NS-3 Implementation of ABB CoDel Queue Discipline

Gongbing Hong
Georgia College and State University
Outline

• Quick review of existing queue disciplines in ns-3

• Why another queue discipline?

• ABB CoDel and its implementation in ns-3

• Preliminary result

• Experience and challenges of ABB CoDel implementation in ns-3

• Future work
Review of existing queue disciplines in ns-3

• **Single-queue** queue disciplines (qdisc’s)
  - Non-AQM: Fifo/pfifo-fast
  - **AQM**: RED/CoDel/PIE

• **Multi-queue** qdisc’s for fair queueing
  - Fq-CoDel (by default, 1024 internal queues)
  - Fq-Pie (by default, 1024 internal queues)
Why another queue discipline?

- **Bandwidth allocation fairness** considerations (especially weighted fairness)
  - Single-queue qdisc’s can hardly provide fair bandwidth allocation
  - Fq-CoDel / Fq-Pie does not provide weighted fair bandwidth allocation
- **Low delay**
  - Buffer bloat is a huge problem that has been recognized and dealt with
  - AQM qdisc’s such as CoDel and PIE provide low delay while others do not
- **Implementation complexity**
  - Fair queueing disciplines such as WFQ often use many internal queues – not scalable
  - Work required per packet is high – O(n) or O(log n)
ABB CoDel ideas

• ABB stands for Adaptive Bandwidth Binning
  ▪ Previous implemented in ns-2 and evaluated under DOCSIS 3.x

• ABB approximates fair queueing by:
  ▪ Using only a few (e.g. 3~5) internal queues (called bins) for low implementation complexity
  ▪ Using CoDel to manage queueing delays for each bin
  ▪ Periodically moving flows from one bin to another depending on flow service rates:
    ➢ Each flow can have a flow weight for bandwidth allocation
    ➢ Flows with similar normalized service rates are classified into the same bin
    ➢ Flows with lowest normalized services rates are classified into first bin
    ➢ Flows with highest normalized services rates are classified into last bin
## ABB CoDel implementation in ns-3

<table>
<thead>
<tr>
<th>Fq-CoDel</th>
<th>ABB CoDel</th>
</tr>
</thead>
<tbody>
<tr>
<td>FqCoDelQueueDisc subclasses QueueDisc by implementing following virtual methods:</td>
<td>• <strong>Modeled after Fq-CoDel</strong> by subclassing QueueDisc and QueueDiscClass</td>
</tr>
<tr>
<td>§ DoEnqueue/DoDequeue</td>
<td>• Implementing all methods on the left</td>
</tr>
<tr>
<td>§ CheckConfig/InitializeParams</td>
<td>• ABBCoDelQueueDisc added methods:</td>
</tr>
<tr>
<td>FqCoDelFlow subclasses QueueDiscClass by adding methods:</td>
<td>§ Optimize</td>
</tr>
<tr>
<td>§ Get/Set/IncreaseDeficit</td>
<td>§ GetFlowInfo (using the callbacks below)</td>
</tr>
<tr>
<td>§ Get/SetStatus</td>
<td>§ <strong>Callbacks:</strong> <code>flowIdCb</code> &amp; <code>flowWeightCb</code></td>
</tr>
<tr>
<td>§ Get/SetIndex</td>
<td>• ABBCoDelBin added methods:</td>
</tr>
<tr>
<td></td>
<td>§ Get/SetBandwidthThreshold</td>
</tr>
<tr>
<td></td>
<td>§ Get/Set/IncreaseWeight</td>
</tr>
</tbody>
</table>
Preliminary result

Simulated network topology
Preliminary result

• Configuration:
  - Five FTP-type of flows sent to five LAN nodes through a LAN router
  - Results to be compared with LAN router runs Fifo, FqCoDel, and ABBCoDel
  - Simulation time: 100s and 1000s
  - ABBCoDel flow reclassification interval: 1s
For performance comparison, LAN router network device is set to use FIFO, Fq-CoDel, and ABBCoDel respectively.

All other network devices are set to use pfifo-fast.

<table>
<thead>
<tr>
<th>Flow</th>
<th>FIFO</th>
<th>FqCoDel</th>
<th>ABBCoDel</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>198.93</td>
<td>142.71</td>
<td>119.06</td>
</tr>
<tr>
<td>2</td>
<td>94.47</td>
<td>147.72</td>
<td>167.48</td>
</tr>
<tr>
<td>3</td>
<td>151.32</td>
<td>143.32</td>
<td>113.83</td>
</tr>
<tr>
<td>4</td>
<td>89.81</td>
<td>144.38</td>
<td>147.61</td>
</tr>
<tr>
<td>5</td>
<td>169.02</td>
<td>140.76</td>
<td>160.17</td>
</tr>
<tr>
<td>JFI</td>
<td>0.9164</td>
<td>0.9997</td>
<td>0.9773</td>
</tr>
<tr>
<td>MMR</td>
<td>0.4515</td>
<td>0.9529</td>
<td>0.6797</td>
</tr>
</tbody>
</table>
## Preliminary result

### Bandwidth Allocation (1000s)

<table>
<thead>
<tr>
<th>Flow</th>
<th>FIFO</th>
<th>FqCoDel</th>
<th>ABBCoDel</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>126.91</td>
<td>145.14</td>
<td>133.02</td>
</tr>
<tr>
<td>2</td>
<td>114.76</td>
<td>142.86</td>
<td>149.08</td>
</tr>
<tr>
<td>3</td>
<td>163.63</td>
<td>142.28</td>
<td>146.71</td>
</tr>
<tr>
<td>4</td>
<td>137.47</td>
<td>144.88</td>
<td>134.23</td>
</tr>
<tr>
<td>5</td>
<td>162.61</td>
<td>144.47</td>
<td>149.43</td>
</tr>
<tr>
<td>JFI</td>
<td>0.9815</td>
<td>0.9999</td>
<td>0.9974</td>
</tr>
<tr>
<td>MMR</td>
<td>0.7013</td>
<td>0.9803</td>
<td>0.8902</td>
</tr>
</tbody>
</table>
Experience and challenges of ns-3 implementation

• ns-3
  ▪ A very nice platform for writing code, but is also a huge “beast” to tame
    ➢ Went through several iterations to get it right
  ▪ Flow monitor: a big saver for getting performance metrics for my network simulation
  ▪ The logging system is great in helping me with the debugging

• Questions for the experts and wishes
  ▪ ns-3 is slow: wish it could be faster
  ▪ Too much log: can we ask for the logs from only the last few seconds before crashing?
  ▪ Default qdisc has been changed from pfifo-fast to FqCoDel:
    ➢ This changes everything for us to do qdisc research: Is it the right move?
Future work

• Tiering support
  ▪ It is the common practice that an ISP provides several service tiers to its subscribers in exchange for different charges

• Better approximation
  ▪ By adjusting bin weights?
Questions?