Where we are in the Course

- Revisiting the layers
  - Quality of Service (QOS) involves both the Network and its users/applications
**Topic**

- QoS relates to the kind of service a user gets from the network
  - E.g., high/low bandwidth, delay, loss
  - Important issue for future Internet
“Best Effort” Service

• What we get in the Internet today with FIFO routers
  – Apps compete for bandwidth; queues add delay and loss
  – Try to deliver with no guarantee of bandwidth, delay, loss
QOS Motivation

• Best effort is not always enough!
  – May want performance guarantees

• What can’t be done:
  – Guarantee more bandwidth or lower delay than exists in the network

• What can be done:
  – Control how bandwidth (hence delay/loss) is allocated to different users
Example – Web and BitTorrent

- Home user browses the Web and runs BitTorrent at the same time
  - Assume access link is the bottleneck
  - What happens? What do we want?
Web and BitTorrent (2)

- What happens?
  - Web and BitTorrent compete for downstream bandwidth using TCP
  - Queues build at ISP end of access...
Web and BitTorrent (3)

- What happens?
  - Web PLT rises because of BitTorrent
  - Less web bandwidth, more delay/loss
Web and BitTorrent (4)

- What do we want to happen?
  - Web is interactive, while BitTorrent runs in the background
  - Prefer to use bandwidth for Web
Web and BitTorrent (5)

- What do we want to happen?
  - Suppose we modify ISP router to give priority to Web packets on access link.
Web and BitTorrent (6)

• What do we want to happen?
  – Would minimize web PLT for user
  – BitTorrent just has less bandwidth
Example – Skype and BitTorrent

- Home user skypes (VoIP only) and runs BitTorrent at the same time
  - Assume access link is the bottleneck
  - What happens? What do we want?
Skype and BitTorrent (2)

• What happens?
  – Skype and BitTorrent compete as before, though not with TCP
  – Queues build at ISP end of access...
Skype and BitTorrent (3)

- What happens?
  - Skype call quality falls due to BitTorrent
  - More delay/loss; little bandwidth issue
Skype and BitTorrent (4)

• What do we want to happen?
  – Skype real-time, BitTorrent background
  – Prefer low-delay for Skype and high-bandwidth for BitTorrent
Skype and BitTorrent (5)

- What do we want to happen?
  - Modify ISP router to give priority to Skype packets on access link
Web and BitTorrent (6)

- What do we want to happen?
  - Maximizes skype call quality without slowing BitTorrent – both win!
QOS Motivation (2)

• Opportunity to allocate bandwidth to improve app/user performance
  – Guarantee bandwidth to an app
  – Satisfy multiple apps at once

• To provide QOS, we need to know what apps require of the network
  – Need for bandwidth, delay, loss
Application Requirements

- **HIGH stringency means high bandwidth, low delay/loss**

<table>
<thead>
<tr>
<th>Application</th>
<th>Bandwidth</th>
<th>Delay</th>
<th>Jitter</th>
<th>Loss</th>
</tr>
</thead>
<tbody>
<tr>
<td>Email</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
<td>Medium</td>
</tr>
<tr>
<td>File sharing</td>
<td>High</td>
<td>Low</td>
<td>Low</td>
<td>Medium</td>
</tr>
<tr>
<td>Web access</td>
<td>Medium</td>
<td>Medium</td>
<td>Low</td>
<td>Medium</td>
</tr>
<tr>
<td>Remote login</td>
<td>Low</td>
<td>Medium</td>
<td>Medium</td>
<td>Medium</td>
</tr>
<tr>
<td>Audio on demand</td>
<td>Low</td>
<td>Low</td>
<td>High</td>
<td>Low</td>
</tr>
<tr>
<td>Video on demand</td>
<td>High</td>
<td>Low</td>
<td>High</td>
<td>Low</td>
</tr>
<tr>
<td>Telephony</td>
<td>Low</td>
<td>High</td>
<td>High</td>
<td>Low</td>
</tr>
<tr>
<td>Videoconferencing</td>
<td>High</td>
<td>High</td>
<td>High</td>
<td>Low</td>
</tr>
</tbody>
</table>

Variation in delay
Topics

• Application requirements
• Real-time transport (VoIP)
• Streaming media (video)
• Fair Queuing
• Traffic Shaping
• Differentiated services
• Rate/Delay guarantees
**Topic**

- Sending interactive real-time media over the network, e.g., VoIP
  - Using the best effort Internet
  - Playout buffer technique
Challenge – Network Delay

- Consider one direction
  - Constant rate of media is generated at source, later consumed at receiver
  - Network must have enough bandwidth, and adds a delay
Network Delay (2)

- Network delay is variable
  - Message latency plus queuing delay
  - Variability in delay is called jitter
Playout

- Ideally want fixed, and small network delay for interactivity
  - Emulate the telephone network
Playout (2)

- Media arrives at receiver after variable network delay
Playout (3)

- Media arrives at receiver after variable network delay
Playout Buffer

- Put media in **playout buffer** at receiver until consumption time
  - Smooth out variable network delay
Playout Buffer (2)

- Media arrival curve determines time in playout buffer and deadline
Playout Buffer (3)

- Pick largest acceptable network delay to set the playout point

![Graph showing latency, queuing, and max delay with a too late indication.]
Playout Buffer (4)

• Tradeoff:
  – Larger acceptable network delay → larger buffer/delay, less loss
  – Smaller acceptable network delay → smaller buffer/delay, more loss

• Typically can’t recover loss for interactive, real-time scenario
  – Instead, do without (glitch)
Topic

• Playback of media over the network
  – Using the best effort Internet
  – YouTube, Netflix, etc.
  – Huge usage!
Streamed vs. Interactive Media

- Streamed is less demanding case:
  - Only a single direction to consider
  - Low delay not essential; affects startup but not interactivity
  - Still need to handle bandwidth, jitter
Handling Jitter

- As before, buffer media at receiver until ready for playout time
  - Smooth out variable network delay
Handling Jitter (2)

- Use **HIGH** and **LOW** watermarks to control source over/underfill
  - Start pulling media at low level
  - Stop pulling media at high level
Handling Bandwidth

• Send file in one of multiple encodings
  – Higher quality encodings require more bandwidth
  – Select best encoding given available bandwidth

Higher quality
More bandwidth

15:1

23:1

46:1

Lower quality
Less bandwidth

144:1

By Toytoy, CC-BY-SA-3.0, from Wikimedia Commons
Streaming over TCP or UDP?

- UDP minimizes message delay for interactive, real-time sessions
- TCP is typically used for streaming
  - Low delay is not essential; startup
  - Loss recovery simplifies presentation
  - HTTP/TCP passes through firewalls
Streaming with RTSP

- Video started using HTTP to get metafile
- Invokes media player
  - Talks RTSP (Real-Time Streaming Protocol) to media server
- Media sent with, e.g., RTP over TCP/UDP
Streaming with HTTP

- Fetch media description data
  - Gives index of clips, rates
- Fetch small segments
  - Put in playout buffer
- Adapt selection of encoding
  - Based on buffer occupancy
- Standards, e.g., DASH
  - Leverages HTTP

Diagram: GET INDEX, GET MEDIA