CM Mechanisms: NIC perspective

Manoj Wadekar (Intel)
Pat Thaler (Broadcom)
Agenda

- NIC design impact of CM mechanism
- Comparison of proposed solutions
- Management challenges
- Summary Recommendations
NIC CN impact: Transmit

Allow practical cost-performance tradeoff
NIC CN impact: Receive

Rx Mappers and Parsers for offloads (e.g. IP checksum)

Tags can add complexity in Receive side processing
NIC: CM protocol impact

- **CM packet consumption:**
  - Need to respond to CM messages received by reducing RL rate on Tx-side
  - Needs real time handling
  - Less signaling is better

- **Reflection in NICs:**
  - NICs (typically) do not forward packets from Rx to Tx
  - Reflection of tags requires forwarding fragment of Rx packet to Tx
  - If reflection is required, packet to packet is preferred over tag to packet
  - Protocol that needs NICs to perform reflection will add burden

- **Per RL state in Transmit:**
  - E.g. rate recovery, drift require timer/packet based state machine per RL
  - Simpler the better
  - Should balance performance-complexity trade-off
2-point CM proposals

- Poll during July Plenary meeting provided TF preference for 2-point CM proposals
  - 3-point may provide some optimization but adds too much complexity
  - 3-point has issues in presence of multi-path and flow coalescing
- Currently BCN and QCN are 2-point CM proposals
- Both mechanisms have many similarities
  - CP provides congestion information to RPs via CN messages
    - Based on Q and Q-derivative
  - RP’s algorithmically adapt their transmission rate to match capacity
    - Distributed congestion management vs. rate allocation by CP
  - Response to congestion event is similar for both mechanisms
  - Similar configuration parameters for both mechanisms
- Proposals differ primarily in their rate recovery mechanisms


Comparison of QCN and BCN proposals

<table>
<thead>
<tr>
<th>Feature</th>
<th>BCN</th>
<th>QCN</th>
<th>RP Impact</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fb computation point</td>
<td>At Reaction Point (RP)</td>
<td>At Congestion Point (CP)</td>
<td>QCN Simplifies RP</td>
</tr>
<tr>
<td>Feedback parameters sent</td>
<td>Qoff, Qdelta (16b, 16b)</td>
<td>Quantized Fb (6b)</td>
<td>QCN Simplifies RP</td>
</tr>
<tr>
<td>Sampling Probability</td>
<td>Fixed w/ Jitter, over-sampling</td>
<td>Fb-based sampling (1% to 10%)</td>
<td>QCN Increases signaling during congestion</td>
</tr>
<tr>
<td>Positive feedback</td>
<td>Yes, Additive increase proportional to Fb</td>
<td>No</td>
<td>QCN Lowers signaling</td>
</tr>
<tr>
<td>RP-CP association in RL-Tag</td>
<td>Yes, as required for positive feedback</td>
<td>No</td>
<td>QCN simplifies RP</td>
</tr>
<tr>
<td>Trigger to self-recovery</td>
<td>Time-based drift</td>
<td>Packet-based &amp; Time-based</td>
<td>QCN adds complexity to RP</td>
</tr>
<tr>
<td>Emergency feedback</td>
<td>BCN0, BCN(Max)</td>
<td>No (Fb-based sampling, Fb w/ better range)</td>
<td>BCN(0), BCN(Max) Valuable if it lowers signaling</td>
</tr>
</tbody>
</table>

QCN provides complexity-performance trade-off
Comparison of QCN and BCN parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>BCN</th>
<th>QCN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Qeq</td>
<td>CP</td>
<td>CP</td>
</tr>
<tr>
<td>W</td>
<td>RP</td>
<td>CP</td>
</tr>
<tr>
<td>Gd</td>
<td>RP</td>
<td>RP</td>
</tr>
<tr>
<td>Rmin</td>
<td>RP</td>
<td>RP</td>
</tr>
<tr>
<td>driftFactor</td>
<td>RP</td>
<td>RP</td>
</tr>
<tr>
<td>driftTimer</td>
<td>RP</td>
<td>RP</td>
</tr>
<tr>
<td>Gi</td>
<td>RP</td>
<td>toThreshold</td>
</tr>
<tr>
<td>samplingInterval</td>
<td>CP</td>
<td>baseProbability</td>
</tr>
<tr>
<td>Qsc</td>
<td>CP</td>
<td>extraFastRecovery</td>
</tr>
<tr>
<td>Qmc</td>
<td>CP</td>
<td>fastRecoveryThreshold</td>
</tr>
<tr>
<td>Qsat</td>
<td>CP</td>
<td>hyperactiveIncrease</td>
</tr>
<tr>
<td>Rd</td>
<td>RP</td>
<td>minDecFactor</td>
</tr>
<tr>
<td>BCN0</td>
<td>CP</td>
<td>EfrMax</td>
</tr>
<tr>
<td>A (additive increase)</td>
<td>RP</td>
<td></td>
</tr>
</tbody>
</table>

6 parameters at CP and 7 at RP | 3 parameters at CP and 11 at RP

Simplified default parameter template is necessary
Simplified Parameter Template

• Identify constants and variables from parameter list

• Reduce number of variables: primary/derived
  – E.g. Qsc, Qmc derived from Qeq, Rmin derived from C etc.

• Reduce number of (experimental) options

• Define default values for majority of variables:
  – That work for large control loop delay up to 500 uS (with certain acceptance criteria)
  – That work for large number of RPs: say, 1000
  – That allow reasonable and practical implementations

• Minimize deployment-specific variables:
  – E.g. Is W for large control loop delay is different than small CLD?

• Number of End Stations is (typically) larger than number of switches in Data Center
  – Avoid End Station tuning as much as possible
Summary Recommendations (from NIC perspective)

- QCN (2-point) provides good-enough solution
  - Leverages BCN fundamentals
  - Improves implementation by reducing signaling, avoiding tagging

- Simplicity of solutions should be maintained
  - Adopt framework and reduce low ROI options

- More attention should be provided now to reduce parameters
  - Validation/Management can be a challenge to CN deployment
  - Discovery protocol may allow switches to distribute configuration parameters to End Stations