Massachusetts Institute of Technology
Department of Electrical Engineering and Computer Science

6.892 Fall 1999 **Adaptive Applications** November 1, 1999

Overview. Application-level framing (ALF). RTP as an example. ILP. Adaptation for audio/video transmissions on the Internet: delay and quality adaptation.

1 ALF

- ALF allows the expression of application semantics on the design of protocol. One size *does not* fit all!
- If there aren't any losses or reordered packets, things usually go fine...
- But if not, mismatches between application and protocol destroy efficiency and performance. (Q: why does reordering happen? Why does performance degrade?)
- Example: Webcasting. TCP's in-order, reliable delivery not the best model. (But that's what's used today!). Can you use TCP to implement selective reliability? Another example is real-time multimedia transmission.
- ADU: Application Data Units. These are what application want, so don't force them to tailor to a particular type of network or technology or protocol semantics. ADU's can be processed in *any order*, so long as they are complete. Transport hands complete ADU to the app, but doesn't worry about preserving inter-ADU ordering. This speeds up "presentation" processing and improves user-perceived latency.
- Questions: How to construct and name ADU? Can we always make ADU = 1 packet?
- ALF *does not* require each application to design its own transport. Typically, one can use a library to achieve code reuse. Modularity instead of layering.
- Example: RTP (Realtime Transport Protocol), RFC 1889. Use class hierarchy of protocols: `RTP --- A/V profile --- H.261 video` is one hierarchy. RTP defines base packet formats, which are customizable and extensible by derived classes.
- RTP provides "thin" transport support that is malleable, allowing it to be easily integrated into application processing.
- Simultaneous with data distribution, RTP uses a control protocol called RTCP (Realtime Transport Control Protocol). Participants in an RTP session periodically announce their status on a multicast channel using RTCP. Bandwidth used by RTCP is dynamically adjusted as a function of the scale of the session.

1.1 ILP

- Layering fosters sequential processing of each layer (ease of implementation).
- Remember, layering is only a design choice, not a law.
- ILP motivated by widening memory/CPU bottleneck.
- Process all "layers" in one integrated loop, perform multiple manipulation (and control) ops in one loop.
- Have to minimize precedence/ordering constraints. (Compiler help).
- Practice
  - Holy grail: many papers in the 1990s. Approaches include automatic synthesis, compiler support, formal languages, etc. Difficult problem.
  - In many cases, application is the bottleneck, or network bandwidth is...

2 Adaptation for audio/video

- Transmission rate: rate at which sender sends data, governed by a congestion control algorithm (e.g., AIMD).
- If playback rate (consumption rate) at receiver is always equal to transmission rate (with time shift for propagation time on forward path), then video quality will vary a lot and be annoying to the viewer.
- Jitter (variation in delay differentials) in the network also needs to be handled.
- These are handled by *playout* or *playback* buffers at the receiver.
- The goal of a playback buffer is to ensure that the "right" consumption rate that reduces viewer disruption and variation in quality is achieved. It does so by buffering data to smooth out bursts and variations.

2.1 Receiver adaptation to cope with packet jitter

- Packet video/audio is conceptually easy: digitize analog signal, compose bit-stream into packets, transmit packets, reassemble at receiver to get bit-stream, deliver to audio/video codec for decoding.
- Best-effort network has variable delays and losses. How to cope with delay variations?
- Jitter $= |A_i - S_i - (A_i - S_{i-1})|$. Calculate smoothed running average of jitter. Set playout buffer to be a small number of std deviations from the mean jitter and absorb. Packets arriving *after* playout time are dropped. If buffer is big enough, can absorb large jitter. If too small, loss occurs. If too big, long delays ensue. Therefore need adaptation.

2.2 Quality adaptation for video playback

- Paper (RHE99) solves problem of "hiding" congestion controlled rate variations from receiver, by using buffering to smooth out fluctuations.
- Three options: (i) re-encode dynamically at source on rate change; (ii) maintain multiple formats; (iii) use layered (hierarchical) data encoding.
- Two questions:
  1. When to add a layer?
  2. When to drop a layer?
  3. How to allocate space amongst layers in buffer?
- When adding new layer, ensure that if the network gets congested soon after and the rate drops to half of the current value, we can continue to achieve the same quality as with the introduction of the new layer.
- When to drop layer? Upon a rate decrease, drop highest layer. But we may need to drop more, esp. because layers are additively spaced. Iteratively do so until buffering is sufficient for the drain phase to work.
- How to allocate buffering to layers? Lower layers are more important, but no layer needs more than C bits/s since that's the consumption rate of a layer.
- Note: Paper has nice pictures and actual equations, which are easy to see pictorially. I haven't reproduced them in these notes, and we did derive them in class.