Slide 0

# Bandwidth Smoothing of Compressed Pre-Recorded Video

**Jennifer Rexford**
**Networking and Distributed Systems**
**AT&T Labs Research, Murray Hill**
**jrex@research.att.com**

**Joint work with Professor Wu-chi Feng at Ohio State University**

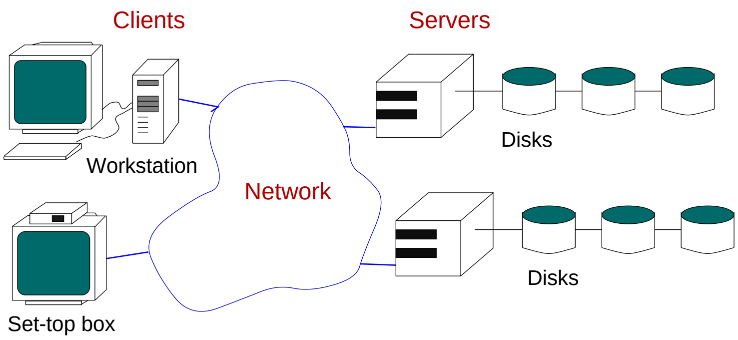
Slide 1

## Outline

- Motivation for smoothing pre-recorded video
- Bandwidth smoothing algorithms
- Performance metrics and motion-JPEG video library
- Simulation comparison of algorithms and video traces
- Conclusions and ongoing work

Slide 2

## Video-on-Demand Service




- Servers with pre-recorded video on large, fast disks
- Network, perhaps with rate or delay guarantees
- Client sites with buffer space for video playback

Slide 3

## Pre-Recorded Video

Emerging applications:

- Distance learning
- Movies, advertisements
- Virtual reality

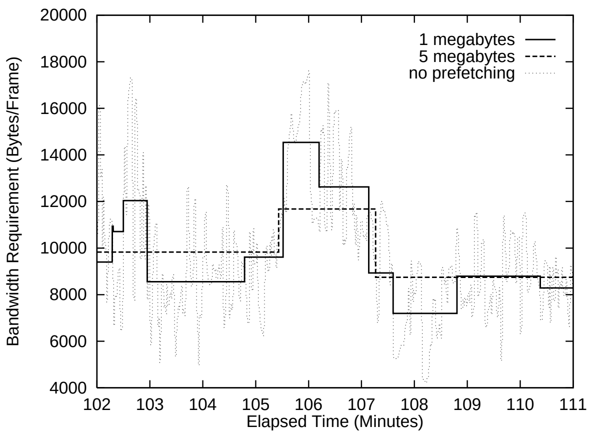
Networking challenges:

- Huge storage and bandwidth requirements for video
- Compression reduces size but generates bursty traffic
- Burstiness complicates server & network provisioning

$\Rightarrow$ reduce burstiness by prefetching video frames into the client playback buffer

Slide 5

## Smoothing the Savage Beast




Transmitting **Jurassic Park** with 1 and 5 megabyte prefetch buffers at the client site

Slide 4

## Motion-JPEG Video Library

| | Size (GB) | Time (min) | Rate (Mbps) |
|---|---|---|---|
| *Beauty and the Beast* | 1.82 | 80 | 3.04 |
| *Big* | 2.26 | 102 | 2.96 |
| *Crocodile Dundee* | 1.82 | 94 | 2.59 |
| *E.T. (quality 75)* | 1.24 | 110 | 1.51 |
| *E.T. (quality 90)* | 1.78 | 110 | 2.17 |
| *E.T. (quality 100)* | 3.11 | 110 | 3.78 |
| *Home Alone 2* | 2.35 | 115 | 2.73 |
| *Honey, I Blew Up...* | 2.12 | 85 | 3.32 |
| *Hot Shots 2* | 1.92 | 84 | 3.06 |
| *Jurassic Park* | 2.50 | 122 | 2.73 |
| *Junior* | 2.71 | 107 | 3.36 |
| *NCAA Final Four* | 1.21 | 41 | 3.95 |
| *Rookie of the Year* | 2.22 | 99 | 2.98 |
| *Seminar1* | 0.98 | 63 | 2.07 |
| *Seminar2* | 1.08 | 68 | 2.12 |
| *Seminar3* | 0.88 | 52 | 2.26 |
| *Sister Act* | 2.06 | 96 | 2.86 |
| *Sleepless in Seattle* | 1.72 | 101 | 2.28 |
| *Speed* | 2.46 | 110 | 2.97 |
| *Total Recall* | 2.34 | 109 | 2.88 |

**Slide 6**

## Bandwidth Smoothing Algorithms

Given the parameters:

- Frame sizes $\{f_i\}$ for $n$ frames
- Client buffer size $b$

And the constraints at the client buffer:

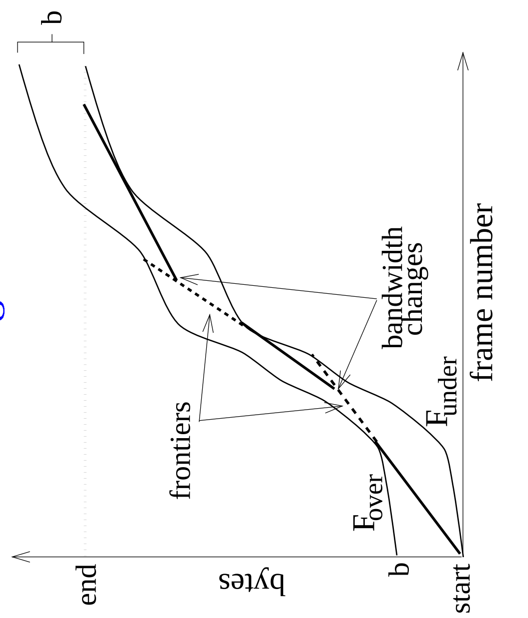
- Avoid underflow $\mathrm{F}_{under}(k) = \sum_{i=0}^{k} f_i$
- Avoid overflow $\mathrm{F}_{over}(k) = b + \sum_{i=0}^{k} f_i$

Find a server transmission schedule consisting of:

- Number of time intervals $m \leq n$
- Transmission of video at rate $r_j$ during interval $j$
- Time intervals of length $t_j$, in units of frame slots

**Slide 7**

## Making Plans




Algorithms differ in

- Where to start interval $j$
- What slope $r_j$ to use

$\Rightarrow$ different server transmission plans

Slide 8

## Existing Smoothing Algorithms

- Critical bandwidth allocation [Feng95]
  - Minimum peak bandwidth
  - Minimum number of bandwidth increases
- Minimum changes bandwidth allocation [Feng95a]
  - Minimum peak bandwidth
  - Minimum number of bandwidth changes
- Minimum variability bandwidth allocation [Salehi96]
  - Minimum peak bandwidth
  - Minimum variability of bandwidth allocations
- Piecewise constant rate transmission [McManus96]
  - Bandwidth changes at periodic intervals

Slide 9

## Server and Network Resources

Reservation models for server and network

- Reserve peak bandwidth for the entire transfer
- Renegotiate bandwidth reservation during the transfer
- Booking all bandwidth changes in advance

Frequency of network rate changes

- Renegotiate whenever transmission rate changes
- Balance renegotiations with the cost of bandwidth
- Renegotiate only on rate *decreases* to avoid failure

**Slide 10**

## Performance Metrics

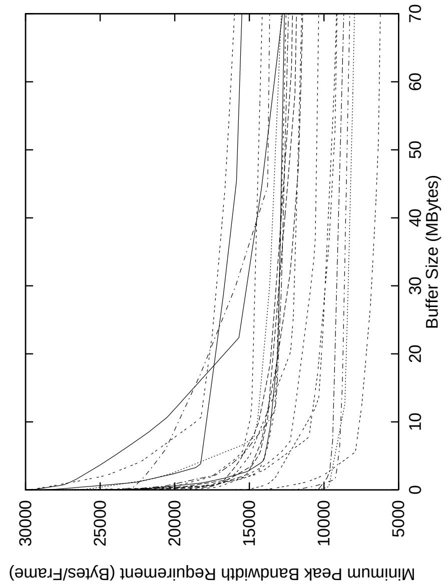
For server transmission plan $\{(r_j, t_j) | j = 0, 1, ..., m-1\}$:

- Peak bandwidth: $\max_j \{r_j\}$
  $\Rightarrow$ minimize server & network resource requirements
- Number of bandwidth changes: $m$
  $\Rightarrow$ minimize control operations and size of plans
- Variability of bandwidth: $\mathrm{stdev}(c_0, c_1, \ldots, c_{n-1})$
  $\Rightarrow$ maximize smoothness & reduce size of rate changes
- Variability of time intervals: $\mathrm{stdev}(t_0, t_1, \ldots, t_{m-1})$
  $\Rightarrow$ reduce worst-case rate of control operations

where the server transmits at rate $c_i = r_j$ if frame slot $i$ occurs during rate interval $j$.

**Slide 11**

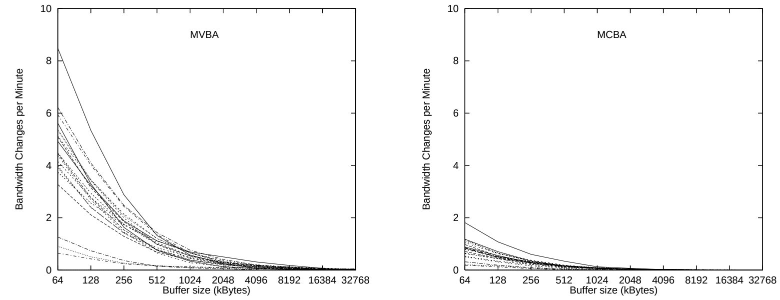
## Peak Rate vs. Buffer Size




- Top curves: *Beauty and the Beast*, *E.T. (quality 100)*, and *Jurassic Park*
- Bottom curves: *Seminar* videos for small buffers; *E.T. (quality 75)* for large buffers
- "Knee" of the curve: varies across the video clips, depending on time-scale(s) of burstiness

Slide 12

## Rate Changes vs. Buffer Size

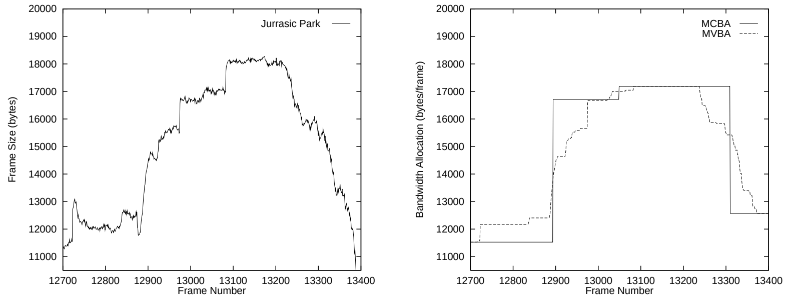



(a) Minimum variability (b) Minimum changes

- Top curve: *NCAA Final Four* (large frames; long-term variation in scene content)
- Bottom curves: *Seminar* videos (small frames; low variability)

Slide 13

## Gradual Changes in Frame Sizes




(a) Segment of *Jurassic Park* (b) MCBA and MVBA plans

⇒ 104 rate changes (55 increases and 49 decreases) for the min-variability plan vs. 3 rate changes for the min-changes plan, under a 128-kilobyte client prefetch buffer

Slide 14

## Summary of Results

### Smoothing algorithms:

- All four algorithms are effective at reducing the peak rate
- All four algorithms have similar bandwidth variability
- MCBA algorithm has 1/3—1/8 the frequency of rate changes
- Minimizing the peak rate results in some short rate intervals

### Video clips:

- Diminishing returns for larger buffer sizes ("knee of the curve")
- Effects of time-scales of burstiness and variability of frame sizes
- Difficulty of handling gradual increases/decreases in frame sizes
- Fairly wide variation in performance metrics across videos

Slide 15

### Conclusions:

- Importance of smoothing of pre-recorded, compressed video
- Collection of smoothing algorithms and performance metrics
- Evaluation across a wide range of full-length video traces

⇒ benefits of extending service boundary to client playback buffer

### Ongoing work:

- Combining bandwidth smoothing with VCR functionality
- Coordinating the sharing of client playback space
- Characterizing relationship between peak rate and buffer size

⇒ new smoothing algorithms that weigh the competing costs of server transmission, network transfer, and client playback