Lecture #03

January 11, 2010

Today's Objectives

- Discuss upcoming classes
- Different spectrums (spectra) of multimedia networking
- Project ideas/discussion/meetings

Notes

- Papers we will be covering span a wide range of time
  - Pay attention to dates, use as contextual information
  - Old is not necessarily bad
- Presented as “spectrum”
  - Interesting parts are often navigating the middle
- Topics in red are specific topics to cover this quarter
- Difference between foundational papers, surveys, and papers about recent innovations

Spectrum #1: Downloading v. Streaming

- Download content
  - Download the whole thing first
  - Disad: if file is large, it takes a long time
- Progressive download (or streaming with buffering)
  - Download part of the content first (same as streaming with a significant buffer)
  - Question is how much to download/buffer before starting
    - Wait a fixed amount of time
    - Dynamically determine based on download and playout rates
  - Also have to deal with buffer management
  - Some fairly sophisticated techniques for delivery
- Streaming content
  - Buffer as little as possible
  - Necessary for interactive applications

Spectrum #2: Different v. Identical Content

- Different Content
  - Deliver different data to users
  - Possibly lots of users (scalability)
    - Have content closer to users (content distribution networks)
    - Use distributed servers
    - Use p2p
  - Some advantages of server-side caching (but limited)

- Identical Content
  - Deliver the same data to users
  - Sub-spectrum #1: is data used at the same time?
  - Sub-spectrum #2: how is data delivered?

Spectrum #2A: Immediate v. Delayed Playout

- Everyone displays content at the same time
  - Examples: TV broadcast, video conference
- Requests for content are aggregated
  - Slight introduction of delay
- Push-Based content
  - Use local filtering and only store content of interest
  - Content is then accessed on-demand
- Give users choices based on existing data delivery
  - Similar to a jukebox service model

Spectrum #2B: Application v. Router Replication

- Unicast
  - One-to-one
- Multicast
  - One/Some/Many-to-One/Some/Many
  - Different ways of implementing are of interest
    - "natively": have routers do packet replication
    - "hierarchical": have *N* levels in the network
    - "appl layer": have user-level software act as replicators
- Broadcast
  - One/Some/Many-to-All

Spectrum #3: Stored v. Live Content

- Stored content
  - No limitations on what can be done
  - Still a question of scalability
    - But mainly a question of fast disks and network connections

- Live content
  - Limitations on type of encoding
    - Different types of encoding introduce more delay
  - Delay may or may not be important
    - Depends on whether content is interactive or not
    - For interactive content (conference), delay needs to be minimized
    - For one-way content (TV broadcast), delay is relatively unimportant

Spectrum #4: Active v. Passive Stream Control

- Active
  - Allow full VCR-style functionality (pause, rewind, fast forward)
  - If content is one-to-one and pure streaming, requests have to go to server
    - Common in the early days of streaming players (e.g., Real Player)
- Limited Active
  - Depends on other system characteristics
    - Ex: can't fast forward past live events (TiVo)
  - How data is delivered has a significant influence
    - Easiest with downloaded content: just manipulate locally
    - Hardest when streamed using multicast
- Passive
  - Standard playout only

Spectrum #5: Interactive v. Passive

- Interactive
  - Typically bi-directional with other participants
    - Ex: games, VoIP, conferencing
    - Can also be local real-time (e.g., augmented reality)
  - Delay becomes the most critical characteristic
  - Greatly limits what else can be done…
    - Type of encoding
  - …though not always
    - VCR-functionality as an example

- Passive
  - Constraints are much reduced

Spectrum #6: Best Effort v. QoS Data Delivery

- Best effort
  - There isn't much the network can do that it isn't already doing so just cope with that
    - Various clever ways of minimizing impact of delay (e.g., buffering, local decision making in games, smart encoding, loss minimization)
- QoS
  - Lots of research in QoS over the years
  - Most required fundamentally changing the Internet
    - Most radical change was to turn Internet into connection-oriented system using ATM
  - Best deployable scenario today: diffserv
    - Diffserv is as simple as using queues and prioritization
    - "If one packet gets better service, some other packet gets worse service"

Spectrum #7: Reliable v. Unreliable Delivery

- In most cases, this is a question of data
- TCP-style reliable data delivery
  - Data must be reliably delivered and in-order
  - Even for some MM-based apps, TCP is fine (YouTube)
- Partial reliability
  - Reliability need not be TCP-style in-order delivery
    - Watch ***and*** record a transmission
  - Only some data must be delivered reliably
  - Repair data can be requested and used if available
    - Possibly from neighbors receiving the same thing
    - Possibly use a threshold to determine when too much data is lost
- No reliability requirement
  - Various clever ways of minimizing impact of delay (e.g., buffering, local decision making in games, smart encoding, loss minimization)

Spectrum #8: Easy v. Hard Environments

- Easy
  - Wired, high-bandwidth, low-delay, low-loss, stable
- Hard
  - Wireless, low-bandwidth, jitter, high-loss, unstable, highly variable
- And all points in between
- Challenges can also come from the device
  - Limited capability (CPU, memory, display)
  - Limited battery power

Project Ideas Status

Project Ideas

- Link is posted
  - http://www.cs.ucsb.edu/~almeroth/classes/W10.290F/ideas.html
  - Read the NSF report!!
- Other ideas
  - Wireless video transmitter
  - Augmented reality (any existing software package?)

Keep in Mind…

- Whether your project requires specialized hardware
- How to evaluate your idea
  - A prototype is fine, but what question does it answer