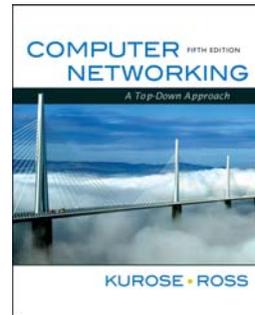


Chapter 7 Multimedia Networking



A note on the use of these ppt slides:

We're making these slides freely available to all (faculty, students, readers). They're in PowerPoint form so you can add, modify, and delete slides (including this one) and slide content to suit your needs. They obviously represent a lot of work on our part. In return for use, we only ask the following:

- If you use these slides (e.g., in a class) in substantially unaltered form, that you mention their source (after all, we'd like people to use our book!)
- If you post any slides in substantially unaltered form on a www site, that you note that they are adapted from (or perhaps identical to) our slides, and note our copyright of this material.

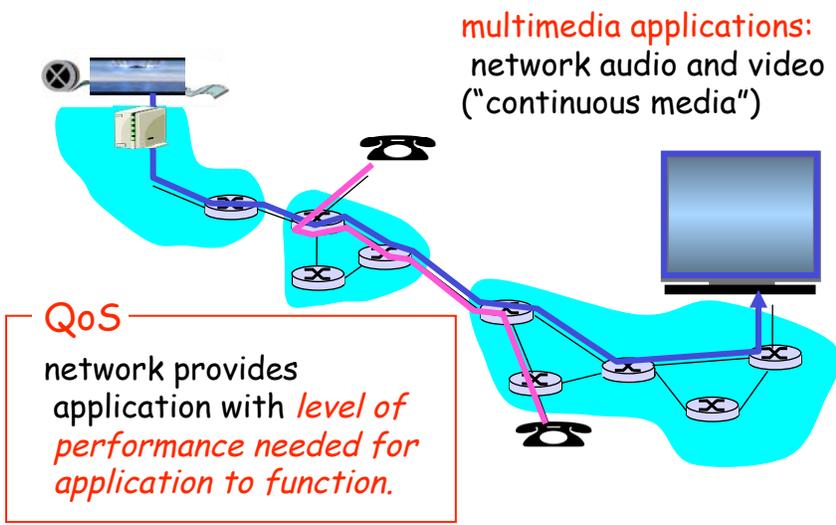
Computer Networking: A Top Down Approach
5th edition.
Jim Kurose, Keith Ross
Addison-Wesley, April 2009.

Thanks and enjoy! JFK / KWR

All material copyright 1996-2009
J.F Kurose and K.W. Ross, All Rights Reserved

7: Multimedia Networking 7-1

Multimedia and Quality of Service: What is it?



7: Multimedia Networking 7-2

Chapter 7: goals

Principles

- ❑ classify multimedia applications
- ❑ identify network services applications need
- ❑ making the best of best effort service

Protocols and Architectures

- ❑ specific protocols for best-effort
- ❑ mechanisms for providing QoS
- ❑ architectures for QoS

7: Multimedia Networking 7-3

Chapter 7 outline

7.1 multimedia networking applications

7.2 streaming stored audio and video

7.3 making the best out of best effort service

7.4 protocols for real-time interactive applications
RTP, RTCP, SIP

7.5 providing multiple classes of service

7.6 providing QoS guarantees

7: Multimedia Networking 7-4

MM Networking Applications

Classes of MM applications:

- 1) stored streaming
- 2) live streaming
- 3) interactive, real-time

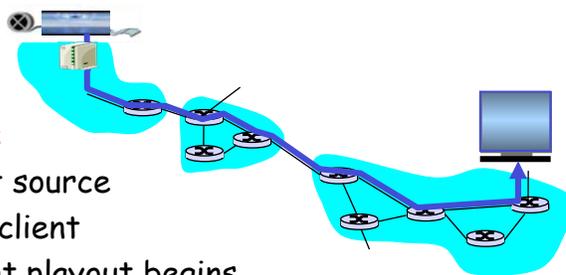
Jitter is the variability of packet delays within the same packet stream

Fundamental characteristics:

- ❑ typically **delay sensitive**
 - end-to-end delay
 - delay jitter
- ❑ **loss tolerant**: infrequent losses cause minor glitches
- ❑ antithesis of data, which are loss *intolerant* but delay *tolerant*.

7: Multimedia Networking 7-5

Streaming Stored Multimedia

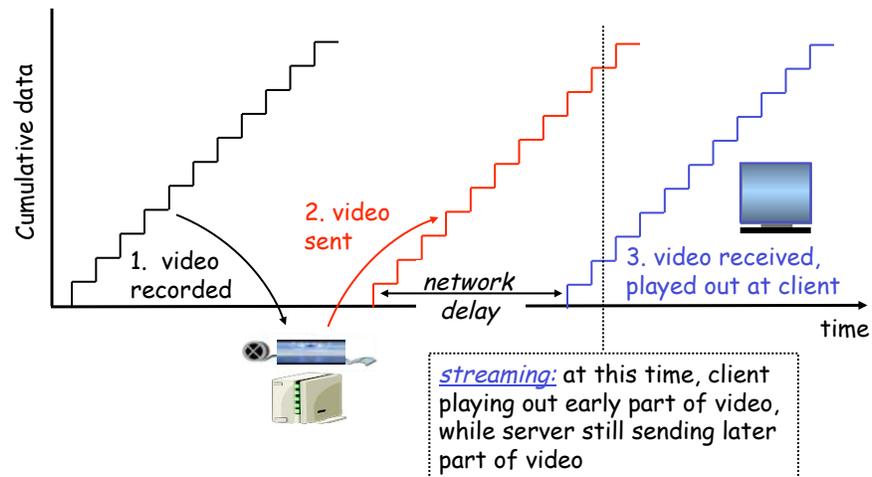


Stored streaming:

- ❑ 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

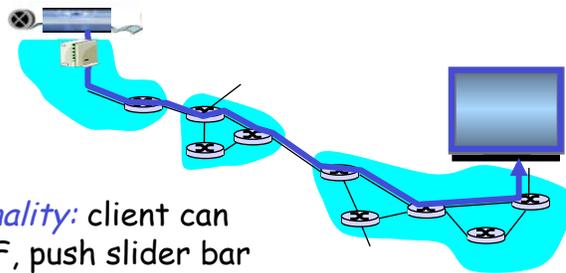
7: Multimedia Networking 7-6

Streaming Stored Multimedia: What is it?



7: Multimedia Networking 7-7

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
- timing constraint for still-to-be transmitted data: in time for payout

7: Multimedia Networking 7-8

Streaming *Live* Multimedia

Examples:

- ❑ Internet radio talk show
- ❑ live sporting event

Streaming (as with streaming *stored* multimedia)

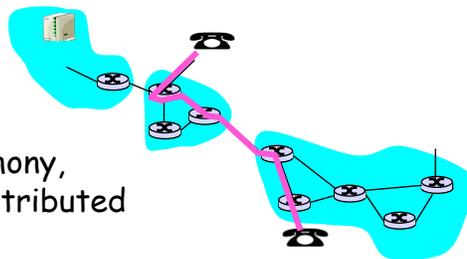
- ❑ playback buffer
- ❑ playback can lag tens of seconds after transmission
- ❑ still have timing constraint

Interactivity

- ❑ fast forward impossible
- ❑ rewind, pause possible!

7: Multimedia Networking 7-9

Real-Time Interactive 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?

7: Multimedia Networking 7-10

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

7: Multimedia Networking 7-11

How should the Internet evolve to better support multimedia?

Integrated services philosophy:

- ❑ fundamental changes in Internet so that apps can reserve end-to-end bandwidth
- ❑ requires new, complex software in hosts & routers

Laissez-faire

- ❑ no major changes
- ❑ more bandwidth when needed
- ❑ content distribution, application-layer multicast
 - application layer

Differentiated services philosophy:

- ❑ fewer changes to Internet infrastructure, yet provide 1st and 2nd class service



What's your opinion?

7: Multimedia Networking 7-12

A few words about audio compression

- analog 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
 - 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

7: Multimedia Networking 7-13

A few words about video compression

- video: sequence of images displayed at constant rate
 - e.g. 24 images/sec
- digital image: array of pixels
 - each pixel represented by bits
- redundancy
 - spatial (within image)
 - temporal (from one image to next)

Examples:

- MPEG 1 (CD-ROM) 1.5 Mbps
- MPEG2 (DVD) 3-6 Mbps
- MPEG4 (often used in Internet, < 1 Mbps)

Research:

- layered (scalable) video
 - adapt layers to available bandwidth

7: Multimedia Networking 7-14

Streaming Stored Multimedia

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

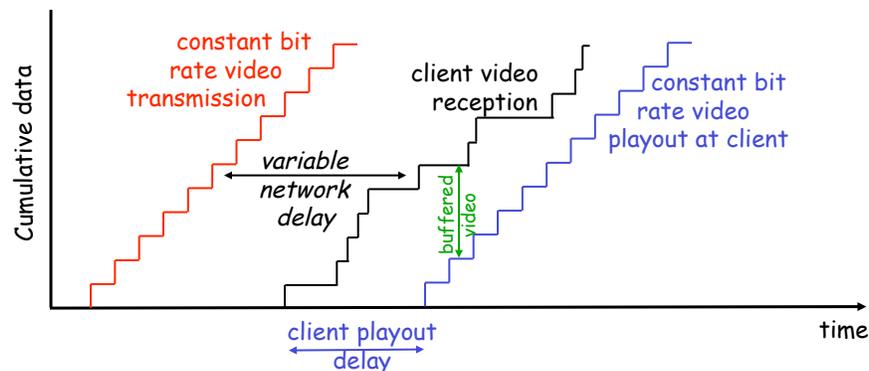
- client-side buffering
- use of UDP versus TCP
- multiple encodings of multimedia

Media Player

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

7: Multimedia Networking 7-15

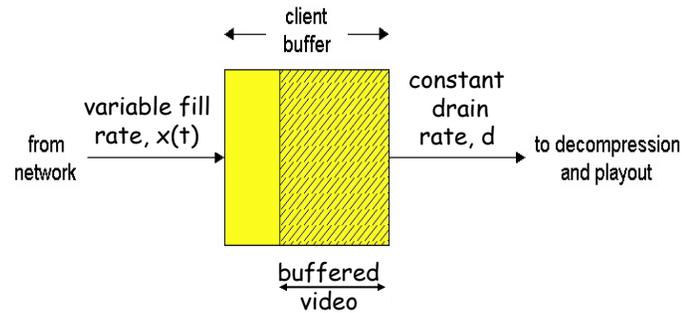
Streaming Multimedia: Client Buffering



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

7: Multimedia Networking 7-16

Streaming Multimedia: Client Buffering



- ❑ client-side buffering, playout delay compensate for network-added delay, delay jitter

7: Multimedia Networking 7-17

Streaming Multimedia: UDP or TCP?

UDP

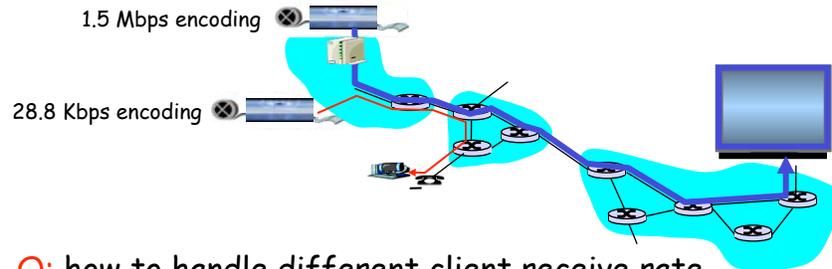
- ❑ server sends at rate appropriate for client (oblivious to network congestion !)
 - often send rate = encoding rate = constant rate
 - then, fill rate = constant rate - packet loss
- ❑ short playout delay (2-5 seconds) to remove network jitter
- ❑ error recover: time permitting

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

7: Multimedia Networking 7-18

Streaming Multimedia: client rate(s)



Q: how to handle different client receive rate capabilities?

- 28.8 Kbps dialup
- 100 Mbps Ethernet

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

7: Multimedia Networking 7-19

Real-time interactive applications

- PC-2-PC phone
 - Skype
- PC-2-phone
 - Dialpad
 - Net2phone
 - Skype
- videoconference with webcams
 - Skype
 - Polycom

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

7: Multimedia Networking 7-20

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

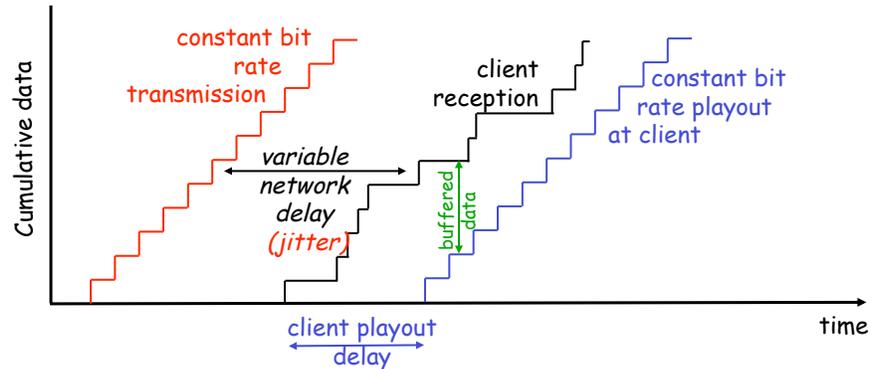
7: Multimedia Networking 7-21

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.

7: Multimedia Networking 7-22

Delay Jitter



- consider end-to-end delays of two consecutive packets: difference can be more or less than 20 msec (transmission time difference)

7: Multimedia Networking 7-23

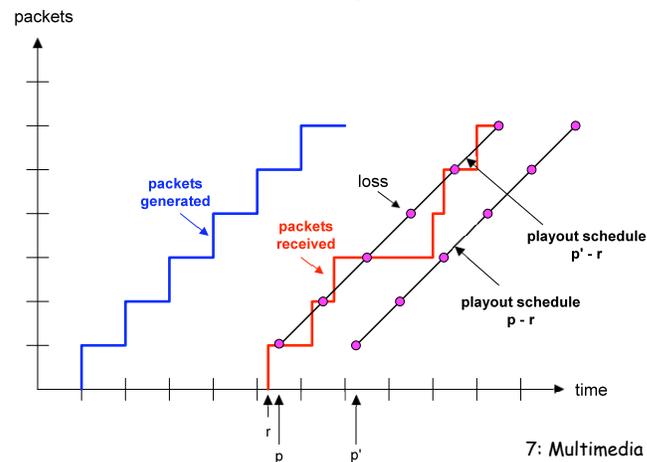
Internet Phone: Fixed Playout Delay

- 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 in choosing q :
 - *large q* : less packet loss
 - *small q* : better interactive experience

7: Multimedia Networking 7-24

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:** minimize playout delay, keeping late loss rate low
- **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.

t_i = timestamp of the i th packet

r_i = the time packet i is received by receiver

p_i = the time packet i is played at receiver

$r_i - t_i$ = network delay for i th packet

d_i = estimate of average network delay after receiving i th packet

dynamic estimate of average delay at receiver:

$$d_i = (1 - u)d_{i-1} + u(r_i - t_i)$$

where u is a fixed constant (e.g., $u = .01$).

7: Multimedia Networking 7-26

Adaptive playout delay (2)

- also useful to estimate average deviation of delay, v_i :

$$v_i = (1 - u)v_{i-1} + u |r_i - t_i - d_i|$$

- estimates d_i , v_i calculated for every received packet (but used only at start of talk spurt)

- for first packet in talk spurt, playout time is:

$$p_i = t_i + d_i + K v_i$$

where K is positive constant

- remaining packets in talkspurt are played out periodically

7: Multimedia Networking 7-27

Adaptive Playout (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 **and** sequence numbers without gaps --> talk spurt begins.

7: Multimedia Networking 7-28

Recovery from packet loss (1)

Forward Error Correction (FEC): simple scheme

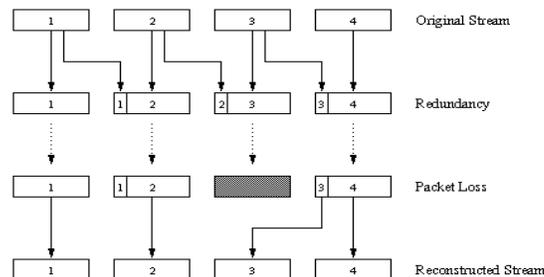
- for every group of n chunks create redundant chunk by exclusive OR-ing n original chunks
- send out $n+1$ chunks, increasing bandwidth by factor $1/n$.
- can reconstruct original n chunks if at most one lost chunk from $n+1$ chunks
- playout delay: enough time to receive all $n+1$ packets
- tradeoff:
 - increase n , less bandwidth waste
 - increase n , longer playout delay
 - increase n , higher probability that 2 or more chunks will be lost

7: Multimedia Networking 7-29

Recovery from packet loss (2)

2nd FEC scheme

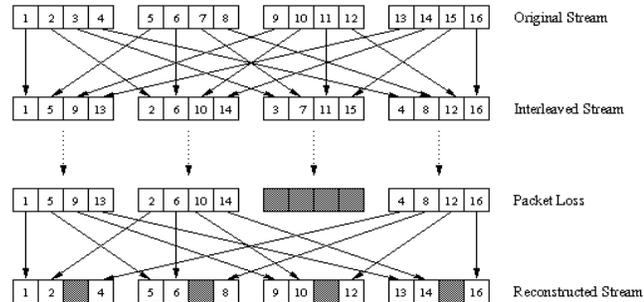
- "piggyback lower quality stream"
- send lower resolution audio stream as redundant information
- e.g., nominal stream PCM at 64 kbps and redundant stream GSM at 13 kbps.



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

7: Multimedia Networking 7-30

Recovery from packet loss (3)



Interleaving

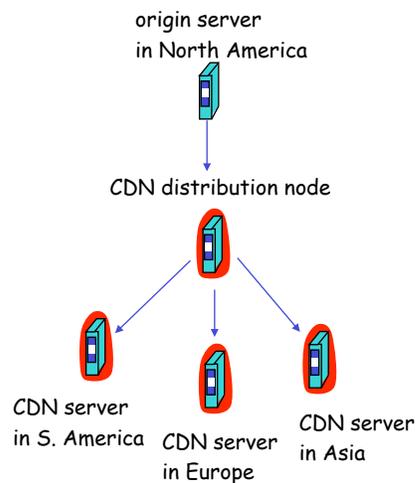
- ❑ chunks divided into smaller units
- ❑ for example, four 5 msec units per chunk
- ❑ packet contains small units from different chunks
- ❑ if packet lost, still have most of every chunk
- ❑ no redundancy overhead, but increases playout delay

7: Multimedia Networking 7-31

Content distribution networks (CDNs)

Content replication

- ❑ challenging to stream large files (e.g., video) from single origin server in real time
- ❑ *solution:* replicate content at hundreds of servers throughout Internet
 - content downloaded to CDN servers ahead of time
 - *placing content "close" to user avoids impairments (loss, delay) of sending content over long paths*
 - CDN server typically in edge/access network

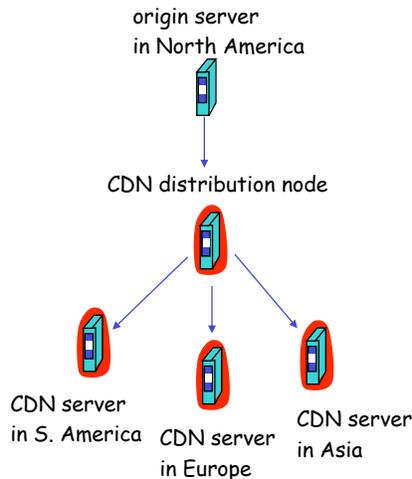


7: Multimedia Networking 7-32

Content distribution networks (CDNs)

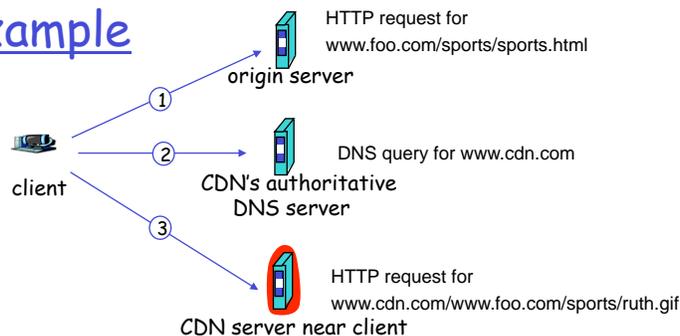
Content replication

- ❑ CDN (e.g., Akamai) customer is the content provider (e.g., CNN)
- ❑ CDN replicates customers' content in CDN servers.
- ❑ when provider updates content, CDN updates servers



7: Multimedia Networking 7-33

CDN example



origin server (www.foo.com)

- ❑ distributes HTML
- ❑ replaces:
<http://www.foo.com/sports.ruth.gif>
 with <http://www.cdn.com/www.foo.com/sports/ruth.gif>

CDN company (cdn.com)

- ❑ distributes gif files
- ❑ uses its authoritative DNS server to route redirect requests

7: Multimedia Networking 7-34

Summary: Internet Multimedia: bag of tricks

- ❑ use **UDP** to avoid TCP congestion control (delays) for time-sensitive traffic
- ❑ 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, error concealment
 - retransmissions, time permitting
- ❑ CDN: bring content closer to clients

7: Multimedia Networking 7-35

Real-Time Protocol (RTP)

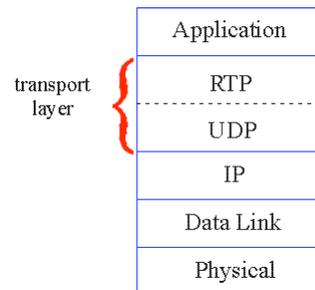
- ❑ RTP specifies packet structure for packets carrying audio, video data
- ❑ RFC 3550
- ❑ RTP packet provides
 - payload type identification
 - packet sequence numbering
 - time stamping
- ❑ RTP runs in end systems
- ❑ RTP packets encapsulated in UDP segments
- ❑ interoperability: if two Internet phone applications run RTP, then they may be able to work together

7: Multimedia Networking 7-36

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



7: Multimedia Networking 7-37

RTP Example

- consider 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.

7: Multimedia Networking 7-38

RTP and QoS

- ❑ RTP does **not** provide any mechanism to ensure timely data delivery or other QoS guarantees.
- ❑ RTP encapsulation is only seen at end systems (not) by intermediate routers.
 - routers providing best-effort service, making no special effort to ensure that RTP packets arrive at destination in timely matter.

7: Multimedia Networking 7-39

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 receiver via 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.

7: Multimedia Networking 7-40

RTP Header (2)

- ❑ Timestamp field (32 bytes long): sampling instant of first byte in this RTP data packet
 - for audio, timestamp clock typically increments by one for each sampling period (for example, each 125 usecs for 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 source of + RTP stream. Each stream in RTP session should have distinct SSRC.

7: Multimedia Networking 7-41

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: # packets sent, # packets lost, interarrival jitter, etc.
- ❑ feedback can be used to control performance
 - sender may modify its transmissions based on feedback

7: Multimedia Networking 7-42

RTCP Packets

Receiver report packets:

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

Sender report packets:

- ❑ SSRC of RTP stream, current time, number of packets sent, 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

7: Multimedia Networking 7-43

SIP: Session Initiation Protocol [RFC 3261]

SIP long-term vision:

- ❑ all telephone calls, video conference calls take place over Internet
- ❑ people are identified by names or e-mail addresses, rather than by phone numbers
- ❑ you can reach callee, no matter where callee roams, no matter what IP device callee is currently using

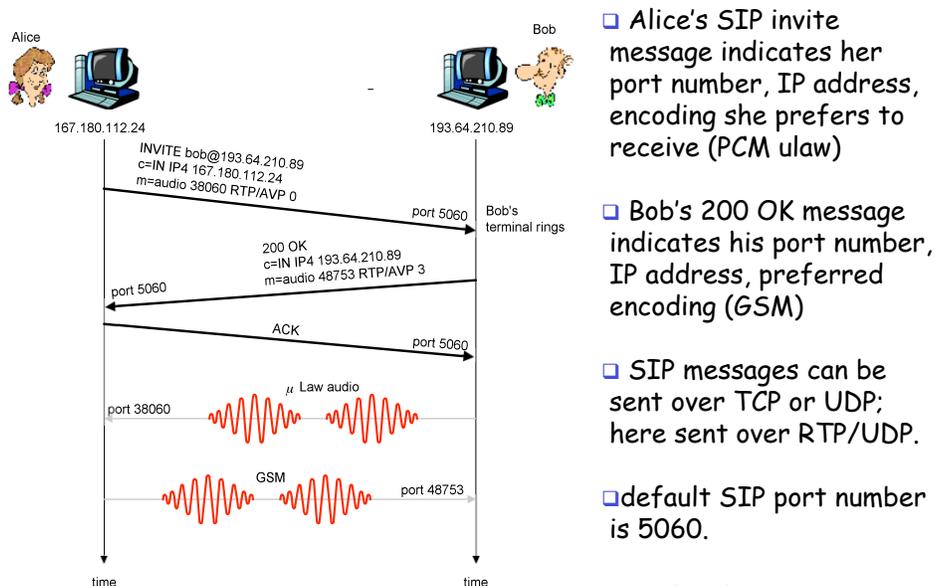
7: Multimedia Networking 7-44

SIP Services

- Setting up a call, SIP provides mechanisms ..
 - for caller to let callee know she wants to establish a call
 - so caller, callee can agree on media type, encoding
 - 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, hold calls

7: Multimedia Networking 7-45

Setting up a call to known IP address



- Alice's SIP invite message indicates her port number, IP address, encoding she 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.

7: Multimedia Networking 7-46

Setting up a call (more)

- ❑ codec negotiation:
 - suppose Bob doesn't have PCM ulaw encoder.
 - Bob will instead reply with 606 Not Acceptable Reply, listing his encoders Alice can then send new INVITE message, advertising different encoder
- ❑ rejecting a call
 - Bob can reject with replies "busy," "gone," "payment required," "forbidden"
- ❑ media can be sent over RTP or some other protocol

7: Multimedia Networking 7-47

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 needed.

❑ Alice sends, receives SIP messages using SIP default port 506

❑ Alice specifies in Via: header that SIP client sends, receives SIP messages over UDP

7: Multimedia Networking 7-48

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

7: Multimedia Networking 7-49

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
```

7: Multimedia Networking 7-50

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
- ❑ proxy analogous to local DNS server

7: Multimedia Networking 7-51

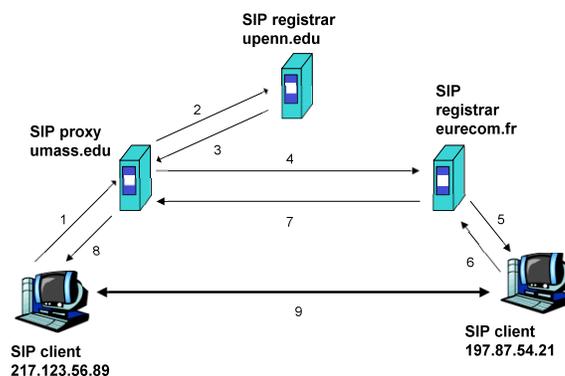
Example

Caller jim@umass.edu
with 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.

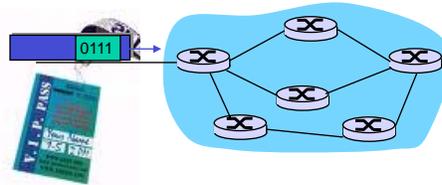
Note: also a SIP ack message, which is not shown.



7: Multimedia Networking 7-52

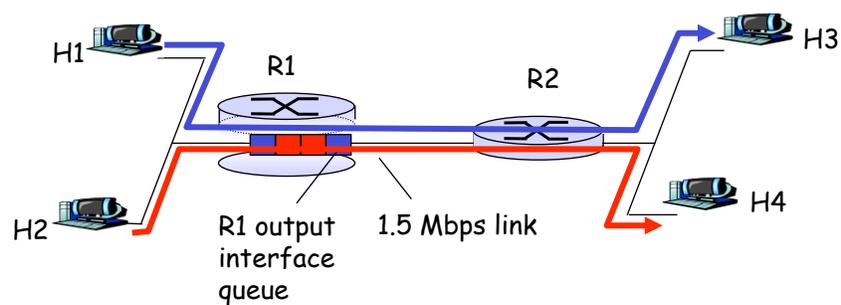
Providing Multiple Classes of Service

- 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 vs regular service)
- granularity: differential service among multiple classes, not among individual connections
- history: ToS bits



7: Multimedia Networking 7-53

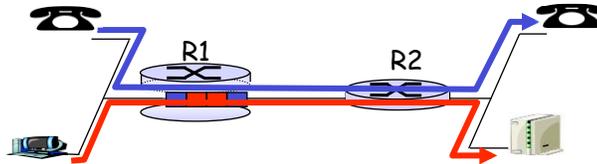
Multiple classes of service: scenario



7: Multimedia Networking 7-54

Scenario 1: mixed FTP and audio

- Example: 1Mbps IP phone, FTP share 1.5 Mbps link.
 - bursts of FTP can congest router, cause audio loss
 - want to give priority to audio over FTP



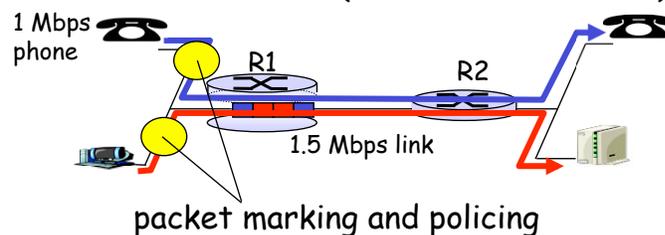
Principle 1

packet marking needed for router to distinguish between different classes; and new router policy to treat packets accordingly

7: Multimedia Networking 7-55

Principles for QOS Guarantees (more)

- what if applications misbehave (audio sends higher than declared rate)
 - policing: force source adherence to bandwidth allocations
- marking and policing at network edge:
 - similar to ATM UNI (User Network Interface)



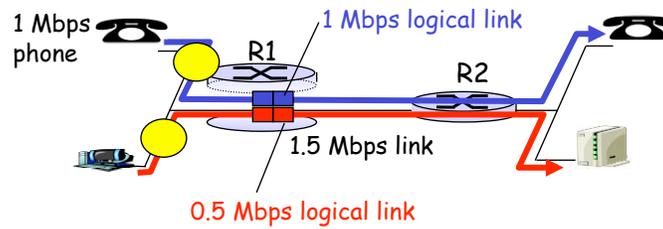
Principle 2

provide protection (*isolation*) for one class from others

7: Multimedia Networking 7-56

Principles for QOS Guarantees (more)

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



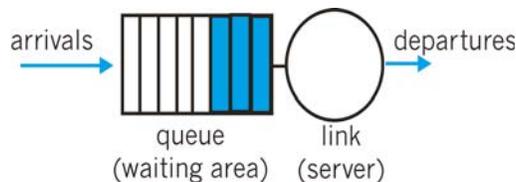
Principle 3

While providing isolation, it is desirable to use resources as efficiently as possible

7: Multimedia Networking 7-57

Scheduling And Policing Mechanisms

- **scheduling**: choose next packet to send on link
- **FIFO (first in first out) scheduling**: send in order of arrival to queue
 - real-world example?
 - **discard policy**: if packet arrives to full queue: who to discard?
 - Tail drop: drop arriving packet
 - priority: drop/remove on priority basis
 - random: drop/remove randomly

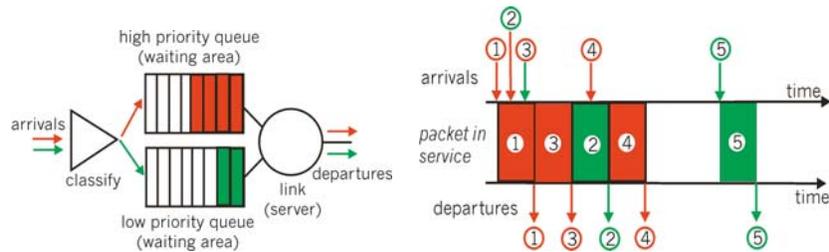


7: Multimedia Networking 7-58

Scheduling Policies: more

Priority scheduling: transmit highest priority queued packet

- multiple *classes*, with different priorities
 - class may depend on marking or other header info, e.g. IP source/dest, port numbers, etc..
 - Real world example?

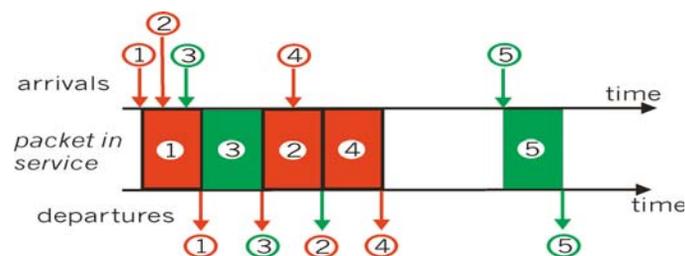


7: Multimedia Networking 7-59

Scheduling Policies: still more

round robin scheduling:

- multiple classes
- cyclically scan class queues, serving one from each class (if available)
- real world example?

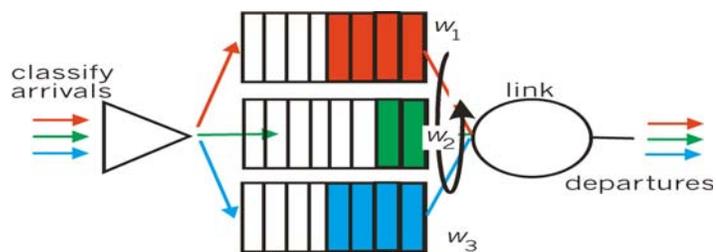


7: Multimedia Networking 7-60

Scheduling Policies: still more

Weighted Fair Queuing:

- ❑ generalized Round Robin
- ❑ each class gets weighted amount of service in each cycle
- ❑ real-world example?



7: Multimedia Networking 7-61

Policing Mechanisms

Goal: limit traffic to not exceed declared parameters

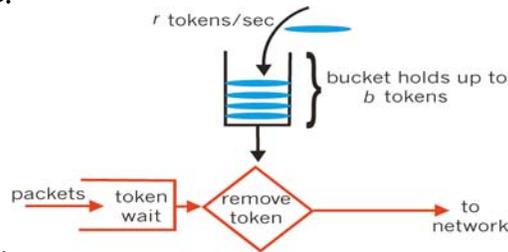
Three common-used criteria:

- ❑ *(Long term) Average Rate:* how many pkts can be sent per unit time (in the long run)
 - crucial question: what is the interval length: 100 packets per sec or 6000 packets per min have same average!
- ❑ *Peak Rate:* e.g., 6000 pkts per min. (ppm) avg.; 1500 ppm peak rate
- ❑ *(Max.) Burst Size:* max. number of pkts sent consecutively (with no intervening idle)

7: Multimedia Networking 7-62

Policing Mechanisms

Token Bucket: limit input to specified Burst Size and Average Rate.

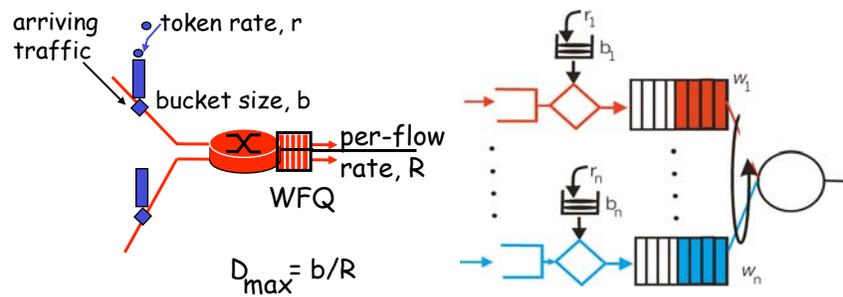


- bucket can hold b tokens
- tokens generated at rate r token/sec unless bucket full
- *over interval of length t : number of packets admitted less than or equal to $(r t + b)$.*

7: Multimedia Networking 7-63

Policing Mechanisms (more)

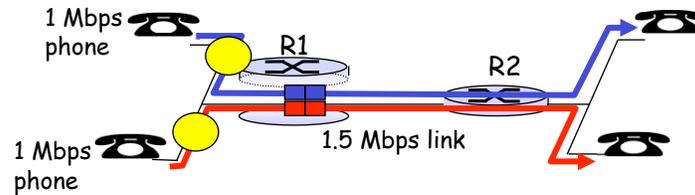
- token bucket, WFQ combine to provide guaranteed upper bound on delay, i.e., *QoS guarantee!*



7: Multimedia Networking 7-64

Principles for QoS Guarantees (more)

- *Basic fact of life:* can not support traffic demands beyond link capacity



Principle 4

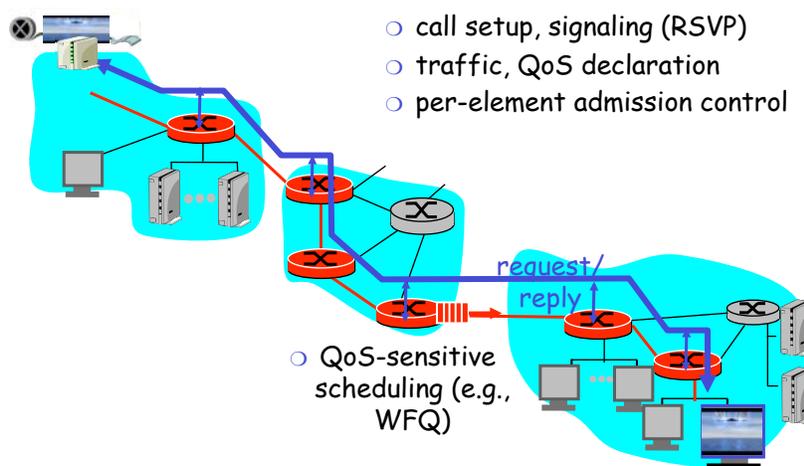
Call Admission: flow declares its needs, network may block call (e.g., busy signal) if it cannot meet needs

7: Multimedia Networking 7-65

QoS guarantee scenario

□ Resource reservation

- call setup, signaling (RSVP)
- traffic, QoS declaration
- per-element admission control



- QoS-sensitive scheduling (e.g., WFQ)

7: Multimedia Networking 7-66

Call Admission

Arriving session must :

- declare its QoS requirement
 - **R-spec**: defines the QoS being requested
- characterize traffic it will send into network
 - **T-spec**: defines traffic characteristics
- signaling protocol: needed to carry R-spec and T-spec to routers (where reservation is required)
 - **RSVP**

7: Multimedia Networking 7-67

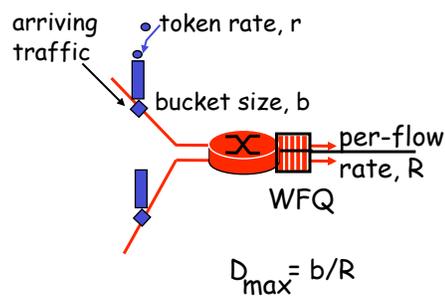
Intserv QoS: Service models [rfc2211, rfc 2212]

Guaranteed service:

- worst case traffic arrival: leaky-bucket-policed source
- simple (mathematically provable) *bound* on delay [Parekh 1992, Cruz 1988]

Controlled load service:

- "a quality of service closely approximating the QoS that same flow would receive from an unloaded network element."



7: Multimedia Networking 7-68

Chapter 7: Summary

Principles

- ❑ classify multimedia applications
- ❑ identify network services applications need
- ❑ making the best of best effort service

Protocols and Architectures

- ❑ specific protocols for best-effort
- ❑ mechanisms for providing QoS
- ❑ architectures for QoS
 - multiple classes of service
 - QoS guarantees, admission control