Reflections on Congestion Control

Praveen Balasubramanian pravb@microsoft.com
State of the Union

• Congestion control defaults in major device platforms
  • Android, ChromeOS: CUBIC
  • iOS, macOS: CUBIC
  • Windows: CTCP

• Datacenter congestion control
  • DCTCP, DCQCN and variants
  • BBR, Timely

• Academia proposals
  • PCC (performance oriented congestion control)
  • Sprout, Remy

• Trends
  • AQM
  • QUIC, User mode transports
  • IaaS
Bufferbloat - RTT inflation during peak load times

Average RTT Over Time of Day Adjusted For Time Zone

<table>
<thead>
<tr>
<th>Day</th>
<th>Median</th>
<th>Percentile90</th>
<th>Percentile95</th>
<th>Percentile99</th>
</tr>
</thead>
<tbody>
<tr>
<td>DayOfWeek</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Friday</td>
<td>47</td>
<td>270</td>
<td>390</td>
<td>1034</td>
</tr>
</tbody>
</table>

Average SRTT information from sampled TCP connections for Windows Desktop and Xbox consoles
Food for thought

• With a multitude of congestion control algorithms sharing bottleneck links, how do we ensure fairness and good performance?
  • How do we deal with user mode transports which may accelerate the variety of congestion control algorithms?
  • Is the onus on each individual developer to test all the various permutations? Will this scale?

• How can we prevent congestion control from becoming an arms race?
  • Can we define a minimum spec for all congestion control to adhere to in the presence and absence of explicit congestion signaling?
  • What does “TCP friendly” mean now?

• What is the role of the IETF here?
  • Is standardization or establishing best practices possible?
  • Is publishing informational RFCs for individual algorithms sufficient?