Distributed Multimedia

March 19, 2001

What is Distributed Multimedia?

- Large quantities of distributed data
- Typically streamed out
- One or many receivers of the data
- Run over general purpose infrastructure
- Data is time sensitive, but not necessarily real time

Four Phases

- Encoding
- Storage (not always required)
- Transport
- Decoding

➡ Need to focus on the bottlenecks!

Transport: Quality of Service

- Issue: Gracefully and dynamically manage against the underlying infrastructure's changing behavior
- Approaches: caching, priorities, resource availability modeling, compression
- Similar to locking - QoS tries to guarantee that a set of resources will be available

➡ Asynchronous network forces us to make tradeoffs

QoS Concerns

- Latency
- Bandwidth
- Loss Rate
- Bursting
- Jitter

➡ This, in a heterogeneous environment

QoS Model

- Some elements of the transport are harder or more expensive to control
- Harder or more expensive:
  - Congestion over a Wide Area Network
  - Protocols used over non-local networks
- Easier
  - Congestion over a Local Area Network
    - e.g., use priorities, increase buffer size
  - Protocols used by the ends of the connection

Resource Reservation Protocol

- Client asks QoS manager for resources
- If resources are available, QoS manager makes reservation (allocates resources for client)
- Otherwise, client can ask for less resources
- Monotonic - QoS manager should not take away a reservation, but client can ask for more resources at any time

Fairness

- Allow maximum resource utilization without allowing resource hogging
- Simple approach - Round robin scheduling
  - Each of N clients gets 1/N of the resource
  - 1/N must be at least what the client reserved from the QoS manager
- Fair Queuing
  - Ensures that over time, resource allocation is fair
    - The more fine-grained, the more fair, the more of a performance hit
  - Weighting can give some clients priorities

Traffic Shaping

- The server can help to ensure QoS requirements are met
- Traffic shaping uses a buffer to control:
  - When data is sent
  - How large a message is sent
- Can control bursting and jitter
- Can manage bandwidth

Buckets for Traffic Shaping

- Leaky bucket
  - R: Max. rate of message sends
  - B: Buffer size
  - Eliminates bursts completely
- Token bucket - Allow bursts bounded by available bandwidth
  - Tokens generated at a fixed rate R
  - A message can be sent immediately if there is a token in the bucket
  - If messages is not available, tokens accumulate

Practical QoS

- QoS is probabilistic - no 100% guarantee in an asynchronous network
- Need to adapt when QoS promises cannot be met
- Simple: Drop packets
  - Problem: Decision made at too low a layer
- Better: Decide what to do at higher layers
  - e.g., choose which messages to drop, choose when to burst messages
  - Use a holistic approach

Scaling & Filtering

- Scaling - start with high-quality stream at the server and adapt
  - Temporal - e.g., less video frames
  - Spatial - e.g., smaller video frame
  - Frequency - e.g., better compression/lower quality
  - Amplitudal - e.g., lower color depth on a per-pixel basis
  - Color map - e.g., less available colors overall
- Filtering - Use scaling at a more fine-grained level