DCTCP recap

• Datacenter TCP
• Informational draft RFC in progress: draft-ietf-tcpm-dctcp-01
• Goals
  – Low latency for short flows
  – High throughput for long flows
  – Solve incast
• Approach
  – Use ECN to quantify the extent of congestion
  – Requires modification of the TCP sender and receiver
  – Requires configuration of switches
• Currently recommended to be deployed only in single administrative domain like datacenter
• Cannot co-exist with traditional TCP sharing the same bottleneck link(s)
DCTCP as L4S scalable congestion control?

• Good starting point for a scalable congestion control for L4S
  – Multiple existing implementations
    – Windows Server
    – Linux mainline
    – FreeBSD
    – Royalty-Free, Reasonable and Non-Discriminatory License to All Implementers

• Experiments show DCTCP
  – can work over large RTT
  – delivers low latency high throughput

• For safe incremental deployment over the Internet
  • Dual Q Coupled AQM
  • TCP Prague
Evolving DCTCP to TCP Prague

*Subset of requirements for incremental deployment on the Internet. Marcelo’s talk will cover the exhaustive list.
*Most of these proposals are applicable and required independent of L4S.

<table>
<thead>
<tr>
<th>Title and description</th>
<th>Reference</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>L4S Identifier: IP level L4S identifier for AQM</td>
<td>draft-briscoe-tsvwg-ecn-l4s-id</td>
<td>ECT(1) seems acceptable</td>
</tr>
<tr>
<td>Suitable ECN feedback and negotiation</td>
<td>draft-ietf-tcpm-accurate-ecn and draft-stewart-tsvwg-sctpecn</td>
<td>“essential” part of Accurate ECN (minus new TCP option) seems most viable.</td>
</tr>
<tr>
<td>DCTCP Fall-back to classic TCP on loss</td>
<td>draft-ietf-tcpm-dctcp-01</td>
<td>Already implemented in Windows. Recommended in the implementation issues section of the DCTCP draft RFC.</td>
</tr>
<tr>
<td>Sub-MSS congestion window</td>
<td>draft-bagnulo-tcpm-tcp-low-rtt</td>
<td>Required for safety</td>
</tr>
<tr>
<td>ECT marking of TCP control packets</td>
<td>draft-bagnulo-tsvwg-generalized-ecn</td>
<td>Explicit negotiation will allow for deviations from 3168 on both sender and receiver as needed.</td>
</tr>
<tr>
<td>Faster than additive increase</td>
<td>None</td>
<td>Combine with CUBIC or Compound? IMO can be left up to the CC implementation</td>
</tr>
</tbody>
</table>
Experimentation enablement for L4S

• Windows Server 2012, 2012 R2 and 2016
  • Negotiate ECN by default for outbound connections
  • Complete ECN negotiation for inbound connections
  • If ECN negotiation succeeds and RTT during SYN handshake < 10 msec, use DCTCP
  • Can be forced globally by running as admin: “netsh int tcp set supplemental template=datacenter”

• Windows 8, 8.1 and 10
  • Undocumented socket option - programmatically choose DCTCP for a given TCP connection
    int optval = 3; // DCTCP
    int optlen = sizeof(optval);
    int rc = setsockopt(socket_handle, IPPROTO_TCP, 12, (const char *)&optval, optlen); // 12 = socket option number
  • Not officially supported, so please don’t use in production!

• Microsoft is very interested in the L4S effort