Bandwidth Modeling for Network-Aware Applications

Jürg Bolliger,
Thomas Gross, Urs Hengartner
ETH Zürich
Network-awareness

- Web delivery: Unpredictable response times
  - heterogeneity in network access
  - bandwidth fluctuations

- Solution: Network-aware delivery
  dynamic adaptation of content (quality) to available bandwidth
  - e.g., to meet a user-specified time limit

- Need to know bandwidth
  - during transfer
  - ideally, estimate of future bandwidth
Bandwidth estimation is important

- Network-aware applications require timely and accurate information

- Bandwidth estimation is not trivial
  » No explicit feedback from receiver at application level
  » Protocol layering makes it hard to figure out what’s going on at the lower layers

- Lower layers have relevant information
Key points

● Bandwidth estimation via simple widening of the transport API

● Enhancements to TCP make a difference
Experiment: Untimely feedback

- Trace 60-seconds transfer in Internet

- Use trace modulation [Noble et al., SIGCOMM ‘97] to emulate bandwidth on LAN

- Network-aware delivery of 90 images within 60 seconds
  » Overall reduction of 1.5 MB (30%) required to meet time limit
Bandwidth: Zürich (CH) - Linz (AUT)
Experiment: Untimely feedback

- Run experiment with different delays for bandwidth estimates
  » Delay of $n$ RTTs: network-aware sender learns about available bandwidth only $n$ RTTs later
Impact of untimely information

- Untimely $\Rightarrow$ inaccurate
- Inaccurate $\Rightarrow$ suboptimal adaptation and loss of performance
Bandwidth estimation: Requirements

● Two issues
  » Dynamic assessment of current bandwidth (modeling)
  » Prediction of bandwidth expected in future

● Bandwidth estimates must be cheap to obtain
  » The fewer parameters required, the better
TCP throughput models

- TCP specification:
  \[ BW \sim M_1(cwnd, \text{rtt}) \]

- Mathis et al. [CCR ‘97]:
  \[ BW \sim M_2(mss, \text{rtt}, \text{loss-rate}) \]

- Padhye et al. [SIGCOMM ‘98]:
  \[ BW \sim M_3(\ldots, \#\text{timeouts}, \emptyset \text{ timeout period}) \]

- M2/3 model asymptotic (steady-state) TCP behavior
  \[ \Rightarrow \text{model only loss-afflicted connections} \]
Internet experiment

- User-level, TCP-based transport protocol
  - Reno: Standard TCP flow & congestion control
  - Several TCP enhancements, e.g.,
    - SACK: Selective acknowledgments
    - FACK: Forward acknowledgement congestion control
      [Mathis et al., SIGCOMM ‘96]

- 6-month Internet experiment
- 25’000 connections (1MB transfers)
- 12 sites involved in Europe (7) & North America (5)
Simple models work fairly well

- **Connections with zero loss:**
  
  \[
  \begin{array}{l|c}
  \text{R}^2 \text{ [coefficient of determination]} & \text{Reno} \\
  \hline
  \text{M1 (cwnd/rtt)} & 0.99 \\
  \text{M2 (Mathis et al.)} & — \\
  \text{M3 (Padhye et al.)} & — \\
  \end{array}
  \]

- **Loss-afflicted connections:**
  
  \[
  \begin{array}{l|c}
  \text{R}^2 & \text{Reno} \\
  \hline
  \text{M1 (cwnd/rtt)} & 0.62 \\
  \text{M2 (Mathis et al.)} & 0.83 \\
  \text{M3 (Padhye et al.)} & 0.92 \\
  \end{array}
  \]
Comparison: model errors (Reno)

M3 is accurate, even in high loss situations (timeouts)

$M1$ (cwnd/rtt) $\quad M3$ (Padhye et al.)
Duration of connection

Model error increases *only slightly* as connections (and hence steady-state phases) get shorter

\[ M3 \ (Padhye \ et \ al.) \]
More sophisticated protocols?

- **Timeouts** are problematic
  - Performance
  - Bandwidth modeling (+ prediction)

- Timeout causes (for Reno)
  - Multiple packet loss 40.1%
  - Non-trigger of recovery 39.5%
  - Lost retransmissions 20.4%

- Protocol enhancements to avoid timeouts
  - e.g., SACK TCP, FACK, NetReno, ...
SACKs are effective

- Effect on timeouts: 48% reduction

- Effect on throughput: 44% improvement

![Diagram showing the effect of SACKs on timeouts and throughput]
## Effect on throughput models

- **Connections with zero loss:**

<table>
<thead>
<tr>
<th>R²</th>
<th>Reno</th>
<th>FACK</th>
</tr>
</thead>
<tbody>
<tr>
<td>M1 (cwnd/rtt)</td>
<td>0.99</td>
<td>0.99</td>
</tr>
<tr>
<td>M2 (Mathis et al.)</td>
<td>——</td>
<td>——</td>
</tr>
<tr>
<td>M3 (Padhye et al.)</td>
<td>——</td>
<td>——</td>
</tr>
</tbody>
</table>

- **Loss-afflicted connections:**

<table>
<thead>
<tr>
<th>R²</th>
<th>Reno</th>
<th>FACK</th>
</tr>
</thead>
<tbody>
<tr>
<td>M1 (cwnd/rtt)</td>
<td>0.62</td>
<td>0.97</td>
</tr>
<tr>
<td>M2 (Mathis et al.)</td>
<td>0.83</td>
<td>0.91</td>
</tr>
<tr>
<td>M3 (Padhye et al.)</td>
<td>0.92</td>
<td>0.93</td>
</tr>
</tbody>
</table>

- **FACK bandwidth fluctuates less**
Comparison: model errors (FACK)

- Differences between models less pronounced
  - Reason: FACK’s effectiveness in avoiding timeouts
  - But: Behavior unchanged for high loss rates

\[ M1 \text{ (cwnd/rtt)} \quad M3 \text{ (Padhye et al.)} \]

![Graph comparing M1 and M3 model errors](image)
Conclusions

- Effectiveness of adaptation strongly depends on the timeliness and accuracy of the bandwidth estimation

- Fairly accurate bandwidth estimation seems possible
  - Simple models capable of providing timely information

- The transport protocol plays an important role

- Simple widening of the transport API may provide the information required by network-aware applications