

Packet Loss Recovery for Streaming Video

N. Feamster and H. Balakrishnan
MIT

In Workshop on Packet Video (PV)
Pittsburg, April 2002



Introduction (1)

- Streaming is growing
- Commercial streaming successful (ie RealPlayer and MediaPlayer)
 - but proprietary and inflexible
 - Use MPEG-4 since open
- Current streaming inflexible
 - Suboptimal
 - Want to adapt to current network
 - Present system that adapts to loss



Introduction (2)

- MPEG video under loss suffers from propagation of errors (what is this)
 - Fundamental tradeoff between bandwidth efficiency and error resilience
- Current FEC approaches effective but
 - Reduces benefits of compression
 - Tough to get adaptation right
- Some say cannot use retransmission for streaming but
 - Selective retransmission (I-frames) ok
- Build model + system
 - (Also TCP-Friendly using CM, but not focus)



Outline

- Introduction (done)
- Model (next)
- System Architecture
- Performance Evaluation
- Related Work
- Conclusions



Model (Outline)

- Problem Description (next)
 - MPEG-4
 - Error Propagation
- Packet loss model
 - Experiments
 - Analytic Model
- Benefits of Selective Repair



Problem Description

- MPEG-4 has three frame types: I, B, P
 - Note, MPEG-4 calls them "Video Object Planes" but frames is fine
- While high compression, suffers from error propagation
 - Loss of I-frame packets can affect subsequent frames, too



Loss in an I-frame

PSNR 21.996

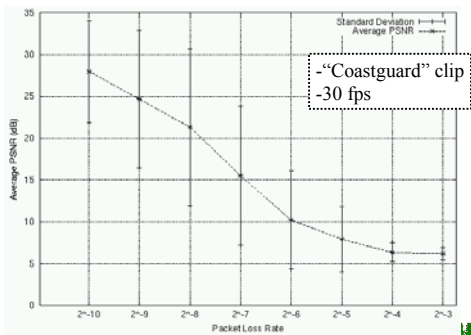


Propagation to Next P-frame

PSNR 17.538



Average PSNR versus Loss Rate



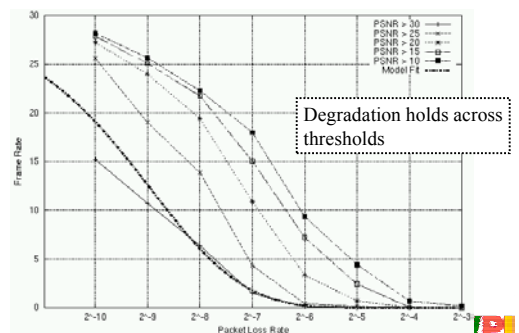
Model (Outline)

- Problem Description (done)
 - MPEG-4
 - Error Propagation
- Packet loss model (next)
 - Experiments
 - Analytic Model
- Benefits of Selective Repair

Packet Loss Model

- Assume packet loss degrades quality
 - True, in general, unless FEC
- Assume below PSNR threshold, would discard
 - But PSNR doesn't model perceived quality
 - This viewability threshold varies with picture
 - + So will analyze several thresholds
 - + Also, can use other quality metrics
 - Generally, under 20 db is bad
 - + Loss of 2^8 (about .4%) trouble and needs correction
 - + (See previous figure)

Effects of Loss on Frame Rate (with Thresholds)



Analytic Model (1)

- Observed frame rate $f = f_0(1-\phi)$
 - ϕ is fraction of frames dropped
 - f_0 is original frame rate (ie- 30 fps)

$$\phi = \sum_i P(f_i) \cdot P(\bar{F}|f_i)$$

- Where i runs over types I, P and B
- $P(f_i)$ can be determined by fraction in stream
- $\bar{F}$ is event that a frame is useless (PSNR below threshold)
- f_i is event that frame is of type i



Analytic Model (2)

$$P(\bar{F}|I) = 1 - (1-p)^{S_I}$$

- p is packet loss rate
- S_i is size of I frame (similar for S_B , S_P , too)
- Assume if any packet lost, frame useless

$$P(\bar{F}|P) = \frac{1}{N_P} \sum_{k=1}^{N_P} \left(1 - (1-p)^{S_I + k S_P}\right)$$

$$P(\bar{F}|B) \leq \frac{1}{N_P} \sum_{k=1}^{N_P} \left(1 - (1-p)^{S_I + (k+1)S_P + S_B}\right)$$

P needs all previous P (and I) frames
 N_P is number of P frames in GOP



Analytic Model (3)

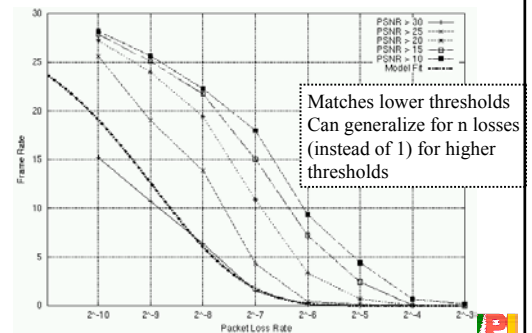
$$P(\bar{F}|P) = 1 - \frac{(1-p)^{S_I}}{N_P (1 - (1-p)^{S_P})} \left(1 - (1-p)^{S_P N_P}\right)$$

$$P(\bar{F}|B) \leq 1 - \frac{(1-p)^{S_I + S_P + S_B}}{N_P (1 - (1-p)^{S_P})} \left(1 - (1-p)^{S_P N_P}\right)$$

- Simplify to closed form above
- Now, using equations and given
 - N_P , S_I , S_B , S_P , f_0
- Can determine
 - $f = f_0(1-\phi)$
- "Model". Compare to measured



Model vs. Measured

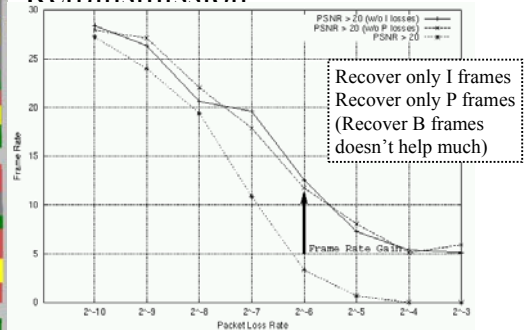


Model (Outline)

- Problem Description (done)
 - MPEG-4
 - Error Propagation
- Packet loss model (done)
 - Experiments
 - Analytic Model
- Benefits of Selective Repair (next)



Benefits of Selective Retransmission



Outline

- Introduction (done)
- Model (done)
- System Architecture (next)
- Performance Evaluation
- Related Work
- Conclusions

- 
- WPI

```
graph LR
    CM[CM] -- "loss control/ RTT" --> MS[MPEG Server]
    MS -- "callbacks" --> CM
    MS -- "data" --> I((Internet))
    I -- "loss RTT/request" --> MS
    I -- "RTP" --> MC[MPEG Client]
    MC -- "RTSP" --> I
    MC -- "data" --> RTCP[RTP/RTCP]
    RTCP -- "loss RTT/request" --> MC
```

Overview of System

- CM provides TCP-Friendly data rate
 - Calls back when can send data
- Data sent over RTP (using UDP)
- Control over RTSP (over TCP)
- Frames put into Application Data Units (ADU)
 - 1 per ADU



Packet Header

0				1				2				3			
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5
V	P	RC	Payload Type-RR				Length								
SSRC of packet sender (Receiver ID)															
SSRC. I (SSRC of first source)															
fraction lost				cumulative number of lost packets											
extended highest sequence number received															
Interarrival Jitter															
Timestamp Echo (LSR)															
Processing Time (DLSP)															
Window Size (kB)								Padding							
ADU Sequence Number															
ADU Fragment Length (bytes)															
ADU Offset (bytes)															

- P used for priority (I-frames)
- Sequence numbers (for loss)
- Total length
- Frames can be more than one packet
- Offset for location in GOP



WPI

Loss Detection and Recovery

- Mid-frame
 - Gaps in ADU using offset plus fragment length
- Start-of frame
 - First offset non-zero
- End-of-frame
 - ADU less than reported length
- Complete loss
 - Detected by gap in ADU sequence numbers
- Can use priorities to decide upon retransmissions
 - (Me: which ones *is* the hard part!)



WORCESTER POLYTECHNIC INSTITUTE

- 

Implementation

- Used OpenDivX for MPEG-4
- Used CM previously built for Linux
- Used RTSP client-server library
- Also, extended `mplayer` for Linux
 - Call-backs give complete ADU to player
 - Retransmit all unless canceled by app

The diagram illustrates the implementation architecture and data flow across two layers:

- Application Layer:** Contains "Encoded Frames" represented as gray rectangles.
- SR-RTP Layer:** Contains a "Reassembly Buffer" represented as a long horizontal bar divided into segments.

Data flow and functions shown:

- An arrow labeled `rtp_read()` points from the Reassembly Buffer up to the Application Layer.
- An arrow labeled `rtp_write()` points from the Application Layer down to the Reassembly Buffer.
- An arrow labeled `rtp_send_to_client()` points from the Reassembly Buffer up to the Application Layer.
- A callout box labeled "ADU with missing packets" points to a specific segment within the Reassembly Buffer.
- An arrow labeled "Complete ADUs in Reassembly Buffer" points to the entire Reassembly Buffer.

In the bottom right corner, there are logos for "UNIVERSITY OF ALABAMA AT BIRMINGHAM" and "JPL".

- 
- www.jointprojectinitiative.org

Postprocessing (Receiver-Based)

- May still have some missing frames
- Simple replacement bad if motion
 - Estimate motion and compensate

The diagram illustrates a receiver-based postprocessing technique for video frames. It shows the flow from a 'P-frame' to an 'I-frame' and back to a 'P-frame'.

Top Section: Data Flow

- Damaged I-frame**: An input to the process.
- Preceding P-frame**: An input to the process.
- Error locations**: An input to the process.
- Motion Vector Estimation**: An input to the process.
- Estimate Appropriate Pixels to Replace**: The core processing step.
- Replace MBs from previous P-frame**: The output of the estimation step.
- Reconstructed I-frame**: The final output of the process.

Bottom Section: Visual Representation

- P-Frame**: A large rectangle containing a small, shaded, irregular shape representing a motion vector or error location.
- I-Frame**: A large rectangle containing a larger, shaded, irregular shape representing a motion vector or error location.
- Arrows**: Two curved arrows connect the shaded shapes in the P-frame to the shaded shape in the I-frame, indicating motion compensation.

-

Outline

- Introduction (done)
- Model (done)
- System Architecture (done)
- Performance Evaluation (next)
- Related Work
- Conclusions

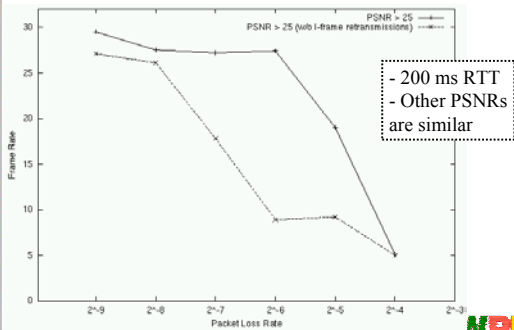


Setup

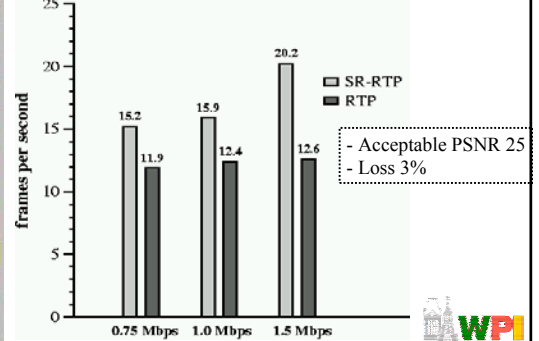
- Server on P4, 1.5 GHz, Linux 2.2.18
- Client on P2, 233 MHz, Linux 2.2.9
- 1.5 Mbps, 50 ms latency, 3% loss
 - using Dummynet, a WAN emulator
- 20 Kbps video at 30 fps
 - Used only 300 frames
- For “Internet”, used 200 ms RTT and used Web cross traffic with SURG (traffic emulator)
- Added buffering to combat jitter
 - (Me: how and how much is not specified)



Benefits of Selective Reliability (1)



Benefits of Selective Reliability (2)

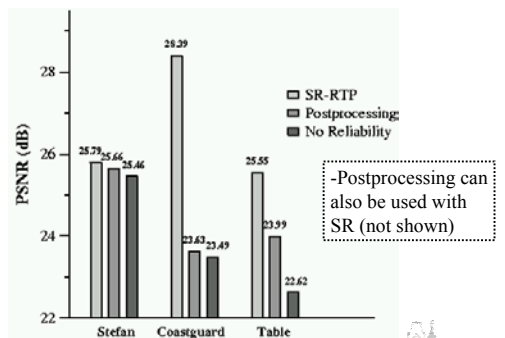


Buffering Requirements

- Buffer for jitter depends upon variance
- Buffer for retrans depends upon RTT
- Buffer for quality adaptation (congestion responsiveness) depends upon data rate (R)
 - $O(R)$ for SQRT (TFRC)
 - $O(R^2)$ for AIMD (TCP)
- Dominant factor depends upon RTT to RTT variance and rate
- (Me: no more analysis than above)



Benefits Receiver Postprocessing



Outline

- Introduction (done)
- Model (done)
- System Architecture (done)
- Performance Evaluation (done)
- Related Work (next)
- Conclusions



Related Work (1)

- Media Transport
 - RTSP for control [49]
 - RTP is application level protocol with real-time data properties (timing info + sequence) [48]
 - RTCP protocol provides reports to sender [40]
 - TFRC [52], CM, RAP[44]
 - + all TCP-Friendly protocols for streaming media



Related Work (2)

- Error and Loss Recovery
 - Survey of techniques [54]
 - Receiver post-processing [22]
 - Avoid propagation but don't delay [54]
 - Effects of MPEG-4's built in repair for bit errors [23]
 - FEC schemes [9,10,33]
 - Priority-based packets [39,50,1]



Conclusion

- Streaming video must account for loss
- This paper models loss to explain effects
- Can use retransmission of important packets for significant gain
- Describe system to do so based on extensions of common tools



Future Work?



Future Work (Me)

- Wider-range of videos
 - Motion content
 - GOPs
 - Resolution
- Alternate measures of quality
- Evaluation of buffering

