HPCC++: Enhanced High Precision Congestion Control

draft-miao-ccwg-hpcc

draft-miao-ccwg-hpcc-info

Rui Miao, Rong Pan, Jeongkeun Lee, Barak Gafni, Yuval Shpigelman, Jeff Tantsura, Guy Caspary, Surendra Anubolu, Allister Alemania, Jiaqi Gao

IETF-117 ccwg

July 2023
Cloud desires hyper-speed networking

Today, clouds have

- Storage-compute separation is norm
- HDD→SSD→NVMe
- Higher-throughput, lower latency
- 1M IOPS / 50~100us

High-performance storage

High-performance computation

- Distributed deep learning, HPC
- CPU→GPU, FPGA, ASIC
- Faster compute, lower latency
- E.g. latency <10us

Resource disaggregation

- More network load
- Need ultra-low latency: 3-5us, >40Gbps (Gao Et.al. OSDI’16)
Hyper-speed network chips to form hyper-speed networking

Hardware-offloading (e.g., RDMA)
Traditional software-based networking stacks cannot keep with the speed

Real-time Congestion Control (CC)
Lots of data and communication => more pressure on the network
Challenges in some CC suites in high-speed networks

• Convergence upon congestion
• Running multiple applications over converged network
  ➢ Queues and buffers are scarce resources
• Parameter tuning
In-band Telemetry

• New networking ASICs have in-band telemetry capabilities
• Packets can collect telemetry on their route
• Various efforts to define inband telemetry
  • IETF IOAM
  • INT/P4.org
  • IFA
Can we use inband telemetry as more precise/richer feedback for congestion control?
In-band telemetry format

• HPCC++ defines the algorithm of using telemetry information
  • including queue length, transmitted bytes, timestamp, link capacity, etc.
  • draft-miao-ccwg-hpcc

• Yet, packet format is up to the environment
  • draft-miao-ccwg-hpcc-info provides examples of different telemetry encodings

<table>
<thead>
<tr>
<th>bits</th>
<th>31-24</th>
<th>23-16</th>
<th>15-8</th>
<th>7-0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Device-ID</td>
<td>PT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>TID</td>
<td>congestion</td>
<td>Tx Bytes Cnt[39:32]</td>
<td>TTL</td>
</tr>
<tr>
<td>2</td>
<td>Rx Timestamp Sec - Upper</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Rx Timestamp Sec</td>
<td>Rx Timestamp Nano Upper</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Rx Timestamp Nano</td>
<td>Tx Timestamp Nano Upper</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Tx Timestamp Nano</td>
<td>Egress Queue Cell Cnt</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Src-Sys-Port</td>
<td>Dest-Sys-port</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Tx Bytes Cnt[31:0]</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Example format of in-band telemetry used by HPCC++
HPCC++ Addresses all the discussed challenges

Using in-band telemetry as the precise feedback enables

• Faster convergence
  ➢ Sender knows the precise rate to adjust to

• Near-zero queue
  ➢ Feedback does not only rely on queue

• Fewer parameters
  ➢ Rich and precise feedback, reduces heuristics which requires more parameters
So, What HPCC++ Actually Is?

• **It is a service**
• This service can be utilized by a given transport
• This service can also be utilized by a routing engine
Additional work

• Multi-queue considerations
• Consider additional receiver feedback
• Extend on encoding examples
Your Feedback is Appreciated!
Thank You