IPv6 Options for Cyclic Queuing and Forwarding Variants

draft-yizhou-detnet-ipv6-options-for-cqf-variant-00

Yizhou Li (Presenter)
Shoushou Ren
Guangpeng Li
Fan Yang

Jeong-dong Ryoo
Peng Liu

July, 2022
Fundamental CQF has attractive “simplicity” features for wider deployments

- 2-buffer per port. Input and output swap once every cycle interval $T_c$.
- E2e time taken:
  Min: $(h-1) T_c + DT^*$
  Max: $(h+1) T_c$
  $^*$: $DT = \text{dead time (revisit later)}$. Very small in fundamental CQF
- Attractive “simplicity” features:
  - Simple calculable latency bound: only relevant to $T_c$ and $h$, $\approx h*T_c$
  - Simple maintenance: no per-stream per-hop state maintenance
CQF has potentials for wider deployments - 1

Wider deployment requires supporting one or combination of the followings:
- Smaller e2e latency bound (1)
- Larger number of hops (2)
- Longer links (3)
- Larger processing time variance as node type diversity increases (4)

Recall that CQF latency bound $\approx h \times T_c$

Higher speed link provides the potential to reduce $T_c$, even with greater value of $h$
- allow at least one 1500B/max size packet to be sent within $T_c$
- With increasing of link speed, the same amount of data can be transmitted within a smaller cycle time
- Counteract larger $h$

Potentials for item (1) and (2), next page for item (3) & (4)

<table>
<thead>
<tr>
<th>Cycle Time ((\mu s))</th>
<th>Buffer Size per Cycle (Byte)</th>
<th>Link bandwidth</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>100Mbps</td>
<td>1Gbps</td>
</tr>
<tr>
<td>1</td>
<td>12.5</td>
<td>125</td>
</tr>
<tr>
<td>1.2</td>
<td>15</td>
<td>150</td>
</tr>
<tr>
<td>2</td>
<td>25</td>
<td>250</td>
</tr>
<tr>
<td>4</td>
<td>50</td>
<td>500</td>
</tr>
<tr>
<td>10</td>
<td>125</td>
<td>1250</td>
</tr>
<tr>
<td>12</td>
<td>150</td>
<td>1500</td>
</tr>
<tr>
<td>120</td>
<td>1500</td>
<td>15000</td>
</tr>
</tbody>
</table>

Cycle time decreasing:
100x \(\mu s\) -> 10x \(\mu s\) -> few \(\mu s\)
Fundamental CQF support req (3) & (4) but with low utilization

- Revisit DT (dead time): the last byte sent by node A in cycle (i-1) has to be ready for sending at node B before the start of cycle i.

- DT is at least: max propagation delay + max processing delay at the next node + max other time variations.

- The longer the propagation or processing delay, the larger the DT.

- DT eats up cycle interval \( T_c \) when \( T_c \) is small (both values < 1ms): result in low utilization or impractical in extreme case (consider prop delay > \( T_c \))

- Hard for fundamental CQF:
  - Shorter \( T_c \) for lower e2e latency bound
  - Larger DT for longer link and/or processing time
  - Smaller ratio of DT/Tc for better utilization
CQF Variant (>2 buffer) has the potential to support (3) & (4)

- 3 buffer works in rotation manner
- A straightforward variant to fundamental 2-buffer CQF:
  - Configuration is similar
  - Can easily deduce from fundamental CQF without the rigid requirement to produce new standard
- More than 3 buffer is required when the receiving time spans over two cycle interval boundaries.
- In general, it is feasible.
A closer look at the CQF variant: a time ambiguity window exists

- Receiving time window swells when the (processing) time variance increases
- Keep DT small
- Time ambiguity window exists for two consecutive cycles
- The larger the time variance and/or the smaller the DT, the larger the ambiguity window
- So setting the time demarcation to differentiate pkts from two consecutive cycles is impractical (see left)

**Way out:** pkt carry cycle id metadata at output to help the downstream node determine the correct buffer to put it in
Summary

• CQF has attractive features and potentials for wider deployments
• CQF variant is a straightforward extension from fundamental CQF:
  • use more than two buffers
  • some extra configurations would be required
  • Other variants may exist
• A missing part in current CQF variant: remove the ambiguity when identifying the packets from the upstream’s two consecutive cycles
• IPv6 options to carry cycle id metadata is proposed.
Solicit feedback

• Is it a good way to address the “ambiguity” issue in order to facilitate the increasing demand to use CQF and its variants in the wider scenarios?

• IPv6 options, whether and/or how to collaborate with other WG (6man)