***Hamza mustajab***

***Id 15045***

***Advanced Computer Networks***

***DR. NAEEM***

***Multimedia Networking***

**Outline**

**Multimedia networking applications**

**Streaming stored video**

**Voice-over-IP**

**Protocols for real-time conversational applications**

**Network support for multimedia**

***Multimedia: audio***

Analog audio signal sampled at constant rate

• Telephone: 8,000 samples/sec

• CD music: 44,100 samples/sec

Each sample quantized, i.e., rounded

• e.g., 2^8=256 possible quantized values

• Each quantized value represented by bits

e.g., 8 bits for 256 values

Example: 8,000 samples/sec,

256 quantized values: 64,000 bps

Receiver converts bits back to

Analog signal:

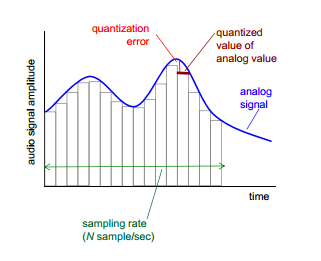
• Some quality reduction

Example rates

CD: 1.411 Mbps

MP3: 96, 128, 160 kbps

Internet telephony: 5.3 kbps and up



***Multimedia: video***

Video: sequence of images displayed at constant rate

• e.g., 24 images/sec

Digital image: array of pixels

• Each pixel represented by bits

Coding: use redundancy with in and between images to decrease # bits used to encode image

• Spatial (within image)

• Temporal (from one image to next)

CBR: (constant bit rate): video encoding rate fixed

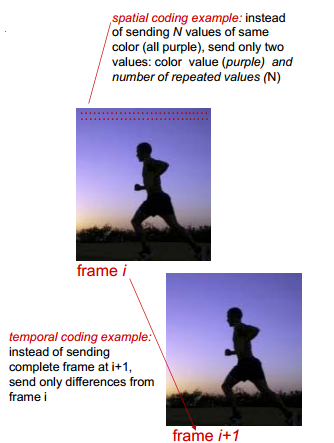
VBR: (variable bit rate): video encoding rate changes as amount of spatial, temporal coding changes

Examples:

• MPEG 1 (CD-ROM) 1.5 Mbps

• MPEG2 (DVD) 3-6 Mbps

• MPEG4 (often used in Internet, < 1 Mbps)



***Multimedia networking: 3 application types***

Streaming, stored audio, video

• Streaming: can begin play out before downloading entire file

• Stored (at server): can transmit faster than audio/video will be rendered (implies storing/buffering at client)

• e.g., YouTube, Netflix, Hulu

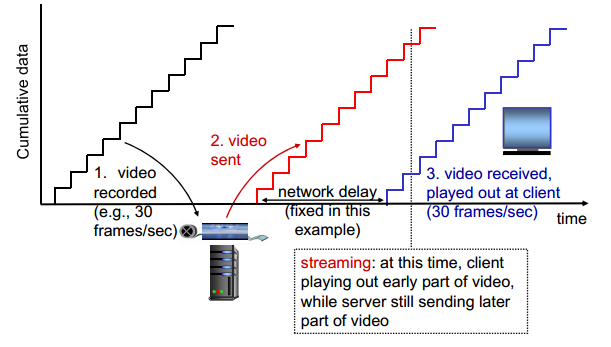
Conversational voice/video over IP

• Interactive nature of human-to-human conversation limits delay tolerance

• e.g., Skype streaming live audio, video

• e.g., live sporting event (futbol)

***Streaming stored video:***



***Streaming stored video: challenges***

Continuous playout constraint: once client playout begins, playback must match original timing

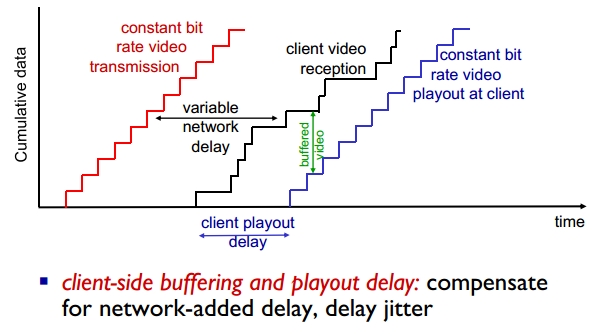
•… but network delays are variable (jitter), so will need client-side buffer to match playout requirements

Other challenges:

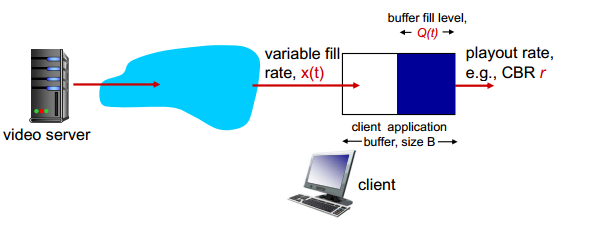
•Client interactivity: pause, fast-forward, rewind, jump through video

•Video packets may be lost, retransmitted

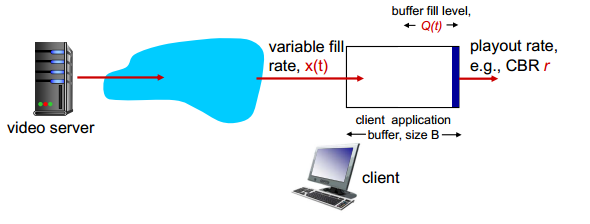
***Streaming stored video: revisited***



***Client-side buffering, playout***



***Client-side buffering, playout***



1. Initial fill of buffer until playout begins at tp

2. Playout begins at tp,

3. Buffer fill level varies over time as fill ratex (t) varies and playout rate r is constant

***Client-side buffering, playout***



Playout buffering: average fill rate (x), playout rate (r):

x < r: buffer eventually empties (causing freezing of video

Playout until buffer again fills)

x > r: buffer will not empty, provided initial playout delay is large enough to absorb variability in x(t)

•Initial playout delay tradeoff: buffer starvation less likely with larger delay, but larger delay until user begins watching

***Streaming multimedia: UDP***

Server sends at rate appropriate for client

• Often: send rate = encoding rate = constant rate

• Transmission rate can be oblivious to congestion levels

Short playout delay (2-5 seconds) to remove network jitter

Error recovery: application-level, time permitting

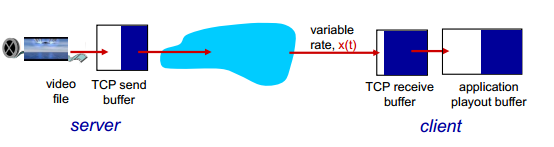
RTP [RFC 2326]: multimedia payload types

UDP may notgo through firewalls

***Streaming multimedia: HTTP***

Multimedia file retrieved via HTTP GET

Send at maximum possible rate under TCP



Fill rate fluctuates due to TCP congestion control, retransmissions (in-order delivery)

Larger playout delay: smooth TCP delivery rate

HTTP/TCP passes more easily through firewalls

***Voice-over-IP (VoIP)***

VoIP end-end-delay requirement: needed to maintain “conversational” aspect

• Higher delays noticeable, impair interactivity

• < 150 msec: good

• > 400 msec bad

•Includes application-level (packetization , playout), network delays

Session initialization: how does callee advertise IP address, port number, encoding algorithms?

Value-added services: call forwarding, screening, recording

Emergency services: 911

***VoIP characteristics***

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 of data

Application-layer header added to each chunk

Chunk+header encapsulated into UDP or TCP segment

Application sends segment into socket every 20 msec during talkspurt

***VoIP: packet loss, delay***

Network loss: IP datagram lost due to network congestion (router buffer overflow)

Delay loss: IP datagram arrives too late for playout at receiver

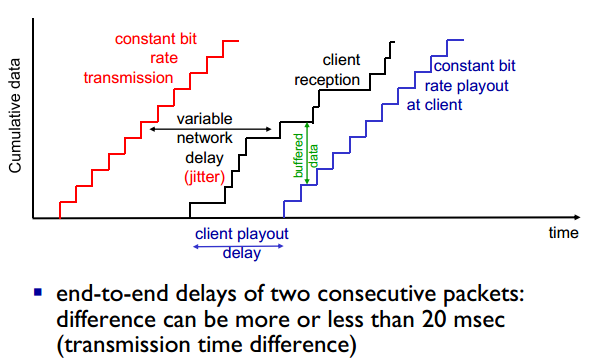
• Delays: processing, queueing in network; end-system (sender, receiver) delays

• Typical maximum tolerable delay: 400 ms

Loss tolerance: depending on voice encoding, loss concealment, packet loss rates between 1% and

10% can be tolerated.

***Delay jitter***



***VoIP: 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 in choosing q:

• Large q: less packet loss

• Small q:better interactive experience

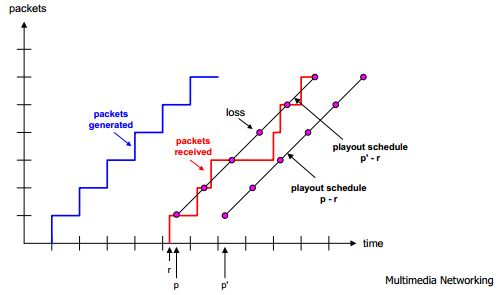
***VoIP: fixed playout delay***

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’



***Adaptive playout delay (1)***

Goal: low playout delay, low late loss rate

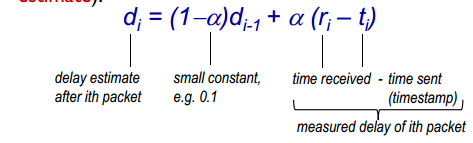
Approach: adaptive playout delay adjustment:

• Estimate network delay, adjust playout delay at beginning of each talk spurt

• Silent periods compressed and elongated

• Chunks still played out every 20 msec during talk spurt

Adaptively estimate packet delay: (EWMA -exponentially weighted moving average, recall TCP RTT estimate):



***Adaptive playout delay (2)***

Also useful to estimate average deviation of delay vi



Estimates di ,vi calculated for every received packet, but used only at start of talk spurt

For first packet in talk spurt, playout time is:



Remaining packets in talkspurt are played out periodically

***Adaptive playout delay (3)***

Q:How does receiver determine whether packet is first in a talkspurt?

if no loss, receiver looks at successive timestamps

• Difference of successive stamps > 20 msec -->talk spurt begins.

with loss possible, receiver must look at both time stamps and sequence numbers

• Difference of successive stamps > 20 msec andsequence numbers without gaps --> talk spurt begins

***Real-Time Protocol (RTP)***

RTP specifies packet structure for packets carrying audio, video data

RFC 3550

RTP packet provides

• Payload type identification

• Packet sequence numbering

• Time stamping

RTP runs in end systems

RTP packets encapsulated in UDP segments

Interoperability: if two VoIP applications run RTP, they may be able to work together

***RTP runs on top of UDP***

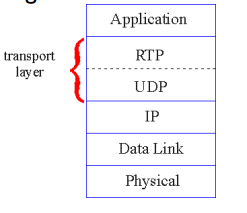
RTP libraries provide transport-layer interface that extends UDP:

• Port numbers, IP addresses

• Payload type identification

• Packet sequence numbering

• TIme-stamping



***RTP example***

Example: sending 64 kbps PCM-encoded voice over RTP

Application collects encoded data in chunks,

e.g., every 20 msec = 160 bytes in a chunk

Audio chunk + RTP header form RTP packet, which is encapsulated in UDP segment

RTP header indicates type of audio encoding in each packet

• sender can change encoding during conference

RTP header also contains sequence numbers, timestamps

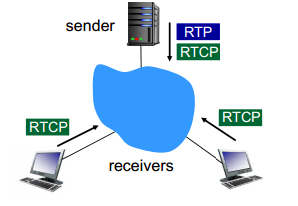
***RTP and QoS***

RTP does not provide any mechanism to ensure timely data delivery or other QoS guarantees

RTP encapsulation only seen at end systems (not by intermediate routers)

• Routers provide best-effort service, making no special effort to ensure that RTP packets arrive at destination in timely matter

***RTCP: multiple multicast senders***

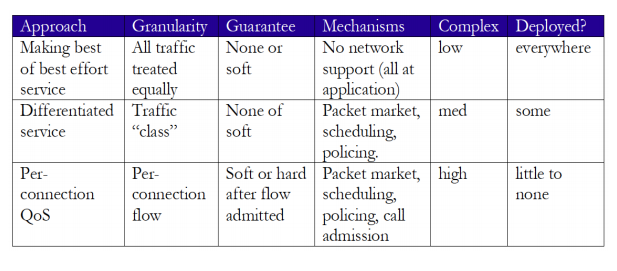


Each RTP session: typically a single multicast address; all RTP /RTCP packets belonging to session use multicast address

RTP, RTCP packets distinguished from each other via distinct port numbers

To limit traffic, each participant reduces RTCP traffic as number of conference participants increases

***Network support for multimedia***



***Dimensioning best effort networks***

Approach: deploy enough link capacity so that congestion doesn’t occur, multimedia traffic flows without delay or loss

• Low complexity of network mechanisms (use current “best effort” network)

• High bandwidth costs

Challenges:

• Network dimensioning: how much bandwidth is “enough?”

• Estimating network traffic demand: needed to determine how much bandwidth is “enough” (for that much traffic)

***Providing multiple classes of service***

Thus far: making the best of best effort service

• one-size fits all service model

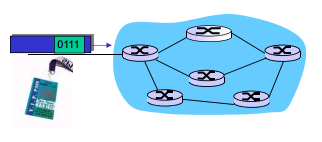
Alternative: multiple classes of service

• Partition traffic into classes

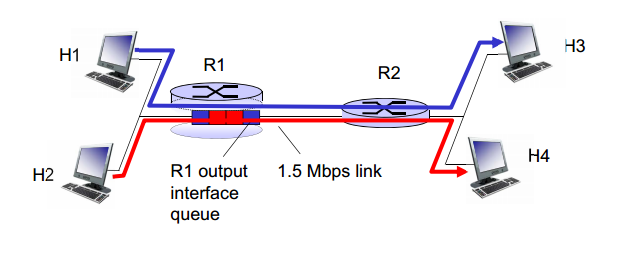
• Network treats different classes of traffic differently (analogy: VIP service versus regular service)

granularity: differential service among multiple classes, not among individual connections

History: ToS bits



***Multiple classes of service: scenario***

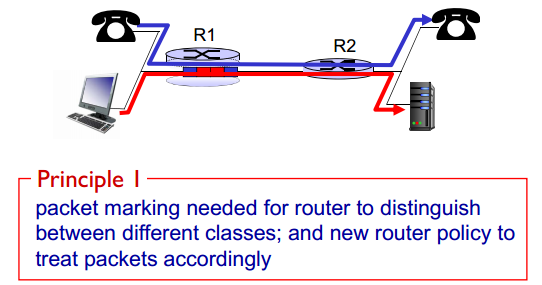


***Scenario 1: mixed HTTP and VoIP***

example: 1Mbps VoIP, HTTP share 1.5 Mbps link.

• HTTP bursts can congest router, cause audio loss

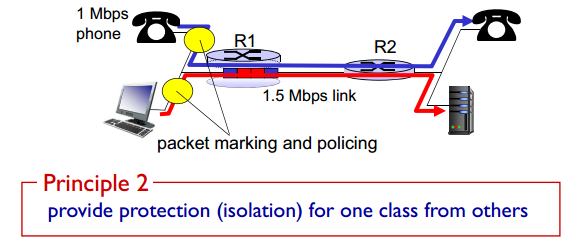
• want to give priority to audio over HTTP



***Principles for QOS guarantees (more)***

What if applications misbehave (VoIP sends higher than declared rate)

• Policing: force source adherence to bandwidth allocations marking, policingat network edge



***Principles for QOS guarantees (more)***

Allocating fixed (non-sharable) bandwidth to flow: inefficient use of bandwidth if flows doesn’t use its

allocation

