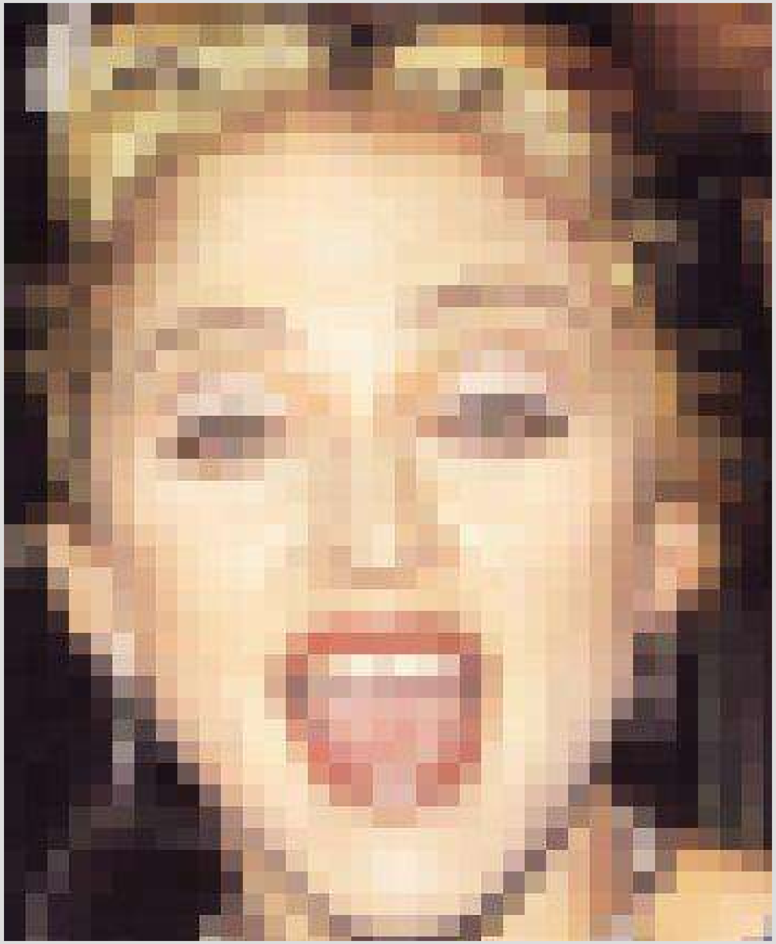
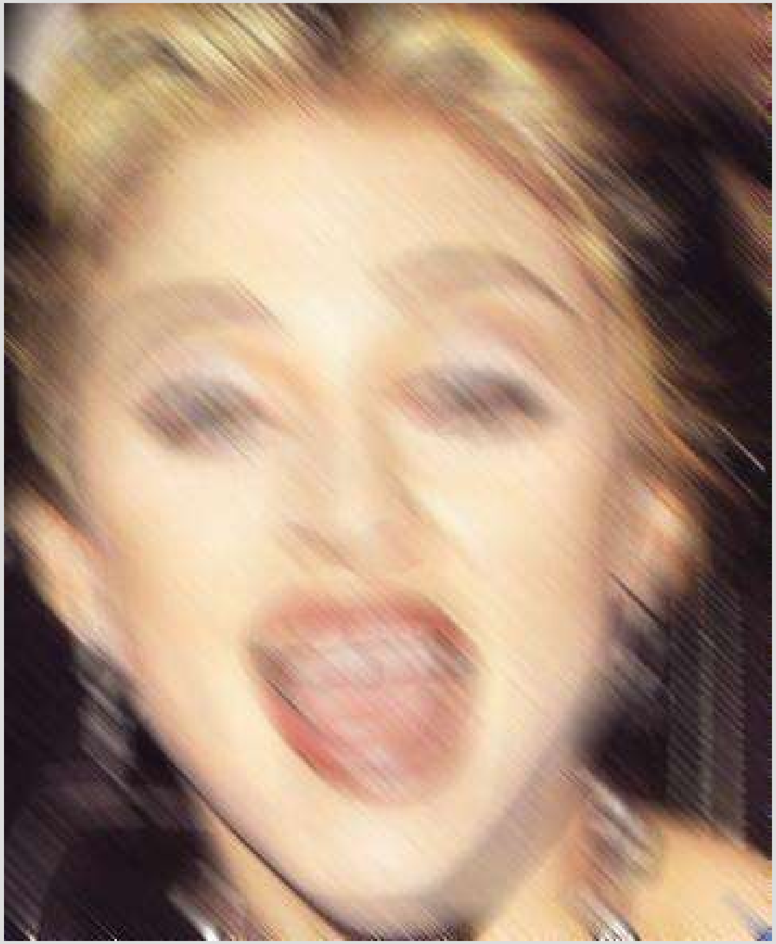
Streaming Audio and Video over the Internet: Challenges and Pitfalls

Amy Csizmar Dalal
Department of Mathematics and Computer Science
Carleton College

September 28, 2004

photo courtesy of http://www.madonnashots.com





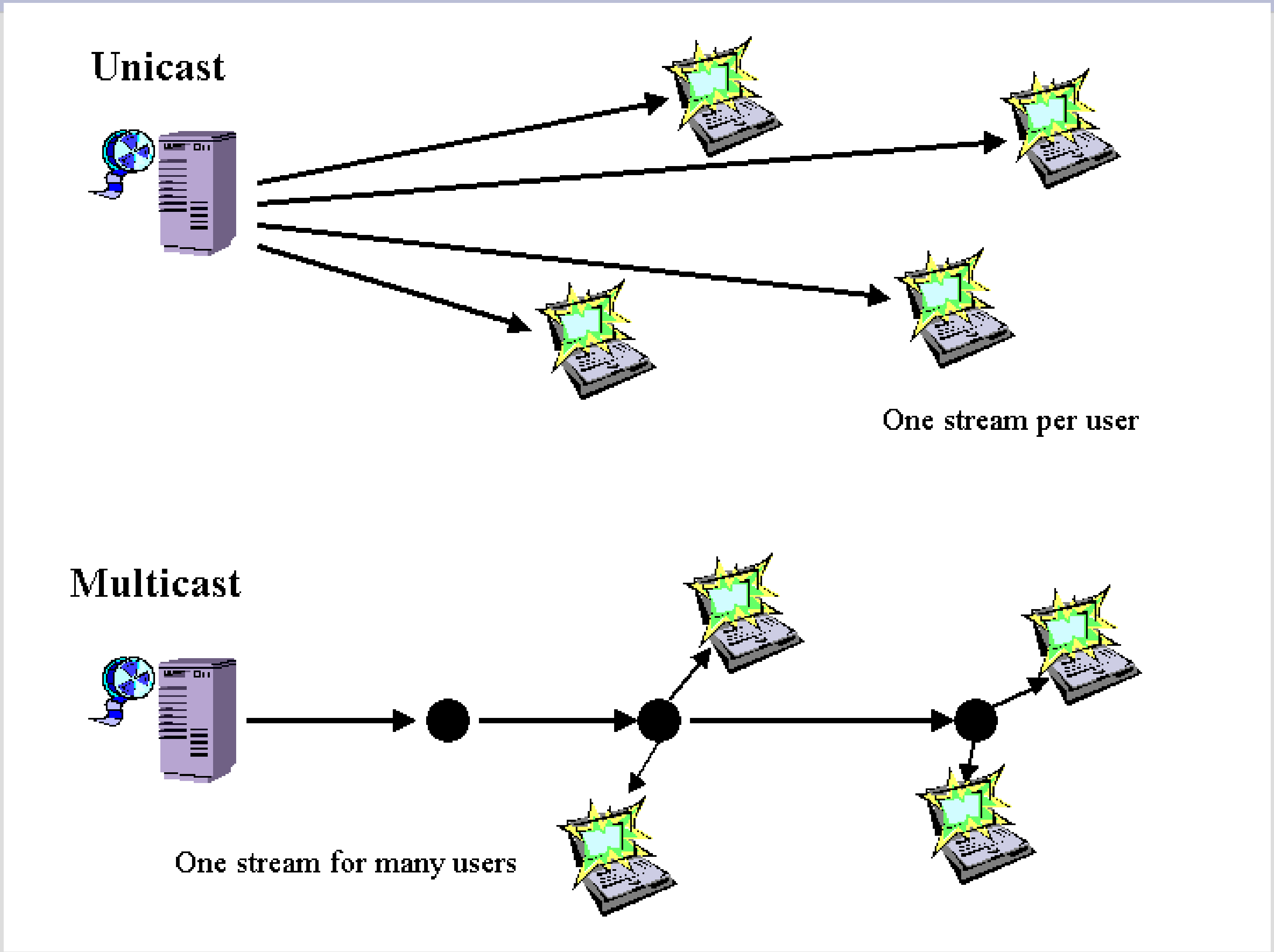
The challenge

- We don't understand *why* streaming audio/video quality is bad
  - Why do webcasts fail even when they have enough resources to handle the traffic load?
- We don't understand *when* streaming audio/video quality is bad
  - How do flash crowds, network congestion, and other conditions affect the quality of a media stream?
- We currently don't have a {good, fast, accurate, comprehensive} way to analyze the user-perceived quality of audio and video streams

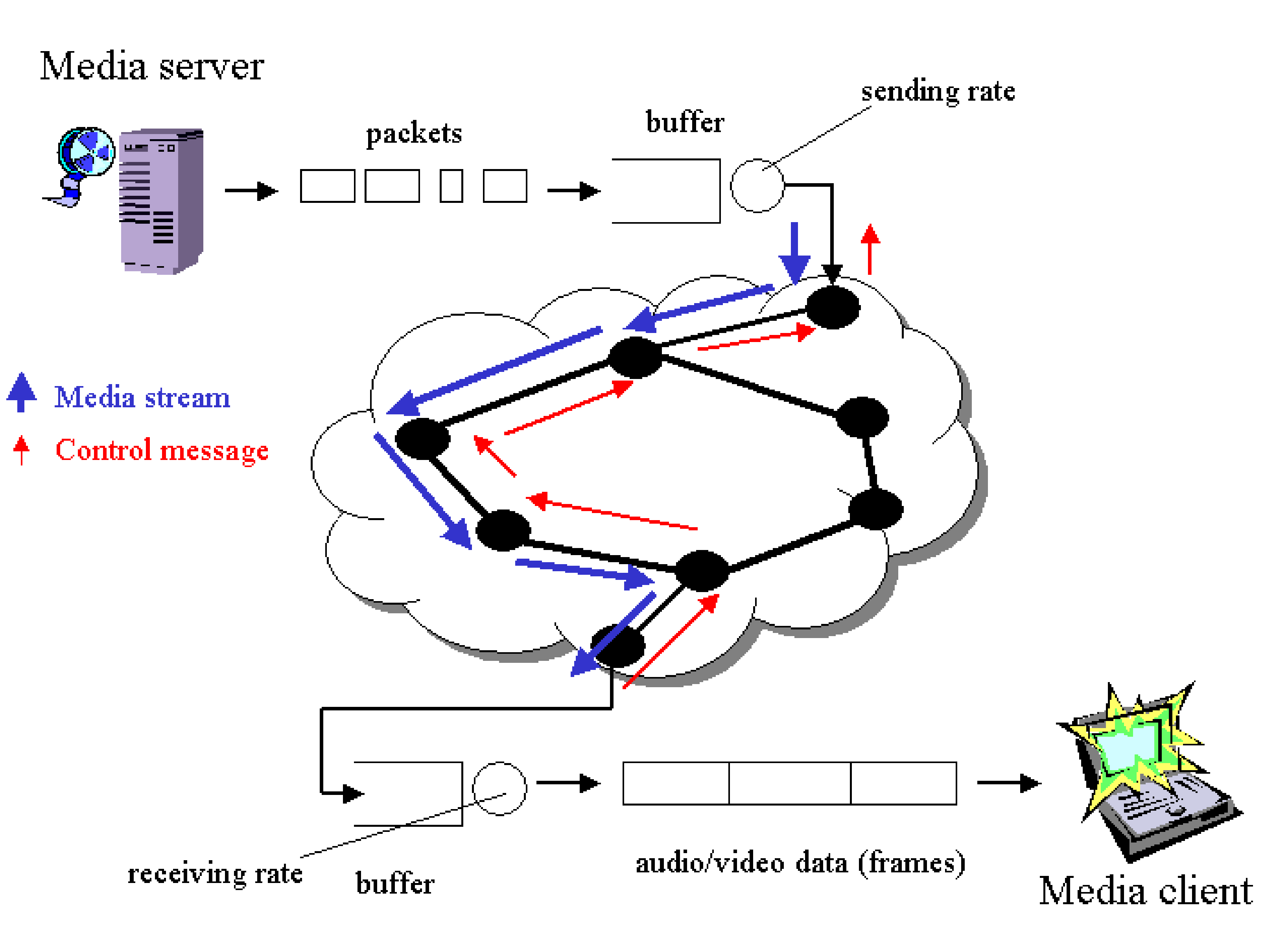
Streaming vs. download

- Streaming: server sends video, you watch as it's being sent
  - espn.com, cnn.com, ...
- Download: server sends entire file first, then you watch it
  - homestarrunner.com, “This Land” parody, ...

Mechanisms for streaming audio and video



A typical audio/video stream



What is “quality”?

- In this context, “quality” means the perception a user has of how good or bad an audio/video stream is

Why do we care about streaming video (and audio) quality?

- Future applications require higher resolution and higher "bandwidth" (more data to send = bigger "pipes" needed to send it)
  - entertainment
  - education
  - telemedicine

Examples of video encoding rates

Source	Bandwidth (Mbps)
Digital video camera (raw)	35
High Definition TV	20
DVD	10
Standard TV	5
DivX	1
Cable modem	[0.25, 3.0]
DSL	[0.26, 1.5]
Dial-up	0.06

Some causes of diminished stream quality

- Lost packets
- Server/router failures
- Packets arriving out of order
- Packets arriving late
- Fragmentation

A real-world fragmentation problem

- Experiments: 5-15% packet loss on network
- Measured packet loss (at media player): 10-40%! (roughly 2-3.5 times greater)
- Cause:
  - media encoder produced packets of 1000 – 4000 bytes
  - network can support 1480 byte packets max
  - network had to break encoded packets into smaller packets (fragments) to “fit” on the network
  - lose 1 fragment --> entire packet is lost

Assessing streaming audio/video quality

- Two main approaches:
  - subjective analysis
  - objective analysis
- Related approaches:
  - control channel
  - commercial software

Subjective assessment

- Idea: user knows best!
- Mean opinion score (MOS)
  - ranking on a scale of 1 (bad) to 5 (perfect)
- +: we know exactly what the user thought of the stream
- -:
  - not scalable
  - one number is not enough information

Objective assessment

- Idea: take measurements
  - network (packet loss, delay, throughput)
  - server logs
- +:
  - easy to measure
  - easy to collect
  - easy to understand
- -:
  - *what* should we be measuring?
  - questionable accuracy

Player feedback (“control channel”)

- RealPlayer, Windows Media Player, RTP protocol
- Idea: send data back to server as the stream is playing (missing data, optimal rate, etc.)
- +: real-time feedback while stream is played
- -: info only goes back to the server

Commercial software tools

- Streamcheck, Chariot, Broadstream
- Synthetic “client” sits “near” customers, collects data, reports sent to various parties
- +:
  - gather data “near” clients
  - software mimics users
- -:
  - no real-time feedback
  - “near” is a relative term

Our goals

- Figure out *why* and *when* the user-perceived quality of a media stream is “bad”
  - “bad” = bad enough so that the user will go away or stop watching the stream
  - do this *without asking the user directly*!
- Use this information to *predict* when the quality of a media stream will deteriorate

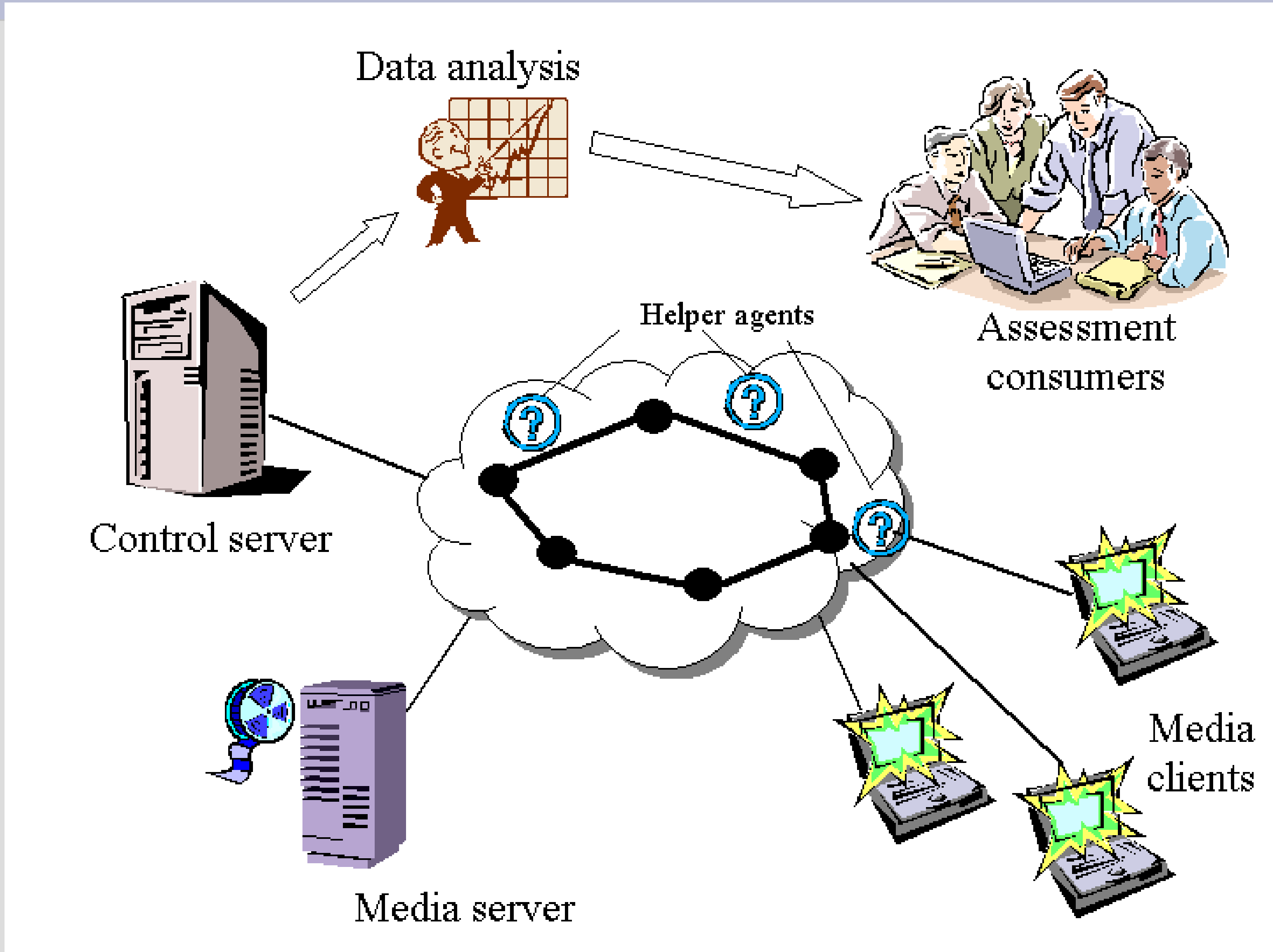
Our approach

- Use measurements that *mimic* the user's experience with the stream
  - take measurements *directly from the media player*
  - combine these with our knowledge of network conditions (loss, delay) during the stream
    - this will tell us *why* and *when* media quality was bad
- What should we measure? --> hard problem!

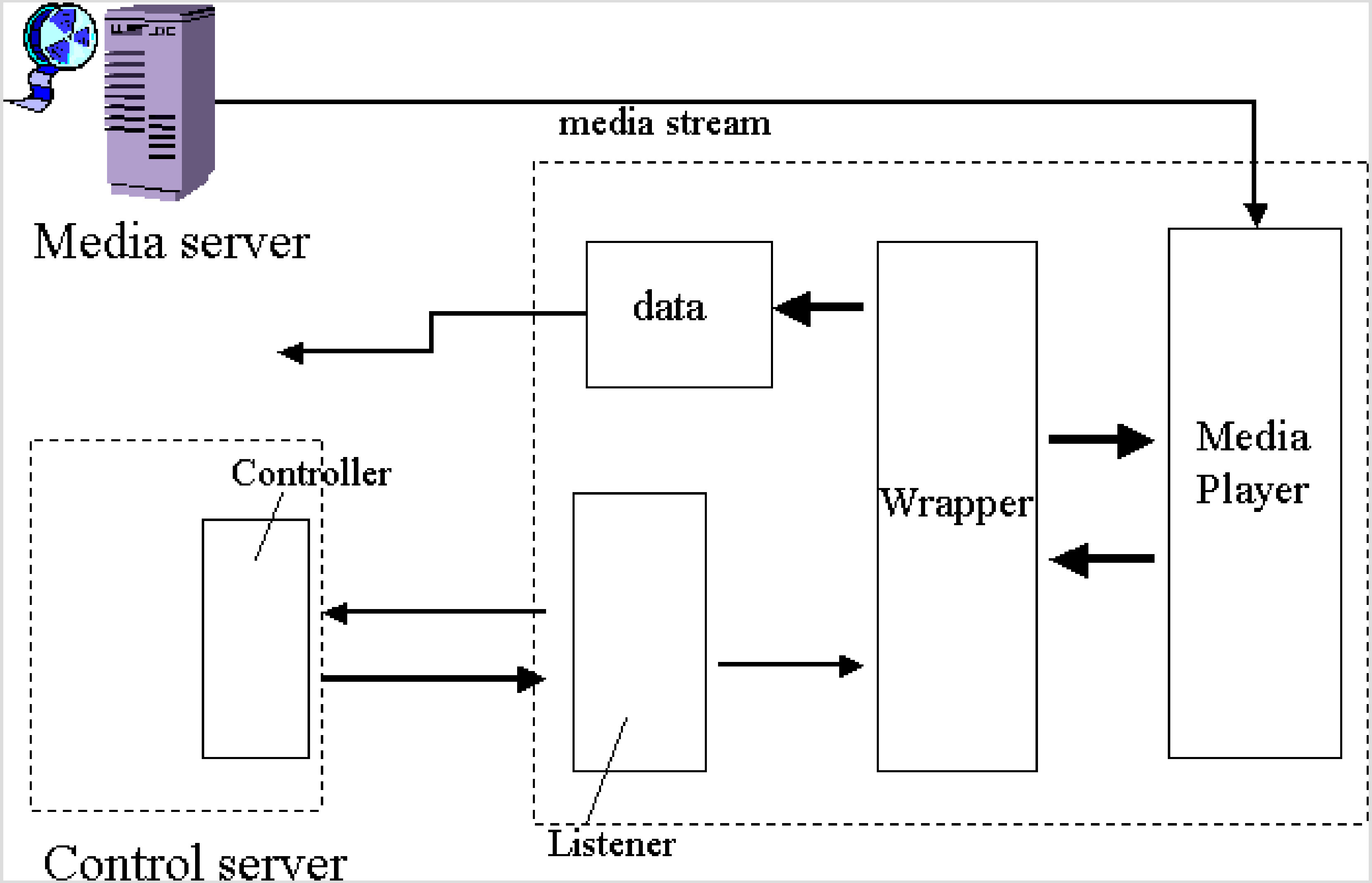
Measurements of interest

- When does the player encounter *buffer starvation*? For how long?
- Are there periods of time over which no new packets arrive at the player? When do they occur, and how long are they?
  - “observing the packet arrival process”
- When are packets lost and/or retransmitted?

System architecture



The framework



Development notes

- Version 1:
  - Windows Media Player
  - ActiveX hooks
  - Java/C#

- Version 2:
  - xine (open-source media player for Linux)
  - modifications to protocol-specific “plugin”
  - C

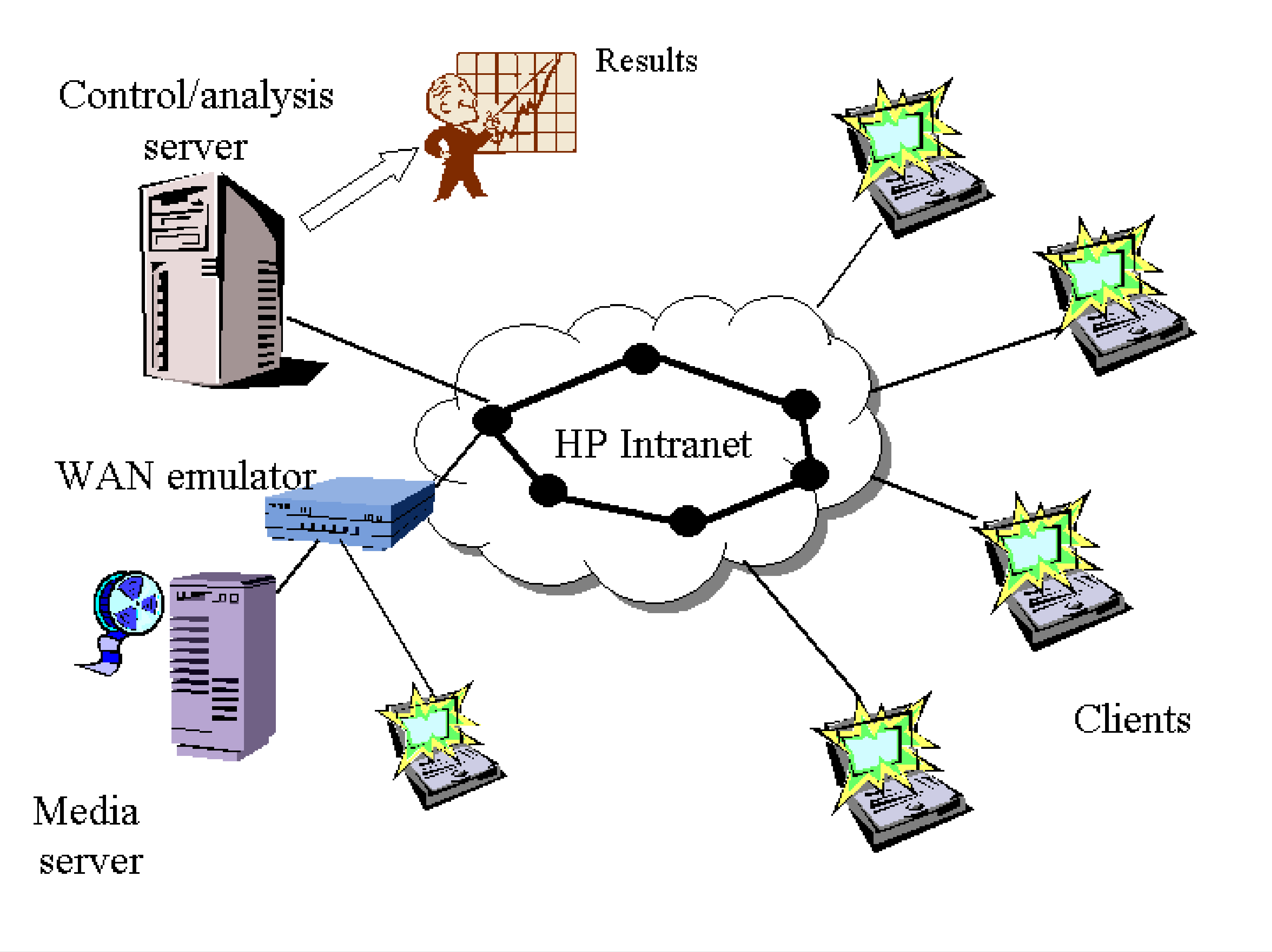
Operational modes

- Test mode
  - remotely-controlled measurements
  - no users, all automated
- User mode
  - measurements from everyday user activity
  - automated data collection
- Differentiates this work from existing solutions!

Validation

- First step: Analytical validation
  - send streams under controlled (known) network conditions
  - match trends in player-derived data with network “snapshots”
- Second step: Subjective validation
  - do these objective measurements accurately reflect the user's view of stream quality?

Analytical validation: network testbed



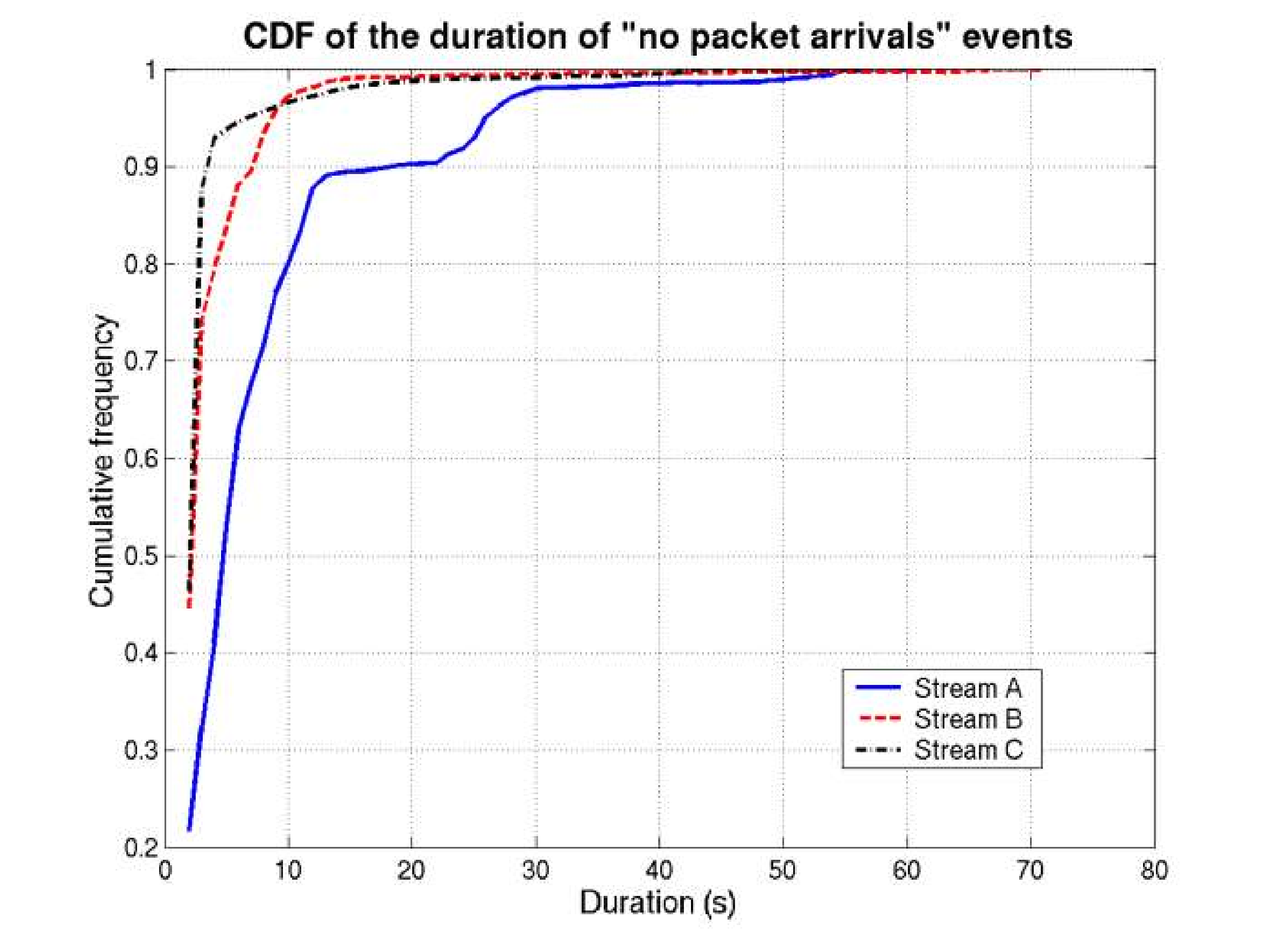
Experiments

- Randomly drop between 1% and 25% of packets from the media server to the clients
- Different types of media streams
  - animated movie trailer (~2:30), > 250 kbps
  - commercial (0:30), ~200 kbps
  - news clip (4:30), ~150 kbps
  - CEO speech (11:30), ~100 kbps
  - technical presentation (30:00), ~84 kbps
- 2001-2004

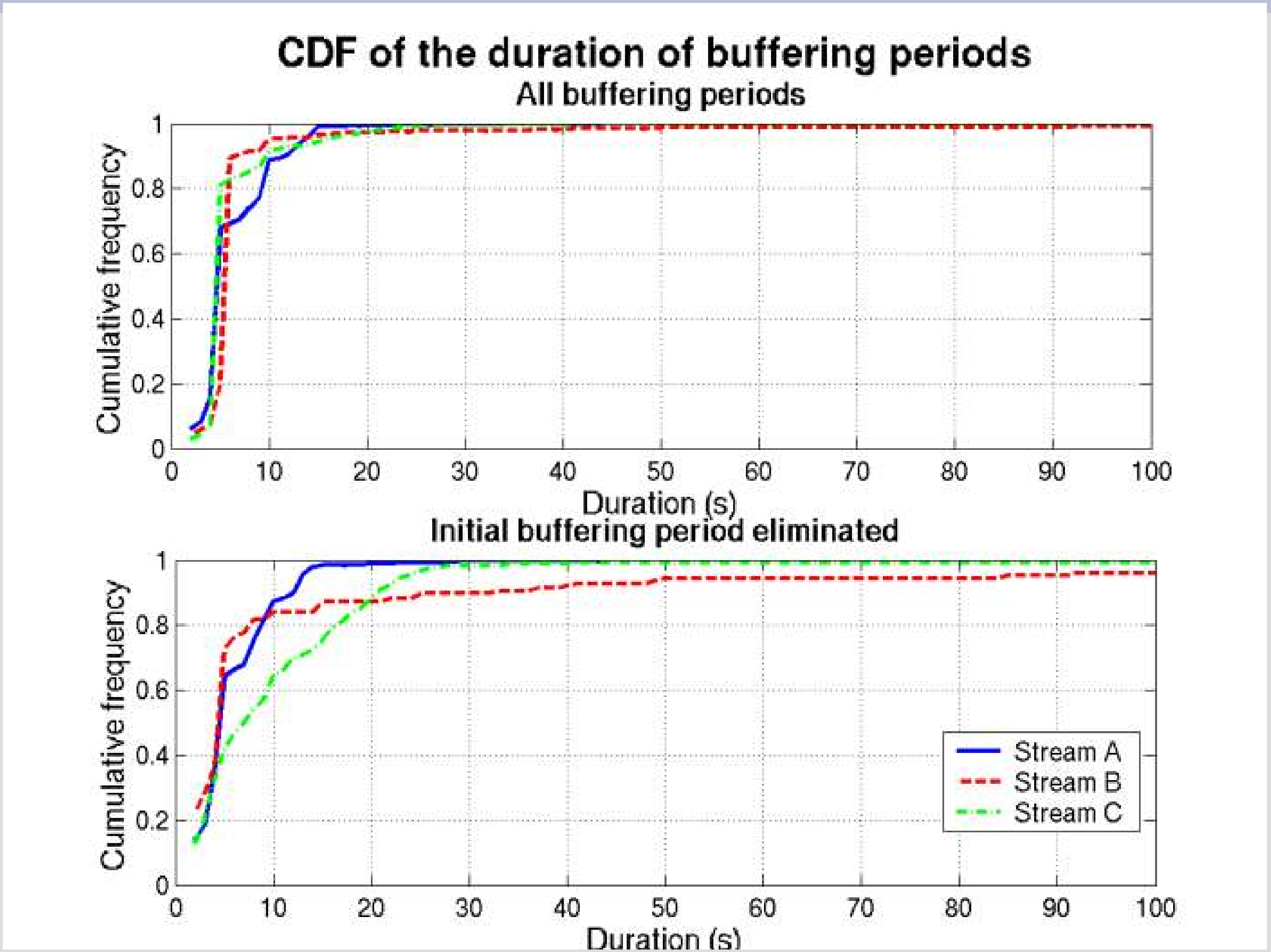
Results

- Buffer starvation: when does it occur, and how long does it last?
- Periods of no packet arrivals
- Predicting future quality degradation of a stream

Duration of “no packet arrival” periods



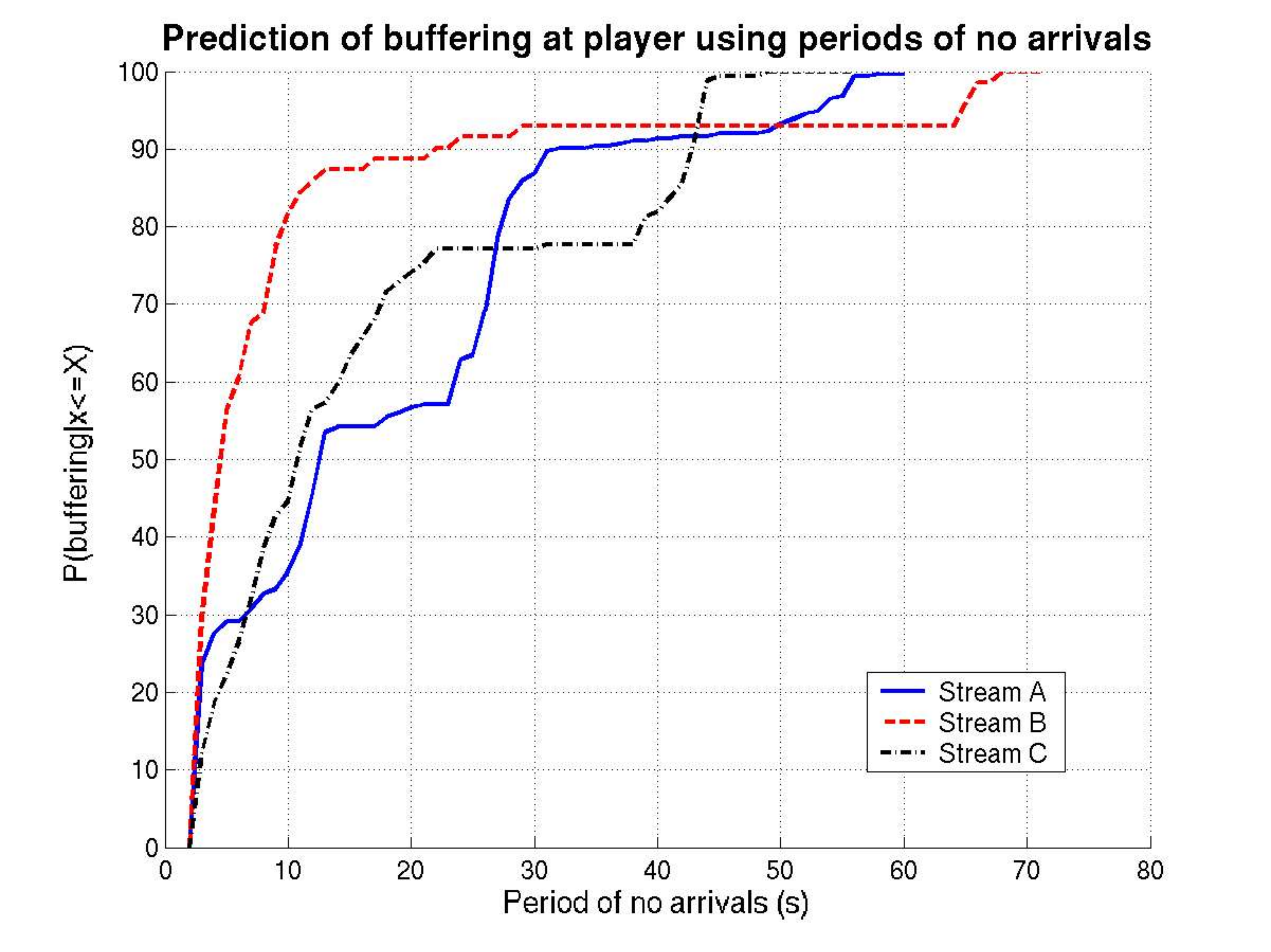
Buffer starvation periods



Predicting future quality degradation

- Idea: can we use client-side measurements to *predict* when the quality of a media stream will degrade?
- Examples
  - many retransmissions = lost packets = video “artifacts”, jerkiness; missing audio
  - no packets arrive = buffer starvation = “freeze frame” video and loss of audio
- Can we predetermine when buffer starvation will occur by observing the packet arrival process?

Predicting buffer starvation from the packet arrival process



What could we do if we could predict degraded stream quality?

- Play out locally-stored content (ads, previews, trailers, etc.)
- Pre-cache content and play out locally when conditions are bad
- Route around the congestion
  - serve content from another server
  - find another network route
- If all else fails, tell the user to come back later

Subjective quality validation

- Show the same test streams to users and ask them to rate them
- 7-point scale
  - catches some nuances not available in 5-point MOS
- User comments too
- Preliminary data (summer 2004): Strong-ish correlation between reported ranking and network loss rate

Ongoing/future work

- Subjective validation*
- Continuing work on xine port*
- Expand architecture to QuickTime, MPEG streaming
- Explore the prediction aspect in more detail*
  - lag time between packet reception pause and onset of buffering
- Automated analysis, “pruning” of data

* with students

Summary

- Understanding the quality of media streams is very important as the amount of media content on the Internet (and on local/campus networks) increases
- I have developed an architecture, tool, and mechanism for measuring received quality of a media stream at the client side
  - using existing media players and infrastructure
  - without interfering with the normal operation of the media stream
  - without requiring the active participation of the end user