Measurement and Performance Study of PERT for On-demand Video Streaming

Bin Qian    A.L.Narasimha Reddy

Department of ECE
Texas A&M University

PFLDNeT 2010
Outline

1. Introduction
2. Background
3. Experiment
   - NS2 Simulation
   - Linux Test
4. Summary
Motivation

- Current TCP is not suitable for video streaming applications.
- In the Internet, many other services (HTTP, FTP, P2P) compete for bandwidth.
**Related Work . . .**

**Boyden et al, 2007**

- TCP can function adequately with a 1.5 higher bandwidth than required stream rate in unconstrained streaming.

**Wang et al, 2008**

- TCP generally provides good streaming performance when the achievable TCP throughput is roughly twice the media bitrate, with only a few seconds of startup delay.
Problem

How well can TCP support streaming, when $T/\mu \leq 2.0$?

- $T$ is the achievable TCP throughput.
- $\mu$ is the video playback bitrate.
PERT = Probabilistic Early Response TCP

Sumitha et al, 2007
- explored the performance of PERT in homogeneous environment.

Kiran et al, 2008
- made PERT adaptive to heterogeneous environments.
Probabilistic Early Response

PERT learns about network congestion by measuring delay
Window Adjustment Mechanism ...

Aggressive Window Increasing

- $W = W + \alpha$
- $\alpha \geq 1$
Window Adjustment Mechanism ...

3 modes

\[ T_{compete} = 0.65 \times \text{maximum queuing delay} \]

- When \( T < T_{\text{min}} \), high-speed mode
- When \( T > T_{\text{compete}} \), TCP-compete mode
- When \( T_{\text{min}} < T < T_{\text{compete}} \), safe mode
Window Adjustment Mechanism ...

**High-speed mode**

\[ \alpha = \alpha_{max} = 32 \]

**TCP-compete mode**

\[ \alpha = 1 + \frac{p'}{p} \]
- \( p' \) is the early response probability
- \( p \) is the congestion loss probability

**Safe mode**

\[ \alpha = \alpha_{min} = 1 \]
**Window Adjustment Mechanism**

**Conservative Window Decreasing**

- \( W = W \times (1 - \beta) \)
- \( \beta = \frac{q'}{q' + q} \)
  - \( q' \) is the estimated queuing delay
  - \( q \) is the maximum queuing delay
- so \( W \geq W/2 \)
Queuing Behavior

PERT enqueues more packet earlier and less later ...
NS2 Simulation

Setup

- 20 - 35 PERT/RENO/CUBIC CBR Senders
- 20 - 35 RENO FTP Senders
- 300 RENO HTTP Senders
- Bottleneck link
  - 25 Mbps
  - 5 - 10 ms
  - Droptail queue
  - 150 packets
- Access links
  - 10 Mbps
  - 15 - 30 ms
- CBR Recvers
- FTP Recvers
- HTTP Recvers
**Parameters Exploration**

**T/u vs. CBR Streams Number**

- **Y-axis**: T/u
- **X-axis**: CBR Streams Number
- Values: 21, 23, 26, 30, 34

**Bandwidth vs. CBR Streams Number**

- **Y-axis**: Bandwidth (Mbits)
- **X-axis**: CBR Streams Number
- Values: 21, 23, 26, 30, 34
- **Categories**: CBR, FTP, HTTP
Performance Metric

- CBR stream is successful if fraction of late packets < $10^{-4}$
- Video streaming quality is evaluated by fraction of successful CBR streams
In low range [1.0-1.4], it drops drastically as $T/\mu$ decreases. In high range [1.4-2.0], it changes slightly as $T/\mu$ increases.
PERT > RENO and CUBIC in $T/\mu$ range [1.0 - 1.4]

Simulation Results ...
Simulation Results...

PERT > RENO & PERT ≈ CUBIC in T/µ range [1.4 - 1.8]
PERT > RENO and CUBIC in loss rate range [0.02 - 0.06]
Test Bed

Bandwidth 15 Mbps
Delay 45 ms
Buffer 500 Kb
Avatar 1080p
HTTP streaming
PERT helps to reduce the playback glitches

<table>
<thead>
<tr>
<th>TCP Variants</th>
<th>PERT</th>
<th>RENO</th>
<th>CUBIC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Late Picture Skipping #</td>
<td>5.5</td>
<td>33.5</td>
<td>30.5</td>
</tr>
<tr>
<td>Audio Output Starving #</td>
<td>3.0</td>
<td>11.0</td>
<td>7.5</td>
</tr>
</tbody>
</table>
PERT responses early before packet loss. PERT adjusts the window smoothly.
Conclusions

- PERT and CUBIC push $T/\mu$ constraint to roughly 1.4.
- PERT $> \text{RENO}$, over all $T/\mu$s, loss rates and start-up delays.
- PERT $> \text{CUBIC}$, over low $T/\mu$s, high loss rates and strict start-up delays constraints.
Future Work

- Carry out more evaluations and comparisons against other protocols.
Thank You!
Probabilistic Early Response Parameters

The parameters are currently fixed, and can be chosen adaptively:

- $T_{min} = 5\, ms$
- $T_{max} = 10\, ms$
- $P_{max} = 0.05$
Steady state throughput equations:

\[
\frac{\beta_{\text{PERT}}(p + p' - p \times p')}{\alpha_{\text{PERT}}} = \frac{\beta_{\text{TCP}} \times p}{\alpha_{\text{TCP}}}
\]

- \(\alpha_{\text{TCP}} = 1\)
- \(\beta_{\text{PERT}} = \beta_{\text{TCP}}\)
- So \(\alpha_{\text{PERT}} = p + p' - p \times p'/p \approx 1 + p'/p\)