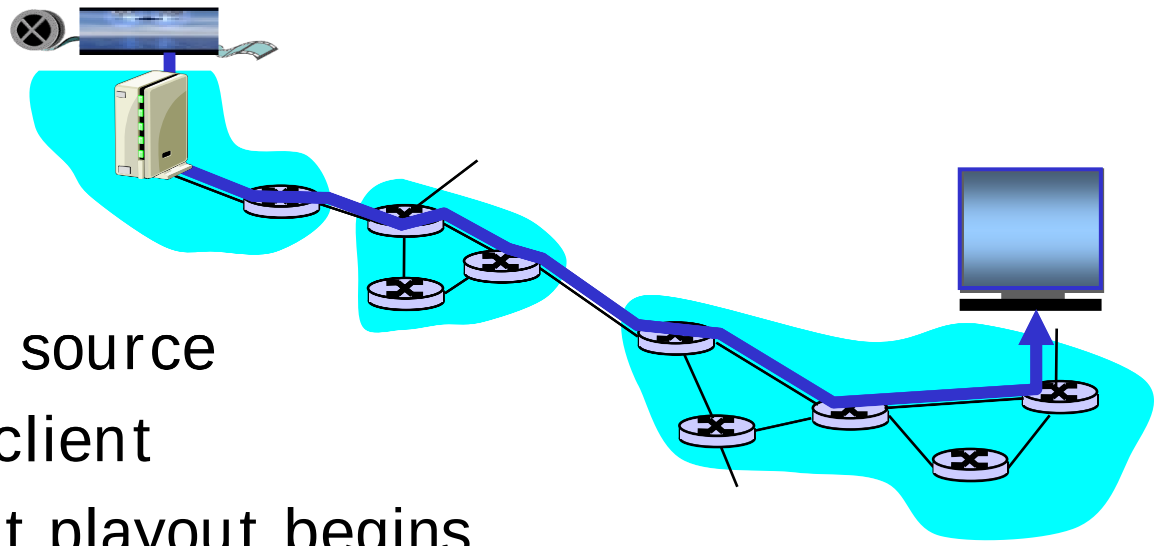
1) Streaming stored audio and video
2) Streaming live audio and video
3) Real-time interactive audio and video

Jitter is the variability of packet delays within the same packet stream

Fundamental characteristics:

- Typically **delay sensitive**
  - end-to-end delay
  - delay jitter
- But **loss tolerant**: infrequent losses cause minor glitches
- Antithesis of data, which are loss intolerant but delay tolerant.

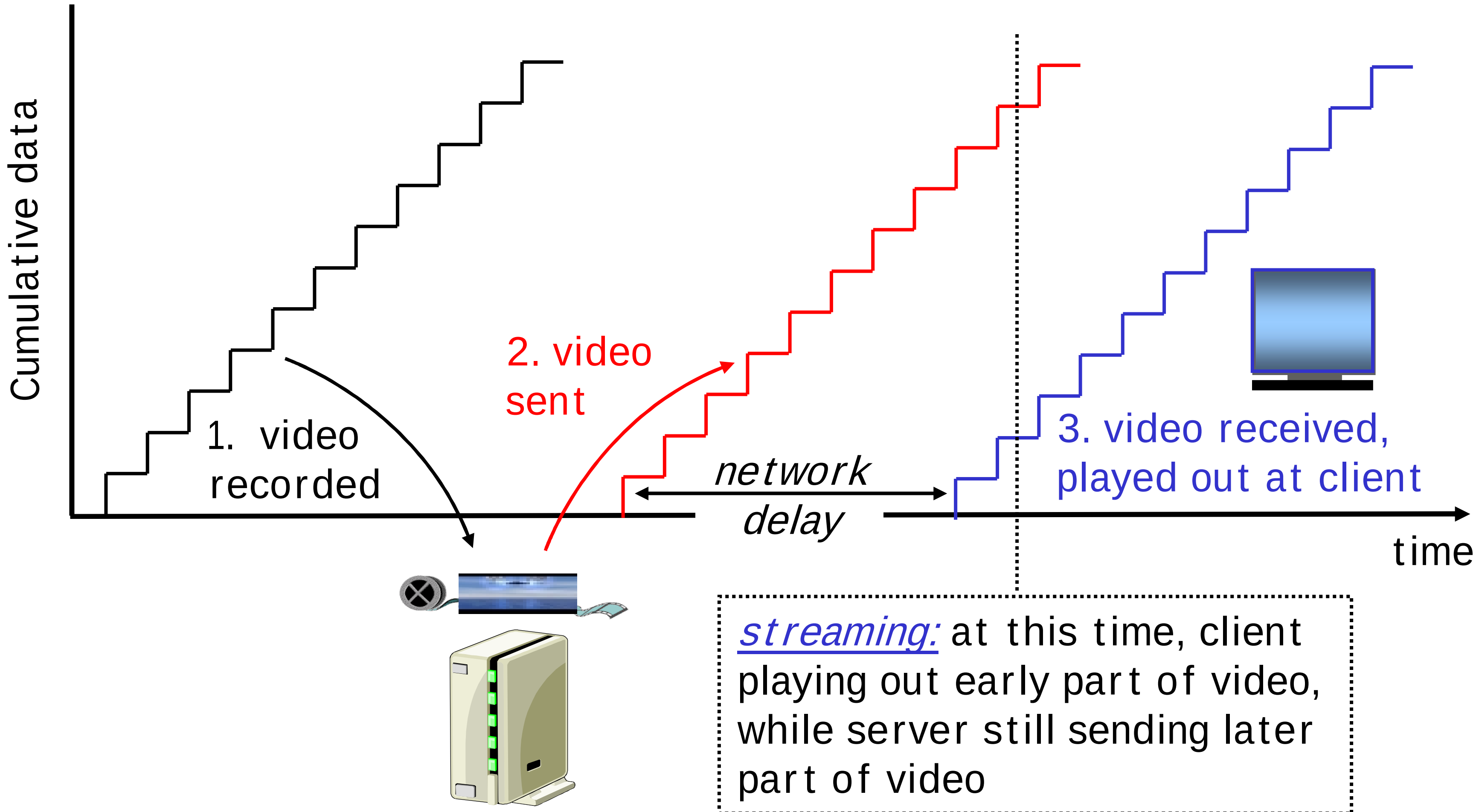
Streaming Stored Multimedia



Streaming:

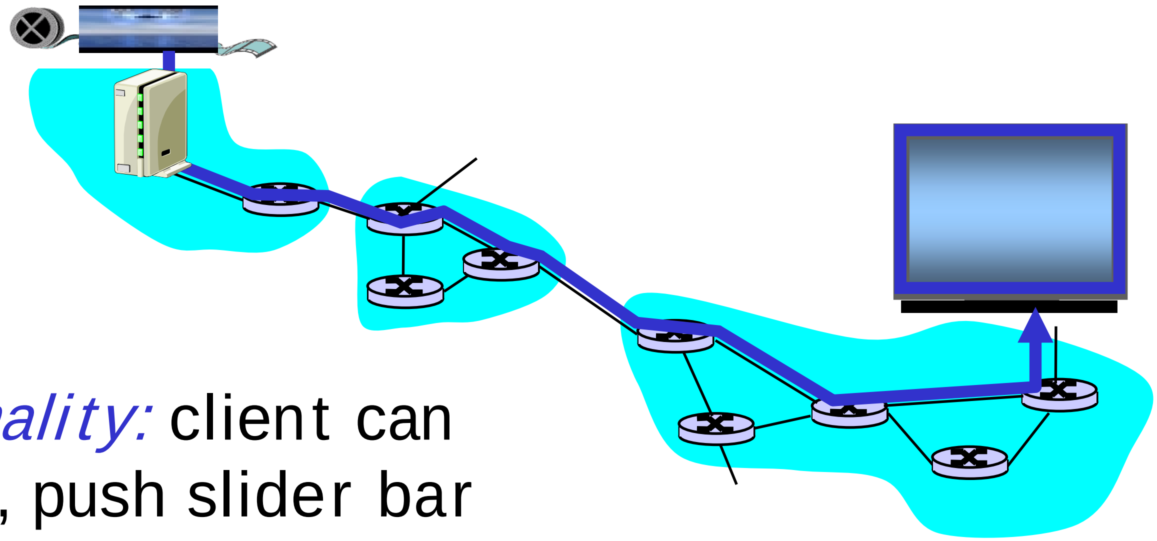
- media stored at source
- transmitted to client
- streaming: client playout begins *before* all data has arrived
  - timing constraint for still-to-be transmitted data: in time for playout

Streaming Stored Multimedia: What is it?



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

Streaming Stored Multimedia: Interactivity



- *VCR-like functionality:* client can pause, rewind, FF, push slider bar
  - 10 sec initial delay OK
  - 1-2 sec until command effect OK
  - RTSP often used (more later)
- timing constraint for still-to-be transmitted data: in time for playout

Streaming Live Multimedia

Examples:

- Internet radio talk show
- Live sporting event

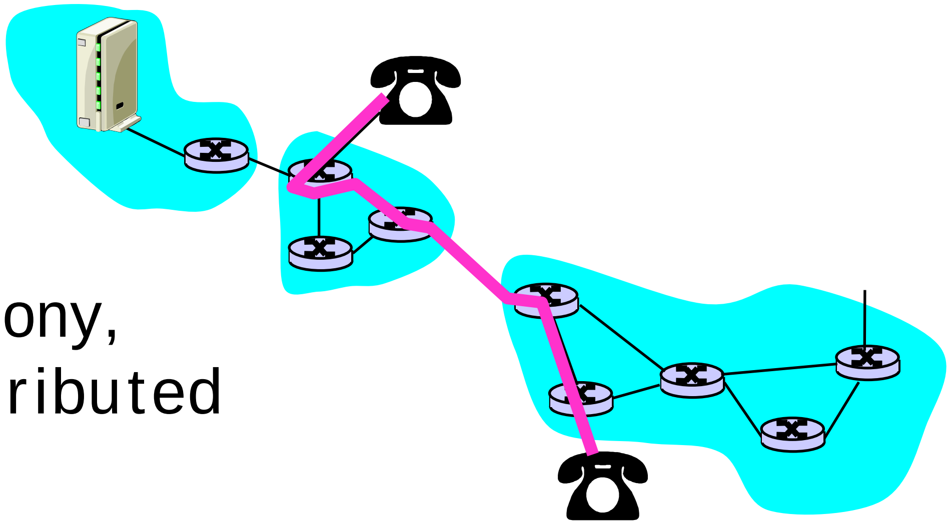
Streaming

- playback buffer
- playback can lag tens of seconds after transmission
- still have timing constraint

Interactivity

- fast forward impossible
- rewind, pause possible!

Interactive, Real-Time Multimedia

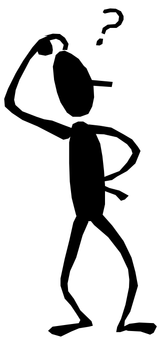


- applications: IP telephony, video conference, distributed interactive worlds
- end-end delay requirements:
  - audio: < 150 msec good, < 400 msec OK
    - includes application-level (packetization) and network delays
    - higher delays noticeable, impair interactivity
- session initialization
  - how does callee advertise its IP address, port number, encoding algorithms?

Multimedia Over Today's Internet

TCP/UDP/IP: "best-effort service"

- *no* guarantees on delay, loss



? ? ? ? ? ?

But you said multimedia apps requires ?
QoS and level of performance to be
? ? effective! ? ?

> Today's Internet multimedia applications use application-level techniques to mitigate (as best possible) effects of delay, loss

Roadmap

- Multimedia Networking Applications
- RTP and RTCP
- Streaming stored audio and video
  - RTSP
- Internet Phone
  - SIP
  - SIP vs. H.323
- Recovery from loss
- Beyond Best Effort

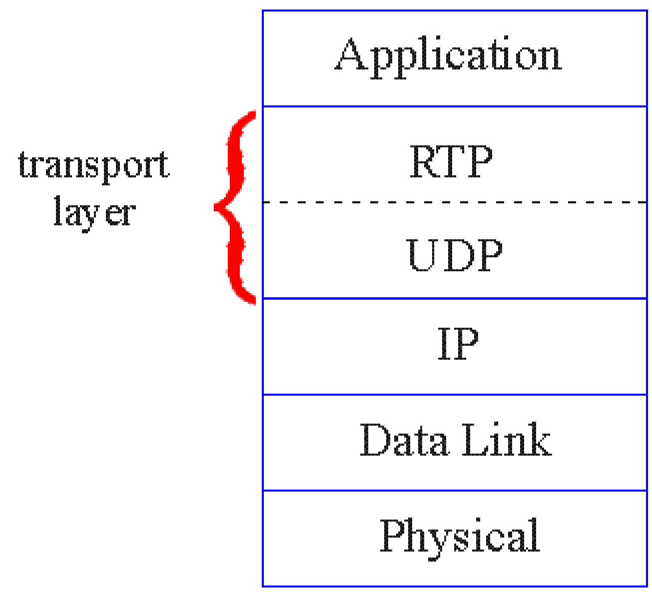
Real-Time Protocol (RTP)

- RTP specifies a packet structure for packets carrying audio and video data – a packetization protocol!
- RFC 1889.
- RTP packet provides
  - payload type identification
  - packet sequence numbering
  - timestamping
- RTP runs in the end systems.
- RTP packets are encapsulated in UDP segments
- Interoperability: If two Internet phone applications run RTP, then they may be able to work together

RTP runs on top of UDP

RTP libraries provide a transport-layer interface that extend UDP:

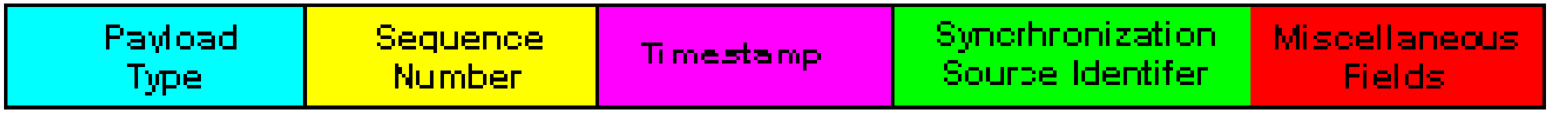
- port numbers, IP addresses
- payload type identification
- packet sequence numbering
- time-stamping



RTP Example

- Consider sending 64 kbps PCM-encoded voice over RTP.
- Application collects the encoded data in chunks, e.g., every 20 msec = 160 bytes in a chunk.
- The audio chunk along with the RTP header form the RTP packet, which is encapsulated into a UDP segment.
- RTP header indicates type of audio encoding in each packet
  - sender can change encoding during a conference.
- RTP header also contains sequence numbers and timestamps.

RTP Header



RTP Header

Payload Type (7 bits): Indicates type of encoding currently being used. If sender changes encoding in middle of conference, sender informs the receiver through this payload type field.

- Payload type 0: PCM mu-law, 64 kbps
- Payload type 3, GSM, 13 kbps
- Payload type 7, LPC, 2.4 kbps
- Payload type 26, Motion JPEG
- Payload type 31. H.261
- Payload type 33, MPEG2 video

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.

RTP Header (2)

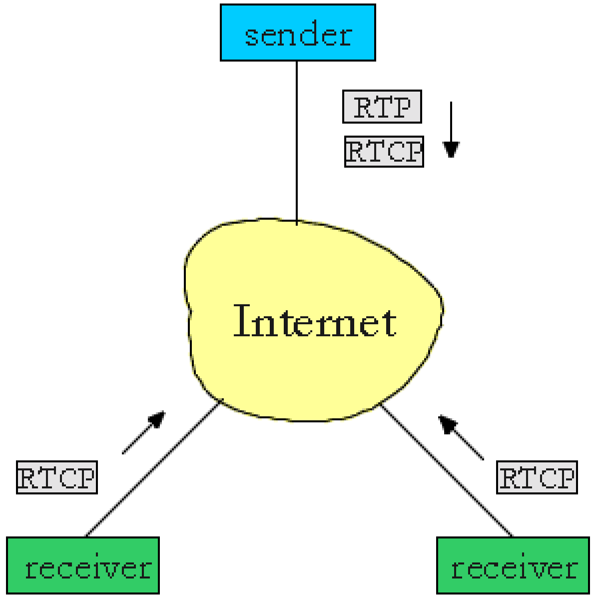
- **Timestamp field (32 bytes long).** Reflects the sampling instant of the first byte in the RTP data packet.
  - For audio, timestamp clock typically increments by one for each sampling period (for example, each 125 usecs for a 8 KHz sampling clock)
  - if application generates chunks of 160 encoded samples, then timestamp increases by 160 for each RTP packet when source is active. Timestamp clock continues to increase at constant rate when source is inactive.

- **SSRC field (32 bits long).** Identifies the source of the RTP stream. Each stream in a RTP session should have a distinct SSRC.

Real-Time Control Protocol (RTCP)

- Works in conjunction with RTP.
- Each participant in RTP session periodically transmits RTCP control packets to all other participants.
- Each RTCP packet contains sender and/or receiver reports
  - report statistics useful to application
- Statistics include number of packets sent, number of packets lost, interarrival jitter, etc.
- Feedback can be used to control performance
  - Sender may modify its transmissions based on feedback

RTCP - Continued



- For an RTP session there is typically a single multicast address; all RTP and RTCP packets belonging to the session use the multicast address.

- RTP and RTCP packets are distinguished from each other through the use of distinct port numbers.

- To limit traffic, each participant reduces his RTCP traffic as the number of conference participants increases.

RTCP Packets

Receiver report packets:

- fraction of packets lost, last sequence number, average interarrival jitter.

Sender report packets:

- SSRC of the RTP stream, the current time, the number of packets sent, and the number of bytes sent.

Source description packets:

- e-mail address of sender, sender's name, SSRC of associated RTP stream.
- Provide mapping between the SSRC and the user/host name.

Synchronization of Streams

- RTCP can synchronize different media streams within a RTP session.
- Consider videoconferencing app for which each sender generates one RTP stream for video and one for audio.
- Timestamps in RTP packets tied to the video and audio sampling clocks
  - not tied to the wall-clock time
- Each RTCP sender-report packet contains (for the most recently generated packet in the associated RTP stream):
  - timestamp of the RTP packet
  - wall-clock time for when packet was created.
- Receivers can use this association to synchronize the playout of audio and video.

RTP and QoS

- RTP does **not** provide any mechanism to ensure timely delivery of data or provide other quality of service guarantees.
- But together with RTCP, it allows monitoring of QoS so that sender and receiver can adjust their operations appropriately
- RTP encapsulation is only seen at the end systems: it is not seen by intermediate routers.
  - Routers providing best-effort service do not make any special effort to ensure that RTP packets arrive at the destination in a timely matter.

Roadmap

- Multimedia Networking Applications
- RTP and RTCP
- Streaming stored audio and video
  - RTSP
- Internet Phone
  - SIP
  - SIP vs. H.323
- Recovery from loss
- Beyond Best Effort

Streaming Stored Multimedia

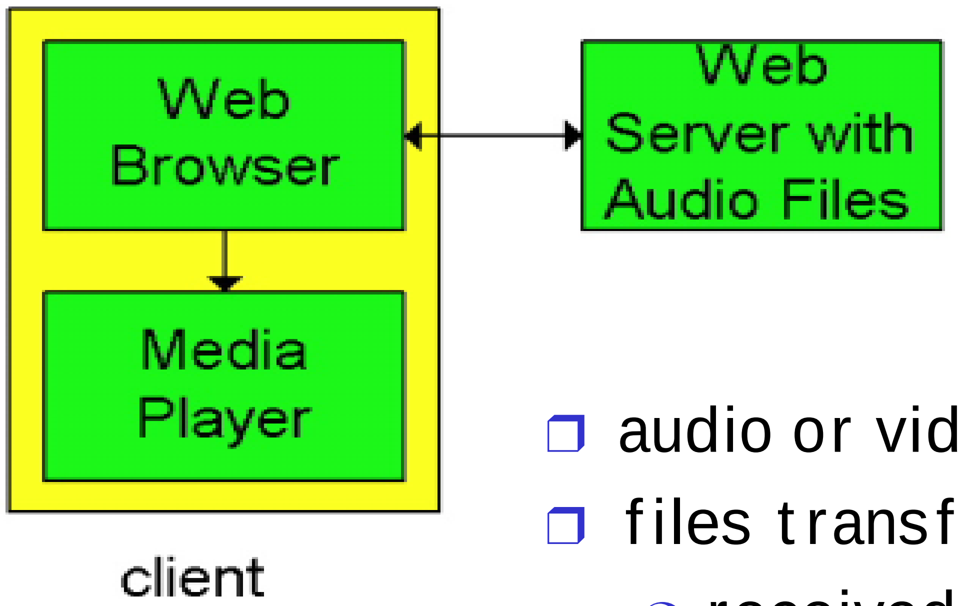
Application-level streaming techniques for making the best out of best effort service:

- client side buffering
- use of RTP/UDP or UDP directly versus TCP
- multiple encodings of multimedia

Media Player

- jitter removal
- decompression
- error concealment
- graphical user interface w/ controls for interactivity

Internet multimedia: simplest approach

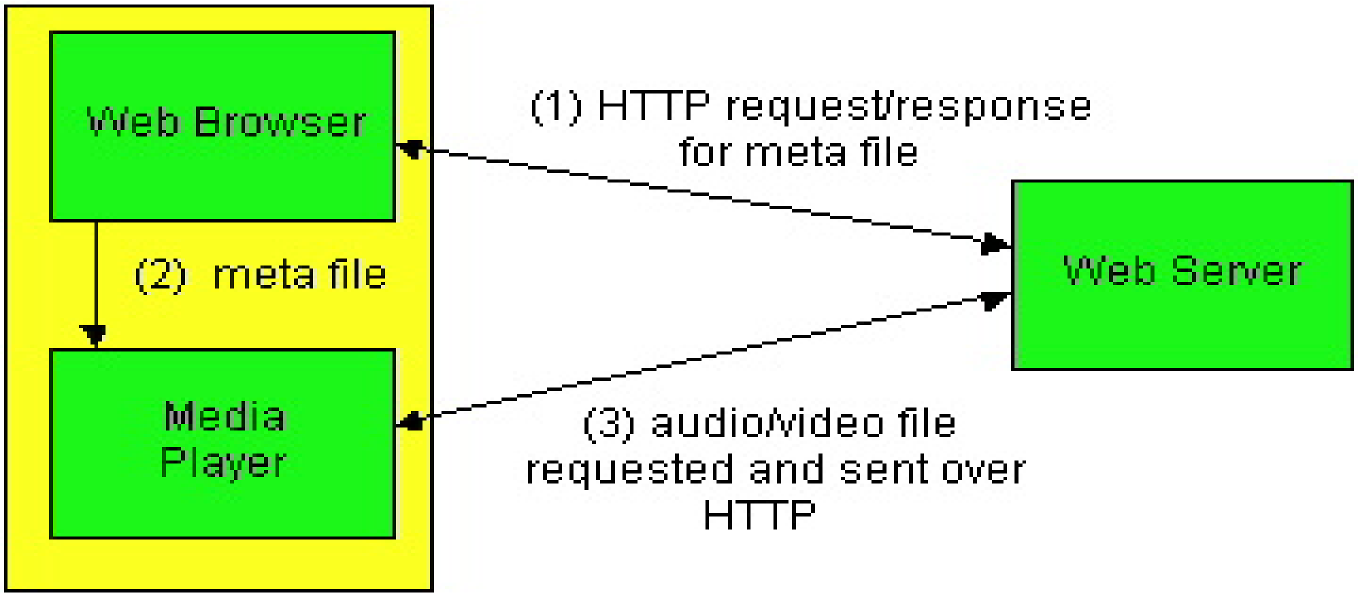


- audio or video stored in file
- files transferred as HTTP object
  - received in entirety at client
  - then passed to player

audio, video not streamed:

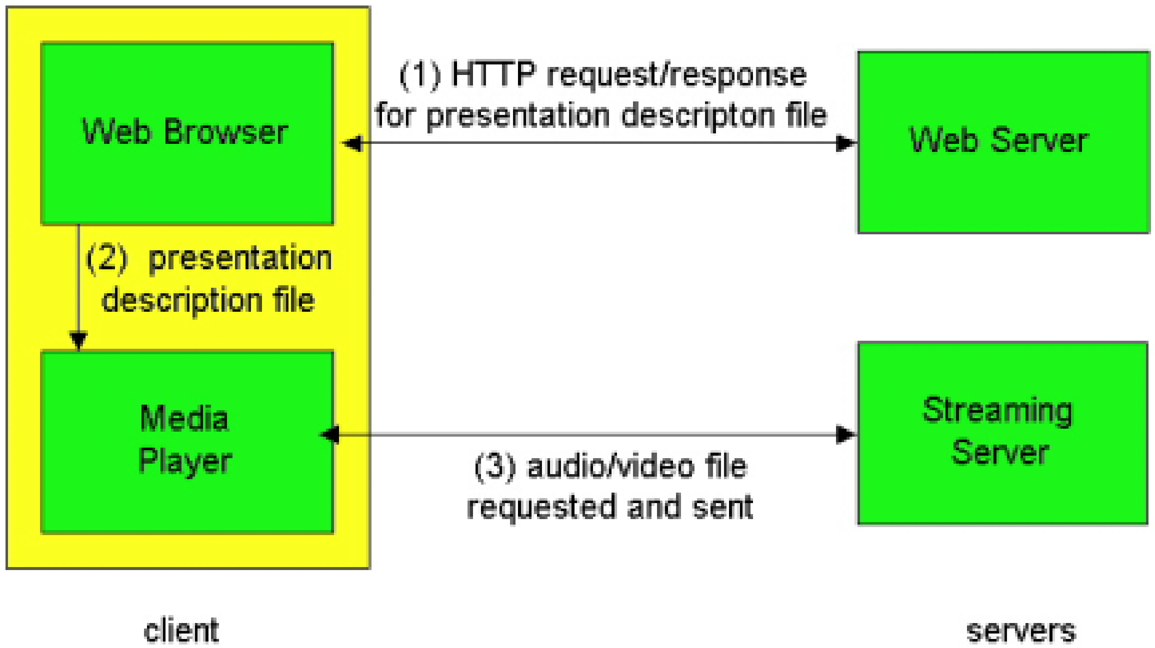
- no, “pipelining,” long delays until playout!

Internet multimedia: streaming approach



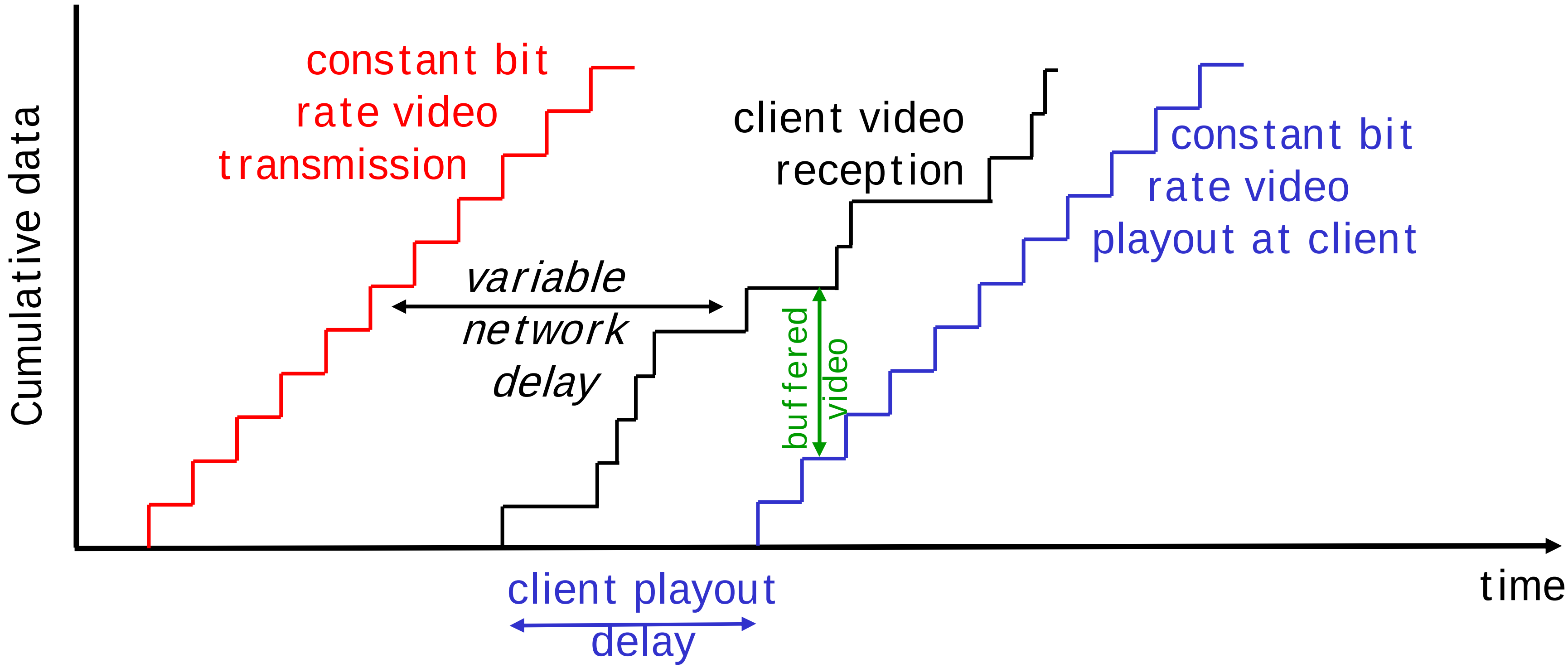
- browser GETs **metafile**
- browser launches player, passing metafile
- player contacts server
- server streams audio/video to player

Streaming from a streaming server



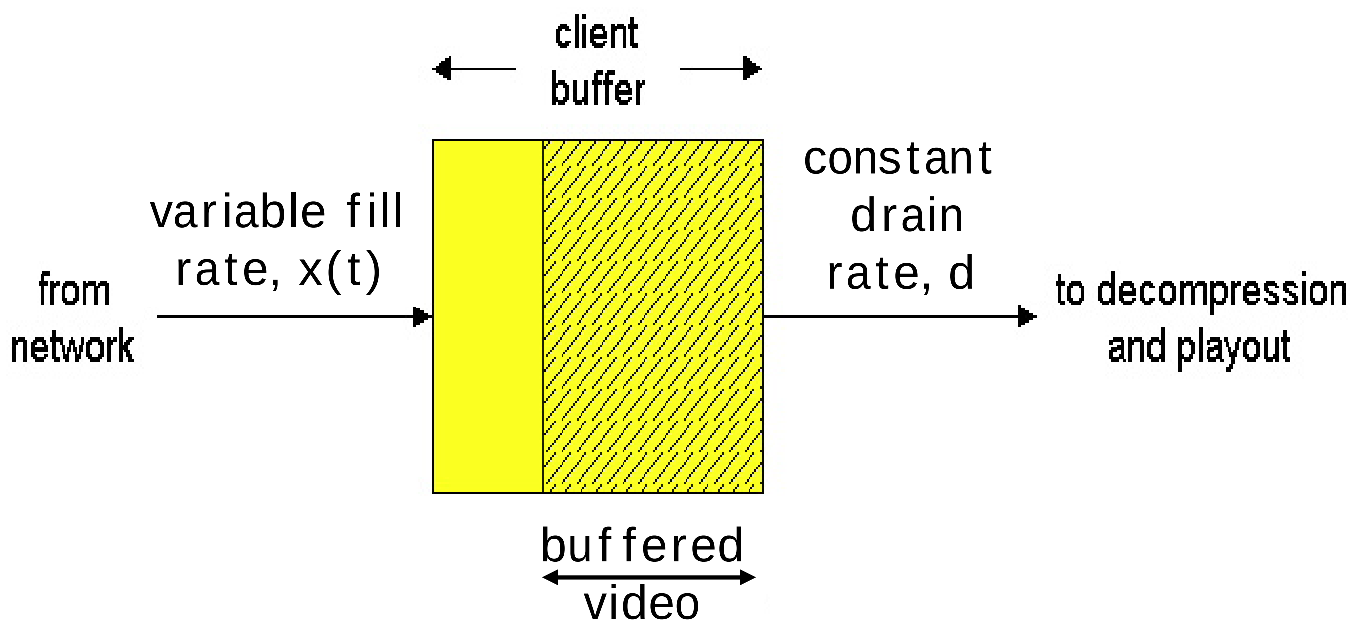
- This architecture allows for non-HTTP protocol between server and media player
- Can also use UDP instead of TCP.

Streaming Multimedia: Client Buffering



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

Streaming Multimedia: Client Buffering



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

Streaming Multimedia: UDP or TCP?

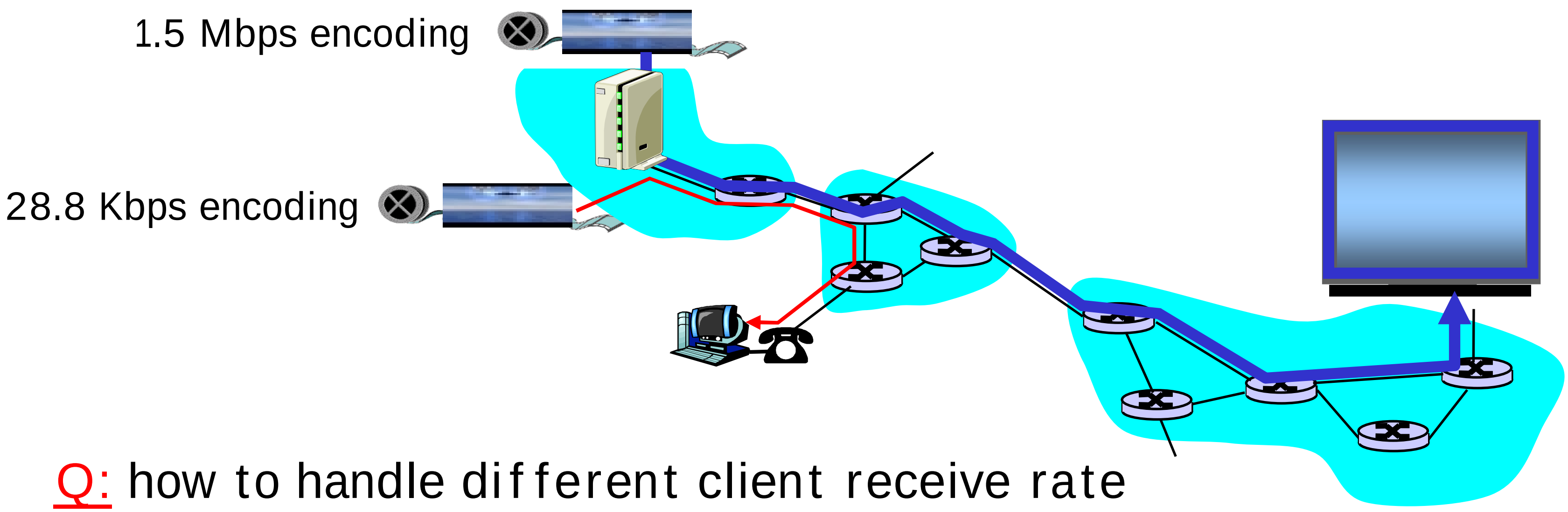
UDP

- server sends at rate appropriate for client (oblivious to network congestion !)
  - often send rate = encoding rate = constant rate
  - then, fill rate = constant rate - packet loss
- short playout delay (2-5 seconds) to compensate for network delay jitter
- error recover: time permitting

TCP

- send at maximum possible rate under TCP
- fill rate fluctuates due to TCP congestion control
- larger playout delay: smooth TCP delivery rate
- HTTP/TCP passes more easily through firewalls

Streaming Multimedia: client rate(s)



Q: how to handle different client receive rate capabilities?

- 28.8 Kbps dialup
- 100Mbps Ethernet

A: (1) server stores, transmits multiple copies of video, encoded at different rates

A: (2) encode video in scalable mode that can be retrieved at different rates! (still at research stage)

User Control of Streaming Media: RTSP

HTTP

- Does not target multimedia content
- No commands for fast forward, etc.

RTSP: RFC 2326

- Client-server application layer protocol.
- For user to control display: rewind, fast forward, pause, resume, repositioning, etc…

What it doesn't do:

- does not define how audio/video is encapsulated for streaming over network
- does not restrict how streamed media is transported; it can be transported over RTP/UDP, UDP or TCP
- does not specify how the media player buffers audio/video

RTSP: out of band control

FTP uses an "out-of-band" control channel:

- A file is transferred over one TCP connection.
- Control information (directory changes, file deletion, file renaming, etc.) is sent over a separate TCP connection.
- The "out-of-band" and "in-band" channels use different port numbers.

RTSP messages are also sent out-of-band:

- RTSP control messages use different port numbers than the media stream: out-of-band.
  - Port 554
- The media stream is considered "in-band".

RTSP Example

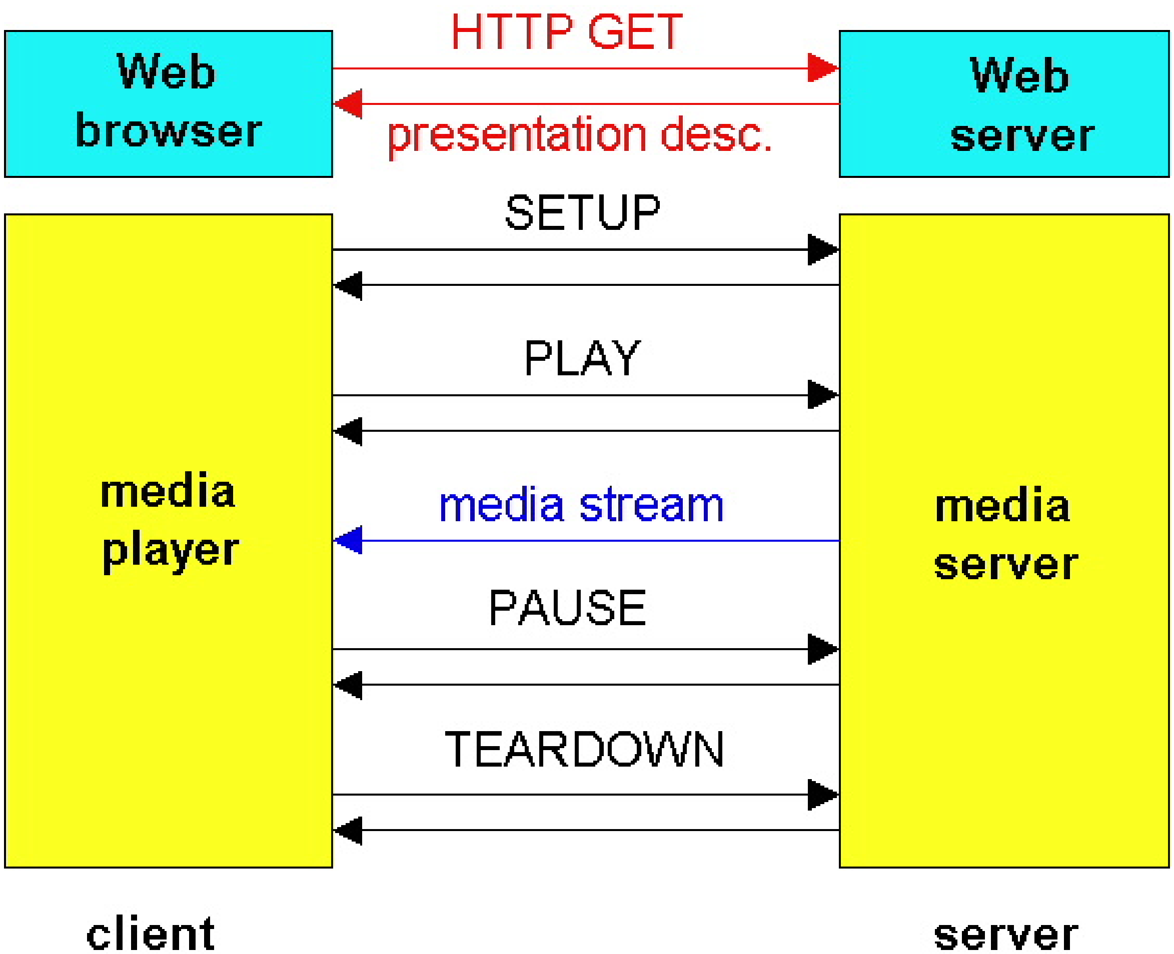
Scenario:

- metafile communicated to web browser
- browser launches player
- player sets up an RTSP control connection, data connection to streaming server

Metafile Example

```
<title>Twister</title>
<session>
       <group language=en lipsync>
               <switch>
                  <track type=audio
                        e="PCMU/8000/1"
                        src = "rtsp://audio.example.com/twister/audio.en/lofi">
                  <track type=audio
                        e="DVI4/16000/2" pt="90 DVI4/8000/1"
                        src="rtsp://audio.example.com/twister/audio.en/hifi">
               </switch>
            <track type="video/jpeg"
                        src="rtsp://video.example.com/twister/video">
        </group>
</session>
```

RTSP Operation



RTSP Exchange Example

```
C: SETUP rtsp://audio.example.com/twister/audio RTSP/1.0
   Transport: rtp/udp; compression; port=3056; mode=PLAY

S: RTSP/1.0 200 1 OK
    Session 4231

C: PLAY rtsp://audio.example.com/twister/audio.en/lofi RTSP/1.0
    Session: 4231
    Range: npt=0-

C: PAUSE rtsp://audio.example.com/twister/audio.en/lofi RTSP/1.0
    Session: 4231
    Range: npt=37

C: TEARDOWN rtsp://audio.example.com/twister/audio.en/lofi RTSP/1.0
    Session: 4231

S: 200 3 OK
```

RTSP/RTP Experiment

- You are provided a server and client program in Java
  - The server encapsulates stored video frames into RTP packets
    - grab video frame, add RTP headers, create UDP segments, send segments to UDP socket
    - include seq numbers and time stamps
  - The server also implements RTSP server functions, in response to client RTSP requests
  - The client extracts RTP headers to obtain payload packets
  - The client also implements RTSP client functions
    - issue play and pause requests
- You test the client and server programs on two separate computers with your partner
- Observe streamed video quality with different background traffic between two of you

Roadmap

- Multimedia Networking Applications
- RTP and RTCP
- Streaming stored audio and video
  - RTSP
- Internet Phone
  - SIP
  - SIP vs. H.323
- Beyond Best Effort

Real-time interactive applications

- PC-2-PC phone
  - instant messaging services are providing this
- PC-2-phone
  - Dialpad
  - Net2phone
- videoconference with Webcams

Going to now look at a PC-2-PC Internet phone example in detail

Interactive Multimedia: Internet Phone

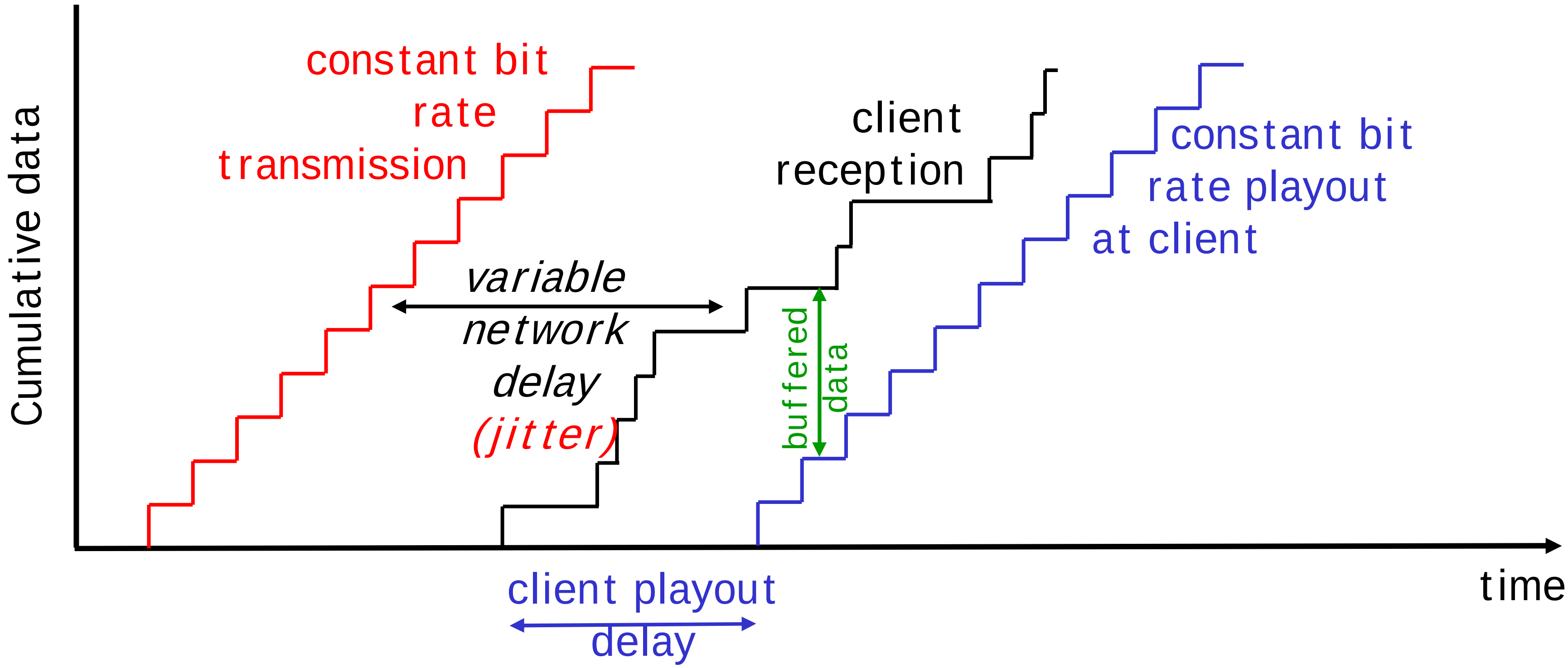
Introduce Internet Phone by way of an example

- speaker's audio: alternating talk spurts, silent periods.
  - 64 kbps during talk spurt
- pkts 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 talkspurt.

Internet Phone: Packet Loss and Delay

- network loss: IP datagram lost due to network congestion (router buffer overflow)
- delay loss: IP datagram arrives too late for playout at receiver
  - delays: processing, queueing in network; end-system (sender, receiver) delays
  - typical maximum tolerable delay: 400 ms
- loss tolerance: depending on voice encoding, losses concealed, packet loss rates between 1% and 10% can be tolerated.

Delay Jitter



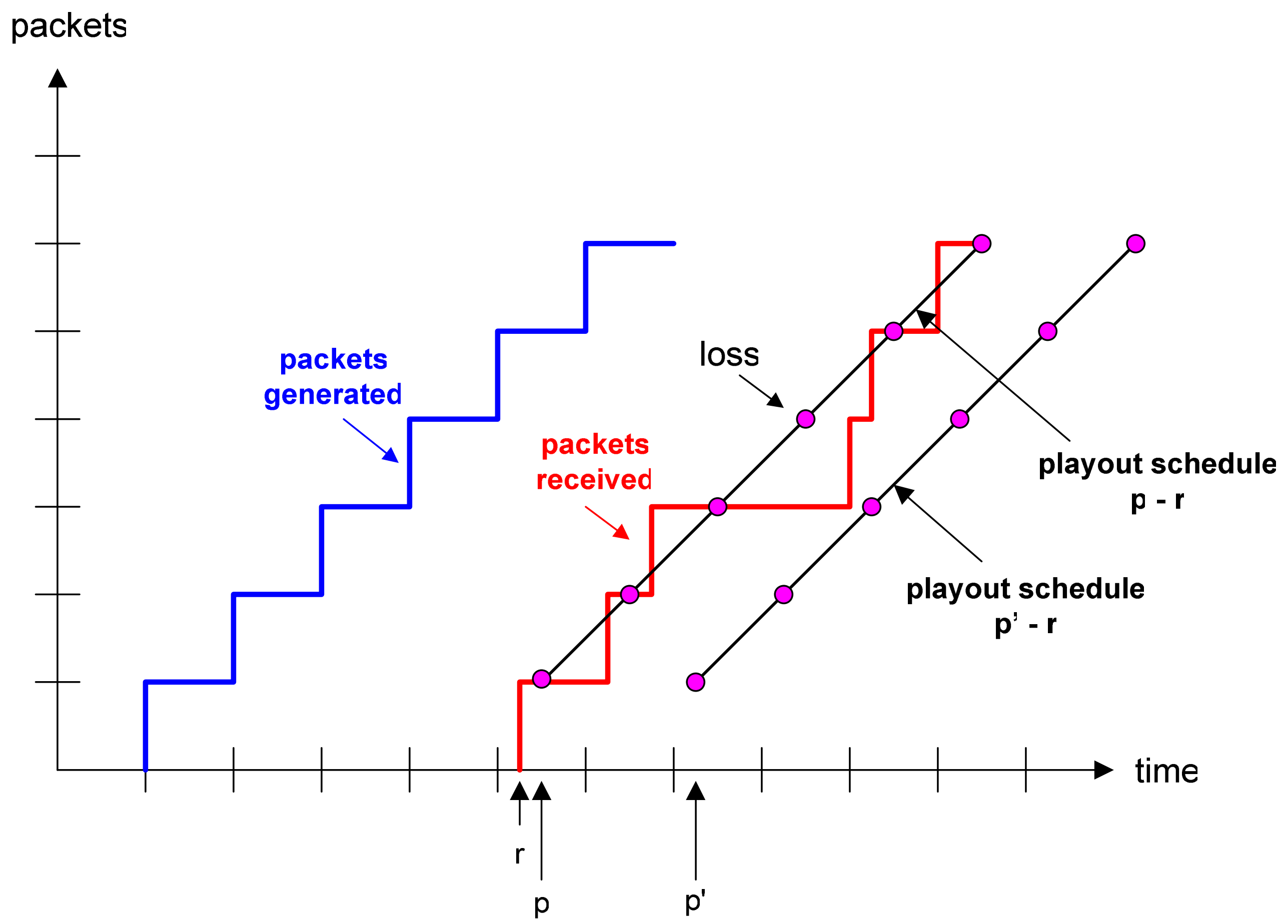
- Consider the end-to-end delays of two consecutive packets: difference can be more or less than 20 msec

Internet Phone: Fixed Playout Delay

- Receiver attempts to playout each chunk exactly q msecs 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, data “lost”
- Tradeoff for q:
  - large q: less packet loss
  - small q: better interactive experience

Fixed Playout Delay

- Sender generates packets every 20 msec during talk spurt.
- First packet received at time r
- First playout schedule: begins at p
- Second playout schedule: begins at p'



SIP

- Session Initiation Protocol
- Comes from IETF

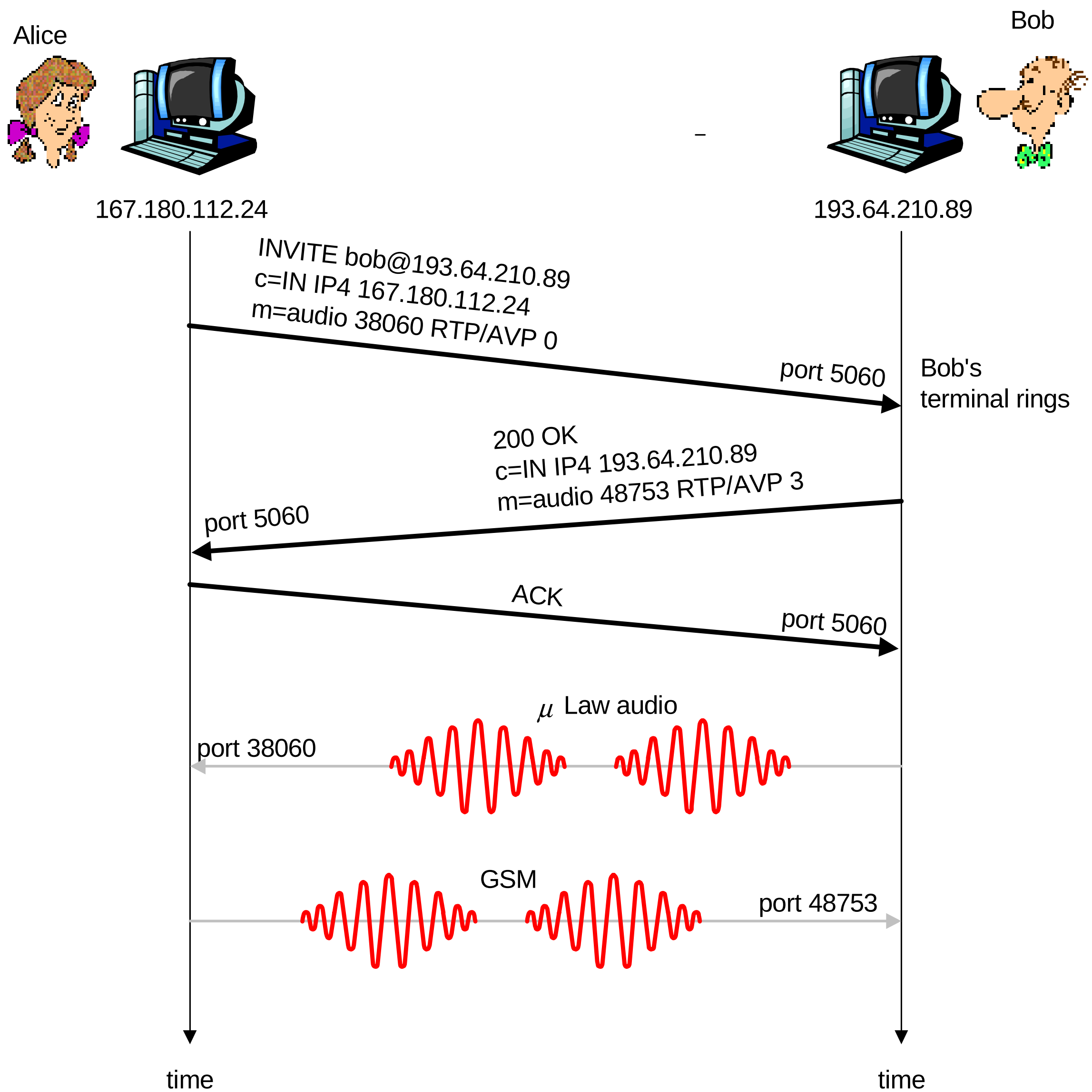
SIP long-term vision

- All telephone calls and video conference calls take place over the Internet
- People are identified by names or e-mail addresses, rather than by phone numbers.
- You can reach the callee, no matter where the callee roams, no matter what IP device the callee is currently using.

SIP Services

- Setting up a call
  - Provides mechanisms for caller to let callee know she wants to establish a call
  - Provides mechanisms so that caller and callee can agree on media type and encoding.
  - Provides mechanisms to end call.
- Determine current IP address of callee.
  - Maps mnemonic identifier to current IP address
- Call management
  - Add new media streams during call
  - Change encoding during call
  - Invite others
  - Transfer and hold calls

Setting up a call to a known IP address



- Alice's SIP invite message indicates her port number & IP address. Indicates encoding that Alice prefers to receive (PCM ulaw)
- Bob's 200 OK message indicates his port number, IP address & preferred encoding (GSM)
- SIP messages can be sent over TCP or UDP; here sent over RTP/UDP.
- Default SIP port number is 5060.

Setting up a call (more)

- Codec negotiation:
  - Suppose Bob doesn't have PCM ulaw encoder.
  - Bob will instead reply with 606 Not Acceptable Reply and list encoders he can use.
  - Alice can then send a new INVITE message, advertising an appropriate encoder.
- Rejecting the call
  - Bob can reject with replies "busy," "gone," "payment required," "forbidden".
- Media can be sent over RTP or some other protocol.

Example of SIP message

```
INVITE sip:bob@domain.com SIP/2.0
Via: SIP/2.0/UDP 167.180.112.24
From: sip:alice@hereway.com
To: sip:bob@domain.com
Call-ID: a2e3a@pigeon.hereway.com
Content-Type: application/sdp
Content-Length: 885

c=IN IP4 167.180.112.24
m=audio 38060 RTP/AVP 0
```

Notes:

- HTTP message syntax
- sdp = session description protocol
- Call-ID is unique for every call.

- Here we don't know Bob's IP address. Intermediate SIP servers will be necessary.
- Alice sends and receives SIP messages using the SIP default port number 5060.
- Alice specifies in Via: header that SIP client sends and receives SIP messages over UDP

Name translation and user locataion

- Caller wants to call callee, but only has callee's name or e-mail address.
- Need to get IP address of callee's current host:
  - user moves around
  - DHCP protocol
  - user has different IP devices (PC, PDA, car device)
- Result can be based on:
  - time of day (work, home)
  - caller (don't want boss to call you at home)
  - status of callee (calls sent to voicemail when callee is already talking to someone)

Service provided by SIP servers:

- SIP registrar server
- SIP proxy server

SIP Registrar

- When Bob starts SIP client, client sends SIP REGISTER message to Bob's registrar server
  (similar function needed by Instant Messaging)

Register Message:

```
REGISTER sip:domain.com SIP/2.0
Via: SIP/2.0/UDP 193.64.210.89
From: sip:bob@domain.com
To: sip:bob@domain.com
Expires: 3600
```

SIP Proxy

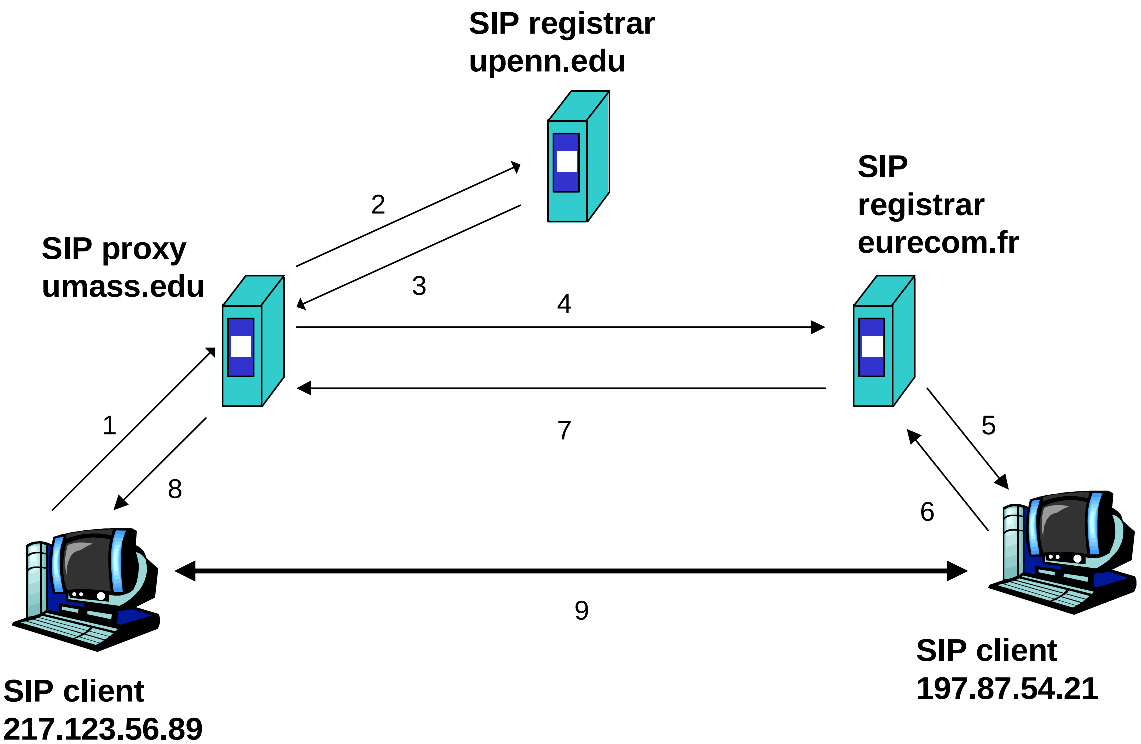
- Alice sends invite message to her proxy server
  - contains address sip:bob@domain.com
- Proxy responsible for routing SIP messages to callee
  - possibly through multiple proxies.
- Callee sends response back through the same set of proxies.
- Proxy returns SIP response message to Alice
  - contains Bob's IP address

- Note: proxy is analogous to local DNS server

Example

Caller jim@umass.edu places a call to keith@upenn.edu

(1) Jim sends INVITE message to umass SIP proxy. (2) Proxy forwards request to upenn registrar server.
(3) Upenn server returns redirect response, indicating that it should try keith@eurecom.fr



(4) umass proxy sends INVITE to eurecom registrar. (5) eurecom registrar forwards INVITE to 197.87.54.21, which is running keith's SIP client. (6-8) SIP response sent back (9) media sent directly between clients.

Note: also a SIP ack message, which is not shown.

ITU-T H.323 Standard

- A protocol suite defining multimedia conferencing over Internet (non-QoS LAN)
  - Control and data channels sent using TCP
  - Audio and video streams sent using RTP/UDP
  - Specifies the operation of gatekeeper (registra), gateway, multipoint control unit (MCU)

system	Audio coding	Video coding	Signaling	Control	Packetization/ synchronization
H.323	G.7xx	H.261/3	Q.931	H.245	H.225.0

Comparison with H.323

- H.323 is a ITU standard for multimedia conferencing. It is a complete, vertically integrated suite of protocols: signaling, registration, admission control, transport (uses RTP/RTCP) and audio /video codecs.
- SIP is a single component, covering signaling, registration, admission control. Works with RTP, but does not mandate it. Works with various types of codecs.
- H.323 comes from the ITU (telephony).
- SIP comes from IETF: Borrows much of its concepts from HTTP. SIP has a Web flavor, whereas H.323 has a telephony flavor.
- SIP uses the KISS principle: Keep it simple stupid.

Roadmap

- Multimedia Networking Applications
- RTP and RTCP
- Streaming stored audio and video
  - RTSP
- Internet Phone
  - SIP
  - SIP vs. H.323
- Recovery from loss – Making the best of best-effort
  - FEC
  - Interleaving
  - Error concealment
- Beyond Best Effort

forward error correction (FEC)

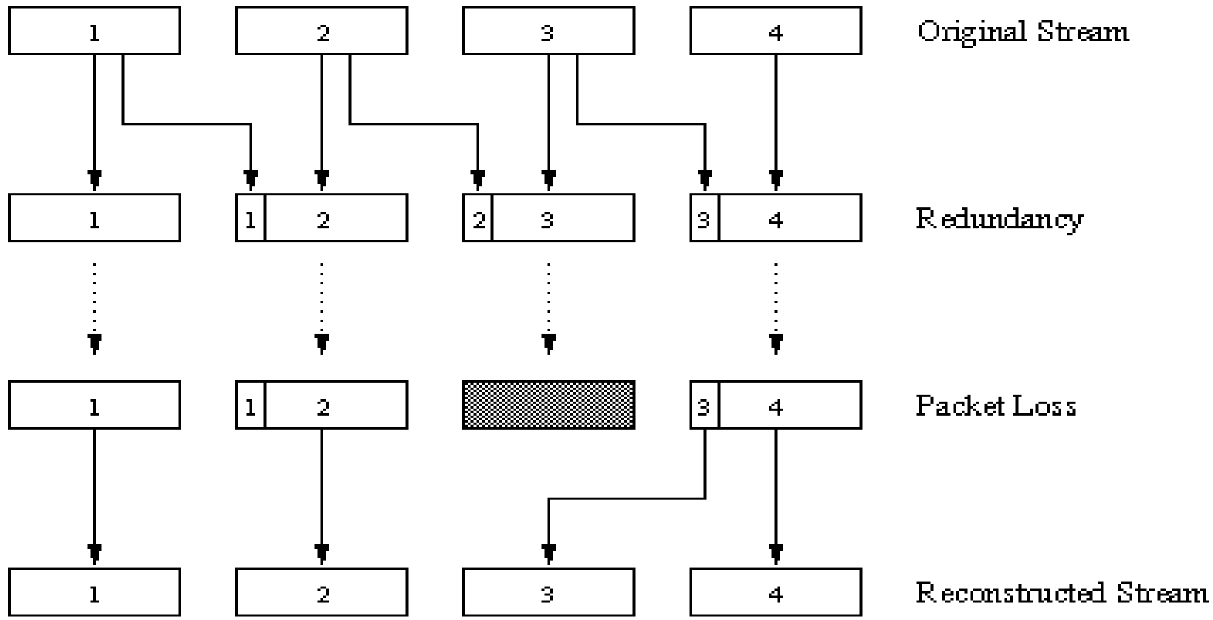
The simplest FEC code:

- 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 by factor 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 fixed to the time to receive all n+1 packets
- Tradeoff:
  - increase n, less bandwidth waste
  - increase n, longer playout delay
  - increase n, higher probability that 2 or more chunks will be lost

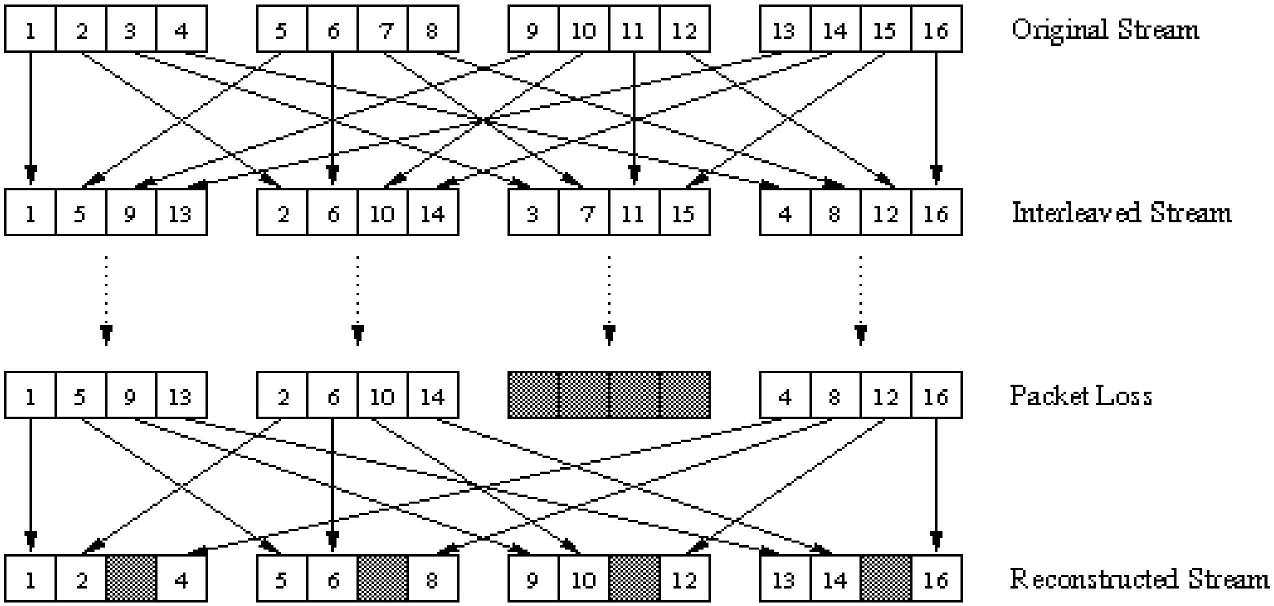
Duplicate Important Part (Unequal FEC)

- "piggyback lower quality stream"
- send lower resolution audio stream as the redundant information
- for example, nominal stream PCM at 64 kbps and redundant stream GSM at 13 kbps.



- Whenever there is non-consecutive loss, the receiver can conceal the loss.
- Can also append (n-1)st and (n-2)nd low-bit rate chunk

Interleaving



Interleaving

- chunks are broken up into smaller units
- for example, 4 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

Internet Multimedia: bag of tricks

- Use UDP to avoid TCP congestion control (delays) for time-sensitive traffic
- Use RTP/UDP to enable QOS monitoring, so that sender and receiver can adjust its operation accordingly
- Client-side adaptive playout delay: to compensate for delay
- Server side matches stream bandwidth to available client-to-server path bandwidth
  - chose among pre-encoded stream rates
  - dynamic server encoding rate
- Error recovery (on top of UDP)
  - FEC, interleaving
  - retransmissions, time permitting
  - Duplicate important parts (unequal error protection)
  - conceal errors: interpolate from nearby data

Roadmap

- Multimedia Networking Applications
- RTP and RTCP
- Streaming stored audio and video
  - RTSP
- Internet Phone
  - SIP
  - SIP vs. H.323
- Recovery from loss
- Beyond Best Effort

Beyond Best Effort

- On-going efforts/proposals
  - Scheduling and policing within the current network structure
  - Next generation Internet
    - RSVP: signaling for resource reservations
    - Differentiated Services (DiffServ): differential guarantees
    - Integrated Services (IntServ): firm guarantees
- For more details, see [Kurose&Ross]

What you should know

- Two types of multimedia applications and requirements
  - One-way streaming vs. interactive
  - How they differ in delay, loss, bandwidth requirement?
- What is RTP/RTCP? How does it enable QoS monitoring? Which layer does it sit on? Does it work with TCP or UDP?
- Streaming pre-encoded video
  - What is network jitter? How to smooth jitter at the receiver for continuous play?
  - What is play-out delay? What are the trade-offs offered by adjusting the play-out delay?
  - What does RTSP do and don't do? What layer does RTSP belong?
- Interactive applications (IP-phone)
  - What is acceptable delay?
  - What are the trade-offs offered by varying the delay?
  - What does SIP do and don't do? What layer does SIP belong?
- What are some of the error control and recovery techniques?
- What are some of the alternatives in next generation Internet?

References

- Jim Kurose, Keith Ross, *Computer Networking: A Top Down Approach Featuring the Internet*,
  2nd edition. Addison-Wesley, 2003. Chap. 6.
  - http://wps.aw.com/aw_kurose_2/
- Henning Schulzrinne's sites for RTP, RTSP, SIP:
  http://www.cs.columbia.edu/~hgs/rtp (rtsp,sip)