Multimedia networking
(chapter 7)

- New applications on the net
- Real-time applications
- Video, audio, streaming applications
  - Delay-sensitive
  - Loss-tolerant
  - Pandora, spotify, iheart radio, netflix, hulu
- Voice over IP
  - Interactive voice
  - Voice over data networks

Types of media

- Streaming media, play while the new content is being downloaded across the network
  - Content stored in servers
- Live media, play as and when content is generated
  - Unicast or multicast
- Interactive or real-time media, content is exchanged among participants in real-time
  - VoIP, teleconferencing etc

Digital representation

- Must convert analog signal to representation
  - Sample
    - How many times (twice the max frequency in the signal)
  - Quantize
    - How many levels or bits to represent each sample
    - More levels → more accuracy
    - More levels → more bits, more bw
  - Compress
    - Compact representation of quantized values

Audio

- Audio range: 20 Hz to 22.05 kHz, Speech
  - 200Hz, 8 kHz
    - Sample up to 16 kHz, but typically 8 kHz
    - Quantization 8 bits (256 levels), 16 bits (65,536 levels)
    - Bit rate for speech 8 k samples/sec * 8 bits = 64 kbps or 8 Kbps
    - Bit rate for audio 44.1 k samples * 16 bits = 705.6 Kbps
    - Bit rate for stereo = 2 * 705.6 = 1.4112 Mbps
    - Compressed MP3 128 Kbps
Image sampling

- Sample = pixel
- Image size = Height x Width
  - 320 x 240
  - 640 x 480
  - 1920 x 1080
- B&W Quantization: 256 levels (gray) - 8 bits per pixel
- Color quantization: 3 Colors x 256 levels - 8 bits per pixel
- Size = 320 x 240 x 8 = 61440 bits, 7680 bytes (raw)
- Compress using JPEG, GIF, TIFF, etc.

Video sampling

- Resolution
  - 704 x 480 aspect ratio 4:3 - Standard TV
  - 1280 x 720 - 16:9 - 720p HD
  - 1920 x 1080 - aspect ratio 16:9 - Full HDTV
- Frame rate:
  - Movies
    - 24 fps, 48 fps frames per second
  - HDTV (Digital)
    - 50 or 60 frames per second
    - 120 Frames per second - Ultra HDTV

Need for compression

- Size of video (color)
  - 1920 x 1080 x 24 bits/pixel = 6.42 MB (uncompressed)
  - 24 FPS = 153.96 MB/sec  60 FPS = 384.912 MB/sec
  - 1 Minute = 9.24 GB
  - 1 Hour = 554.27 GB
  - 5 Hours = 2.771 TB
- Need for compression
- Look at frequency of values (how often something appears)
- Use least number of bits for those values
- Audio MP3: 129 Kbps
- Audio JPEG, TIFF, GIF
- Video MPEG, MPEG-4: 1.5 Mbps, 64 Kbps
- Teleconferencing: H.261 40 Kbps - 2 Mbps

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., 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
  - Example rates
    - CD: 1.411 Mbps (2 x 44,100)
    - MP3: 64, 128, 160 kbps
    - Internet telephony: 5.3 kbps and up
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

Sending Streaming content

- Audio or video signal is sampled
- Number of samples (Digitized)
- Number of bits per sample (Quantization)
  - Examples (PCM): 8000 Hz or samples/sec at 8 bits per sample 64 Kbits/sec
  - CD mono 44 KHz at 16 bits per sample 705.6 Kbps
  - CD stereo 1.411 Mbps
- Actual number of bits (encoding/compression)
  - MP3 (compression) 96 Kbps to 160 Kbps
  - MPEG-4
  - How much to put in a packet (length of the sample) or frame length
- For audio: let us say 40 msec length of PCM quality
  - Which gives 320 byte packets
  - 64k * 40 * 10^3 / 8 = 320 Bytes

Multimedia Protocols

- RTP, RTCP - time stamps
- RTSP - streaming
- SIP - signaling

Client-Server Model

- Server: runs software which is waiting for requests
- Client: contacts the server for a particular service (e.g. web page transfer)
Information needed to send streaming data

- Packet # or sequence number
- Timestamp (receiver has an idea when to deliver the packet to output)
- Type of encoding/compression used etc
- Source identifier
  - left microphone, right microphone etc
- A new protocol for transferring multimedia data -- RTP

RTP

- Real-time transport protocol
- Lightweight, flexible
  - Provides mechanisms for transporting multimedia
  - No algorithms specified
- Scalable
  - Works with unicast and multicast

RTP

- Provides mechanisms for data and control
- Data
  - Timing, loss detection, content labeling, etc
- Control/RTCP
  - Periodic messages giving feedback about quality

RTP encapsulation
RTP header

```
0 1 2 3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|V=2|P|X|  CC   |M|     PT      |       sequence number         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           timestamp                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          synchronization source (SSRC) identifier            |
+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
|            other fields            |
|                             ....                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

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.

RTP header fields

- **Timestamp field (32 bits long):** increments 1 per sample, random starting value, a 20 msec sample at 8 KHz (has 160 samples), timestamp increments by 160 per RTP packet
- **Synchronization source identifier (SSRC) 32 bits long:** Identifies the source of the stream. A given multimedia stream can have many sources
- **Left microphone, right microphone**
Sources
- End systems: cameras, microphones
- Translators: Forward RTP packets after changing encoding
- Mixers: Receives RTP packets from multiple sources, combines them and/or changes encoding, forwards new RTP packets

Source identification
- Synchronization Source (SSRC):
  - source of a stream of RTP packets
  - globally unique, randomly generated ID
  - can be an originating source (microphone) or a mixer
- Contributing Source (CSRC):
  - actual source of data in the packet.

RTCP
- For transmission of control information
  - Receiver statistics (Receiver report)
  - Source statistics (Sender report)
  - Source information (Source Description)
- Designed to be scalable

RTP/RTCP
- Typically used with UDP as transport
- Used in conjunction with RTCP or RTP control protocol
- RTCP provides feedback to sender with receiver reports and to the receiver with sender reports
- RTCP sender reports (SR)
  - Total packets sent, wall clock time, etc
- RTCP receiver reports (RR)
  - Packet loss, jitter etc
Receiver Report
- SSRC of the source
- Lost packets
  - cumulative, fraction
- highest sequence number received
- interarrival jitter
- last source report

Source Report
- SSRC of the sender
- wallclock time (NTP format)
- RTP timestamp
- number of packets transmitted
- number of bytes transmitted

RTP/RTCP
- Provides
  - Time stamps
  - Sequence numbers
  - Feedback
  - Source/Payload identification
- Multicast friendly
- Scalable

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

- allows for non-HTTP protocol between server, media player
- UDP or TCP for step (3), more shortly

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>
    <track type=audio e="PCMU/8000/1" src="/audio.example.com/twister/audio.en/lofi">
    <track type=audio e="DVI4/16000/2" pt="90 DVI4/8000/1" src="/audio.example.com/twister/audio.en/hifi">
  </group>
  <track type=video/jpeg src="/video.example.com/twister/video">
</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
Range: npt=0-
Session: 4231
C: PAUSE rtsp://audio.example.com/twister/audio.en/lofi RTSP/1.0
Range: npt=37
Session: 4231
C: TEARDOWN rtsp://audio.example.com/twister/audio.en/lofi RTSP/1.0
S: 200 3 OK

DASH

- Dynamic Adaptive Streaming over http
- Used by Netflix and other video streaming services
- Client centric approach to video delivery
- Client performs rate adaptation
- Server is standard http server
- Provides content in multiple formats, multiple encodings
- Allows use of http CDNs

Source: Stockhammer MMSys 2011, Sadagar IEEE Mmedia 2011
DASH Data model

Media has several periods
Each period has several Adaptation Sets: Audio, video, closed caption
Several Representations per Adaptation
Several segments per Representation

Media Presentation Descriptor

- MDP requested over http
- Describes all segments
- Timing information, byte ranges

Dynamic changing of streams

DASH Summary

- Widely used in video streaming services
  - Silverlight, Apple streaming, HTML5, Flash
- Allows independent request sizes and segment sizes
  - HTTP byte ranges, chunks, streamlets
- Combined and separate audio video streams
- Ad insertions
- Works well with CDNs
Internet telephony

- Phone service over internet
- Voice converted into packets
- Use RTP to transport audio to end-points
- VOIP: implement all features available in the traditional phone network (Public Switched Telephone Network-PSTN)

Main Problems: Should interoperate with existing PSTN
Need to identify endpoints

VOIP modes

I) Caller has phone,
II) Callee has Phone
III) Both have IP Phones-PCs

Session initiation protocol

- SIP core protocol for establishing sessions
- Handles setup, modification, and teardown of VoIP session
- Transports session information among end-points
- Signaling involves sending messages

SIP addressing

- SIP uses naming similar to e-mail
- SIP URLs
  - SIP:badri@cs.rutgers.edu
  - SIP:17324452082@rutgers.edu
  - SIP:17324452082@rutgers.edu;user=phone
SIP allows mobility of endpoints

SIP network entities
- SIP user agent
  - IP phone, runs VoIP software
- SIP redirect server
  - Redirects connection requests
- SIP proxy server
  - Handles SIP requests on behalf of user
- SIP registrar
  - Maintains mapping from names to addresses

SIP redirect server
- Accepts SIP requests and maps destination address to zero or more new addresses
- Allows caller to contact new location directly
- Redirect server does not initiate any SIP requests of its own

SIP proxy server
- Accepts SIP requests on behalf of user
- Proxy server initiates SIP requests at new location if user has changed end-points
- Caller sees proxy server as end-point
- Proxy server hides end-point mobility from callee
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

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
```

Locating users
- Find Domain of proxy server using SIP URL
- From DNS lookup find proxy servers (registrar as well)
- From proxy server learn IP address of endpoint
- May have to go through multiple redirections (URL remappings)
SIP messages

- Similar to HTTP
- Request, responses are text messages
- Messages can contain body which relate to session descriptions

SIP methods

- INVITE – initiate call
- ACK – confirm response
- BYE – terminate call
- CANCEL – cancel searches
- OPTIONS - features supported on the other side
- REGISTER – register with location service

SIP responses

- Can convey a lot of status information with responses
- 200: OK
- 100 : TRYING
- 180 : Ringing
- 181: call forwarded
- 300: multiple choices

SIP call Establishment
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.

PSTN internetworking

- A gateway is required to convert SIP messages to circuit switched signaling and vice-versa
- Proxy server will contact PSTN gateway
- PSTN gateway initiates call to the PSTN callee
- Two-way audio conversation occurs through the gateway

Network support for Multimedia Ch 7.5

Best effort service
- Does not offer any guarantees on delay, bandwidth, and loss
- Many applications (IP telephony, video, etc.) require delay bounds
- QoS in a network principles:
  - Network, application-level support for multimedia
  - Making the best of best effort service
  - Mechanisms for providing QoS
- Specific protocols, architectures for QoS
Protocols for QoS

- Support protocols for multimedia applications and requirements
- Making the best of today’s best effort service
- Policing mechanisms
- Resource reservation protocols
  - Intserv
  - RSVP
  - Diffserv

Multimedia, Quality of Service: What is it?

Multimedia applications: network audio and video

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

Multimedia Performance Requirements

Requirement: deliver data in “timely” manner

- Interactive multimedia: short end-end delay
  - e.g., IP telephony, teleconf., virtual worlds, DIS
  - Excessive delay impairs human interaction
- Streaming (non-interactive) multimedia:
  - Data must arrive in time for “smooth” playout
  - Late arriving data introduces gaps in rendered audio/video
- Reliability: 100% reliability not always required

Policing mechanisms = Leaky Bucket

- Used in conjunction with resource reservation to police the host’s reservation
- At the host-network interface, allow packets into the network at a constant rate
- Packets may be generated in a bursty manner, but after they pass through the leaky bucket, they enter the network evenly spaced
Leaky Bucket: Analogy

Leaky Bucket (cont’d)
- The leaky bucket is a “traffic shaper”: It changes the characteristics of packet stream
- Traffic shaping makes more manageable and more predictable
- Usually the network tells the leaky bucket the rate at which it may send packets when the connection begins

Leaky Bucket: Analogy

Doesn’t allow bursty transmissions
- In some cases, we may want to allow short bursts of packets to enter the network without smoothing them out
- For this purpose we use a token bucket, which is a modified leaky bucket

Token Bucket
- The bucket holds tokens instead of packets
- Tokens are generated and placed into the token bucket at a constant rate
- When a packet arrives at the token bucket, it is transmitted if there is a token available. Otherwise it is buffered until a token becomes available.
- The token bucket has a fixed size, so when it becomes full, subsequently generated tokens are discarded
**Token Bucket**

Packets from host

Token Generator
(Generates a token once every T seconds)

Network

---

**Token Bucket vs. Leaky Bucket**

**Case 1: Short burst arrivals**

Arrival time at bucket

Departure time from a leaky bucket
Leaky bucket rate = 1 packet / 2 time units
Leaky bucket size = 4 packets

Departure time from a token bucket
Token bucket rate = 1 token / 2 time units
Token bucket size = 2 tokens

**Case 2: Large burst arrivals**

Arrival time at bucket

Departure time from a leaky bucket
Leaky bucket rate = 1 packet / 2 time units
Leaky bucket size = 2 packets

Departure time from a token bucket
Token bucket rate = 1 token / 2 time units
Token bucket size = 2 tokens

---

**IETF Differentiated Services**

**Problems with Intserv:**

- **Scalability:** signaling, maintaining per-flow router state difficult with large number of flows
- **Flexible Service Models:** Intserv has only two classes. Also want "qualitative" service classes
  - "behaves like a wire"
  - relative service distinction: Platinum, Gold, Silver

**Diffserv approach:**

- simple functions in network core, relatively complex functions at edge routers (or hosts)
- Don't define service classes, provide functional
**Diffserv Architecture**

**Edge router:**
- per-flow traffic management
- marks packets as in-profile and out-profile

**Core router:**
- per class traffic management
- buffering and scheduling based on marking at edge
- preference given to in-profile packets
- Assured Forwarding

**Edge-router Packet Marking**
- profile: pre-negotiated rate A, bucket size B
- packet marking at edge based on per-flow profile

Possible usage of marking:
- class-based marking: packets of different classes marked differently
- intra-class marking: conforming portion of flow marked differently than non-conforming one

**Classification and Conditioning**
- Packet is marked in the Type of Service (TOS) in IPv4, and Traffic Class in IPv6
- 6 bits used for Differentiated Service Code Point (DSCP) and determine PHB that the packet will receive
- 2 bits are currently unused

**Classification and Conditioning**
- may be desirable to limit traffic injection rate of some class:
  - user declares traffic profile (e.g., rate, burst size)
  - traffic metered, shaped if non-conforming
### Per hop behavior

- PHB result in a different observable (measurable) forwarding performance behavior
- PHB does not specify what mechanisms to use to ensure required PHB performance behavior
- Examples:
  - Class A gets x% of outgoing link bandwidth over time intervals of a specified length
  - Class A packets leave first before packets from class B
  - Class B Packets get dropped over Class A packets

### Implementing PHB

- **Bandwidth**
  - Can use multiple queues
  - Schedule packets in one queue more often than packets in the other queue: Round robin with different rates
- **Delay**
  - Use priority queue; schedule lower level queue after higher level queue
- **Loss**
  - When need to drop packets prefer one class over another