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 within and between images to decrease # bits used to encode image
  - spatial (within image)
  - temporal (from one image to next)

Coding examples:
- Spatial coding example: instead of sending $N$ values of same color (all purple), send only two values: color value (purple) and number of repeated values ($N$)
- Temporal coding example: instead of sending complete frame at $i+1$, send only differences from frame $i$

Streaming stored video:

Simple scenario:

- Video server (stored video)
- Internet
- Client
Multimedia vs. Other Applications

**Classes of multimedia applications:**
1) Stored streaming
2) Live streaming
3) Interactive, real-time

Packet Loss?
- Tolerant?
- Intolerant?

Variable delay between packets (jitter)?
- Tolerant?
- Intolerant?

1) Best Effort Service, UDP

- Laissez-faire approach
- No major changes, no guarantees for delay or loss
- Options with historical/existing Internet architecture
  1. Use UDP
  2. Delay playback by > 100ms
  3. Timestamp packets (chunks of data)
  4. Prefetch data
  5. Send redundant packets (to mitigate loss)
- Content distribution network (CDN)
- Application-layer multicast
- More bandwidth will be available as needed

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
- UDP might not get through firewalls

2) Differentiated Quality of Service

- Differentiated services
  - Implemented via HTTP with evolution toward DASH
- Few changes to Internet infrastructure
- Provide 1st and 2nd class service
- 1st Class
  - Limit the number of 1st class packets
  - These receive priority in router queues
- Net neutrality?
3) Guaranteed Quality of Service

- First implementation is “DASH”
- Fundamental changes for the Internet
  - Protocols to reserve link bandwidth for entire path, from sender to receiver
  - Modify router queues so reservations can be honored
  - To identify honored packets, applications must be able to label packets as such
  - Network must be able to determine if there is sufficient bandwidth
- Requires new & complex software in hosts & routers

Streaming multimedia: DASH

- DASH: Dynamic, Adaptive Streaming over HTTP
- “intelligence” is implemented at client: client determines
  - when to request chunk (so that buffer starvation, or overflow does not occur)
  - what encoding rate to request (higher quality when more bandwidth available)
  - where to request chunk (can request from URL server that is “close” to client or has high available bandwidth)

Streaming stored video:
Streamlining stored video: with delay

Discussion Questions
- Suppose that the client begins playout as soon as the first block arrives at \( t_1 \). In the figure, how many blocks of video (including the first block) will have arrived at the client in time for their playout? Discuss.
- Suppose that the client begins playout now at \( t_1 + \Delta \). How many blocks of video (including the first block) will have arrived at the client in time for their playout? Discuss.
- In the same scenario as above, what is the largest number of blocks that is ever stored in the client buffer, awaiting playout? Discuss.
- What is the smallest playout delay at the client, such that every video block has arrived in time for its playout?

Summary for Part 1
- Evolution: UDP \( \rightarrow \) HTTP \( \rightarrow \) DASH
- Streaming stored video