Implementation of PI\(^2\) Queuing Discipline for Classic TCP Traffic in ns-3

Rohit P. Tahiliani, Hitesh Tewari
School of Computer Science & Statistics,
Trinity College Dublin, Ireland
tahiliar@tcd.ie
Outline of the presentation

- Introduction: *Bufferbloat*, PIE, PI$^2$
- Motivation
- Contributions
- Implementation details
- Model evaluation
- Functional verification
- Conclusions & Future Work
- Relation to the Future of Internet Transport
- Acknowledgements
Introduction: Bufferbloat

- Inexpensive memory.

- *Side effect:* Bloated buffers at routers!

- *Bufferbloat:* large queueing delays

- *Potential solution:* deploy AQM algorithms to control queue delay

Popular AQM algorithms:

- RED / Adaptive RED [S. Floyd, V. Jacobson, ...]

- CoDel / Fair Queue CoDel [K. Nichols, V. Jacobson, ...]

- PIE [R. Pan, P. Natarajan, ...]
Introduction: PI$^2$

- PI$^2$ - Extends PIE to support Classic & Scalable Congestion Control.

Three major components of PI$^2$:

- Random dropping
  - based on drop probability. PI$^2$ applies the squared drop probability.

- Drop probability calculation
  - happens at a regular interval.

- Average departure rate estimation
  - only when there is sufficient amount of data.
Motivation

- Latency of 300ms appears to be “slow” [1]

- *Bufferbloat* makes the situation worse.

*Why implement $PI^2$ in ns-3:*

- No support of $PI^2$ in network simulators.

- Adds value to the ongoing research work to solve *Bufferbloat*.

- ns-3: several new features compared to other simulators.

Contributions

- Developed a new model for PI\(^2\) in ns-3.
- Preliminary verification by writing test cases in ns-3.
- Evaluation by comparing results obtained from ns-3 PIE model and ns-3 PI\(^2\) model.
- ns-3 PI\(^2\) model is currently under review and can be accessed here [1].

Limitations:

- Currently, this ns-3 PI\(^2\) model supports only Classic Traffic.

Implementation details

Source location:
src/traffic-control/model/pi-square-queue{.h, .cc}

Fig. 1: Class diagram for PI² model in ns-3.
Fig. 2: Interaction between the core methods of PI²
Model Evaluation

• A test suite for evaluating the working of PI$^2$ algorithm.
  - verifies the attribute settings of PI$^2$ parameters.
  - basic enqueue / dequeue of packets.

• Compare PI$^2$ in ns-3 with PIE in ns-3 under same scenarios.

• Performance metrics under observation:
  - Queue delay.
  - Throughput.
Functional verification

Four simulation scenarios:
1. Light TCP traffic
2. Heavy TCP traffic
3. Mix TCP and UDP traffic
4. CDF of Queuing Delay

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Topology</td>
<td>Dumbbell</td>
</tr>
<tr>
<td>Bottleneck RTT</td>
<td>76ms</td>
</tr>
<tr>
<td>Bottleneck buffer size</td>
<td>200KB</td>
</tr>
<tr>
<td>Bottleneck bandwidth</td>
<td>10Mbps</td>
</tr>
<tr>
<td>Bottleneck queue</td>
<td>PI^2</td>
</tr>
<tr>
<td>Non-bottleneck RTT</td>
<td>2ms</td>
</tr>
<tr>
<td>Non-bottleneck bandwidth</td>
<td>10Mbps</td>
</tr>
<tr>
<td>Non-bottleneck queue</td>
<td>DropTail</td>
</tr>
<tr>
<td>Mean packet size</td>
<td>1000B</td>
</tr>
<tr>
<td>TCP</td>
<td>NewReno</td>
</tr>
<tr>
<td>target</td>
<td>20ms</td>
</tr>
<tr>
<td>tupdate</td>
<td>30ms</td>
</tr>
<tr>
<td>alpha</td>
<td>PIE - 0.125, PI^2 - 0.3125</td>
</tr>
<tr>
<td>beta</td>
<td>PIE - 1.25, PI^2 - 3.125</td>
</tr>
<tr>
<td>dq_threshold</td>
<td>10KB</td>
</tr>
<tr>
<td>Application start time</td>
<td>0s</td>
</tr>
<tr>
<td>Application stop time</td>
<td>99s</td>
</tr>
<tr>
<td>Simulation stop time</td>
<td>100s</td>
</tr>
</tbody>
</table>

Table 1: Simulation Setup
Functional verification: Light TCP traffic

Fig. 3: Queue Delay with Light TCP traffic.
Fig. 4: Link Throughput with Light TCP traffic.
Fig. 5 : Queue Delay with Heavy TCP traffic.
Functional verification: Heavy TCP traffic

Fig. 6: Link Throughput with Heavy TCP traffic.
Fig. 7: Queue Delay with mix TCP and UDP traffic.
Fig. 8: Link Throughput with mix TCP and UDP traffic.
Functional verification: CDF of Queue Delay

20 TCP flows and target delay = 5ms  
20 TCP flows and target delay = 20ms

Fig. 9: CDF of Queuing Delay with 20 TCP flows.
Functional verification: CDF of Queue Delay

Fig. 10: CDF of Queuing Delay with 5 TCP and 2 UDP flows.

5 TCP + 2 UDP with target delay=5ms  
5 TCP + 2 UDP with target delay=20ms
Conclusions & Future Work

- A ns-3 model for PI$^2$ has been implemented and evaluated.

- Results obtained are compared to those of ns-3 PIE model.

Next Tasks:

- Extend PI$^2$ to work with Explicit Congestion Notification (ECN).

- Merge it into the main line of ns-3.

- Extend PI$^2$ in ns-3 for Scalable Congestion Control such as DCTCP.

- Compare PI$^2$ in ns-3 with PI$^2$ implementation in Linux.
This work is inline with the ongoing research in the area of:

- DualQ Coupled AQM for Low Latency, Low Loss Scalable throughput.
- TCP Prague.
Acknowledgements

- To all the reviewers, for the highly effective reviews!
- Prof. Stein Gjessing for shepherding our paper.
- Workshop organizers for providing this opportunity.
- Trinity College Dublin, Ireland for providing financial assistance.
Thank you.