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

Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

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.

[1] Grigorik, I. (2013). High Performance Browser Networking: What every web developer should know about networking and web performance. "O'Reilly Media, Inc.".

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.

[1] https://codereview.appspot.com/314290043/

Implementation details

Source location:

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

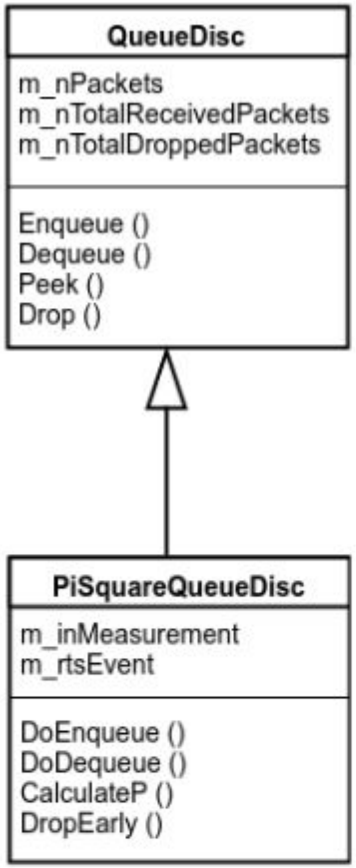


Fig. 1: Class diagram for PI^2 model in ns-3.

Implementation details

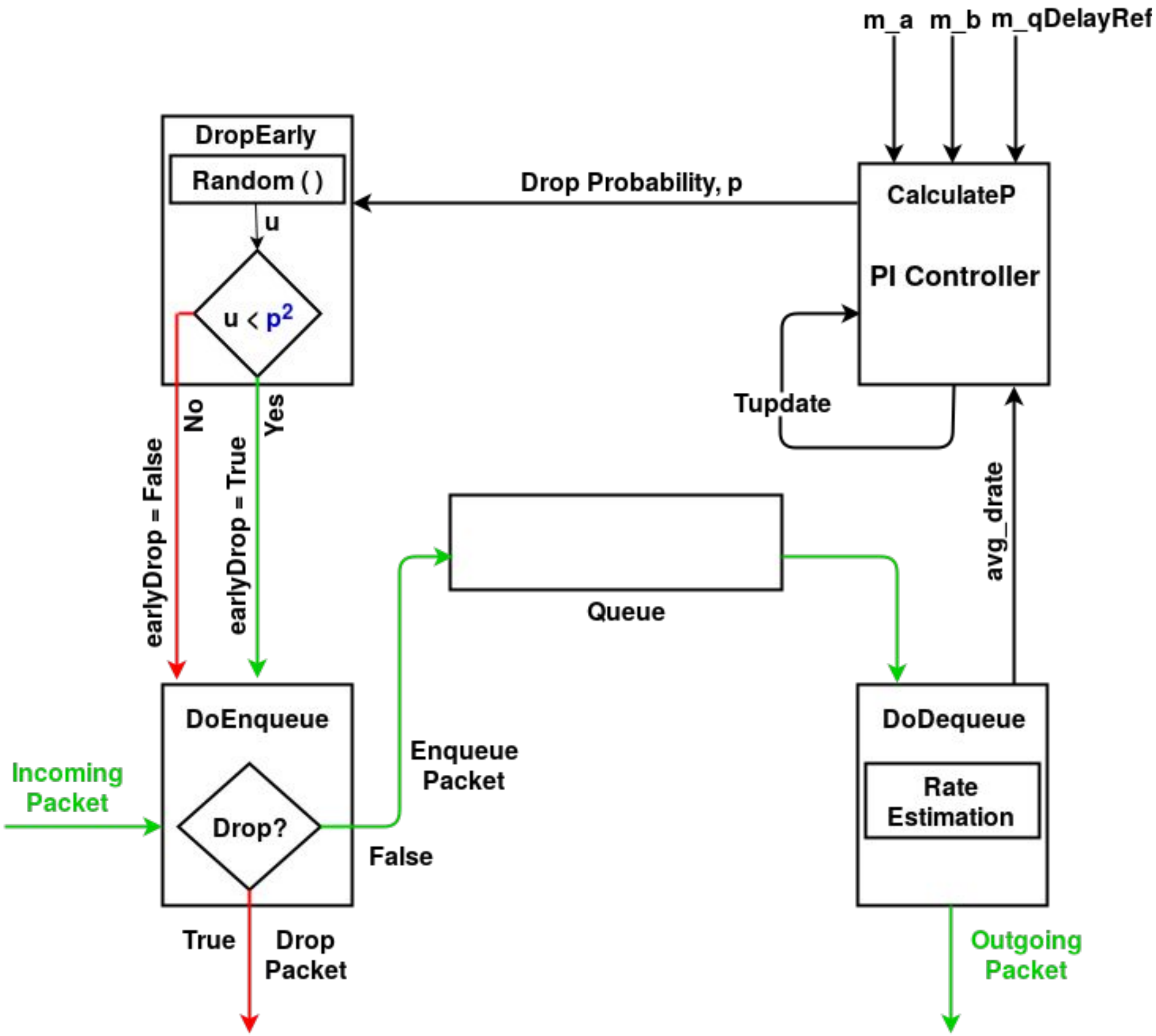


Fig. 2: Interaction between the core methods of PI^2

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

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

Table 1: Simulation Setup

Functional verification: Light TCP traffic

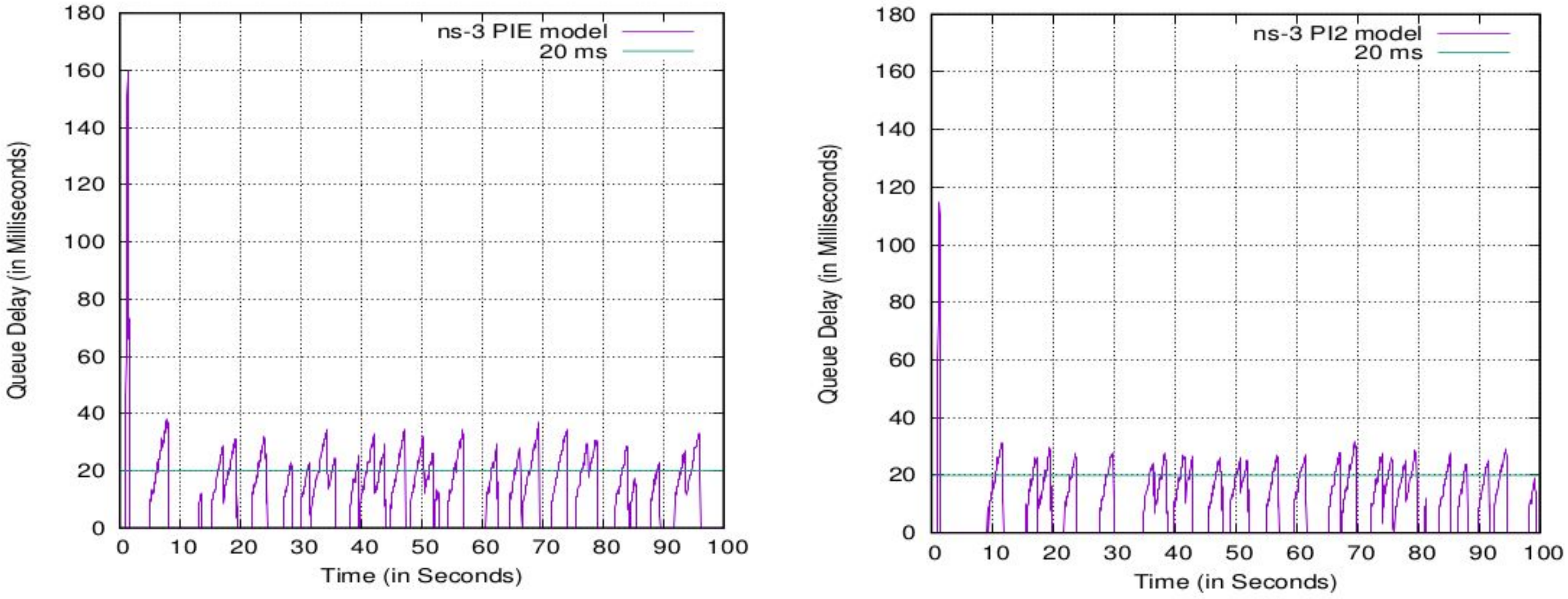


Fig. 3: Queue Delay with Light TCP traffic.

Functional verification: Light TCP traffic

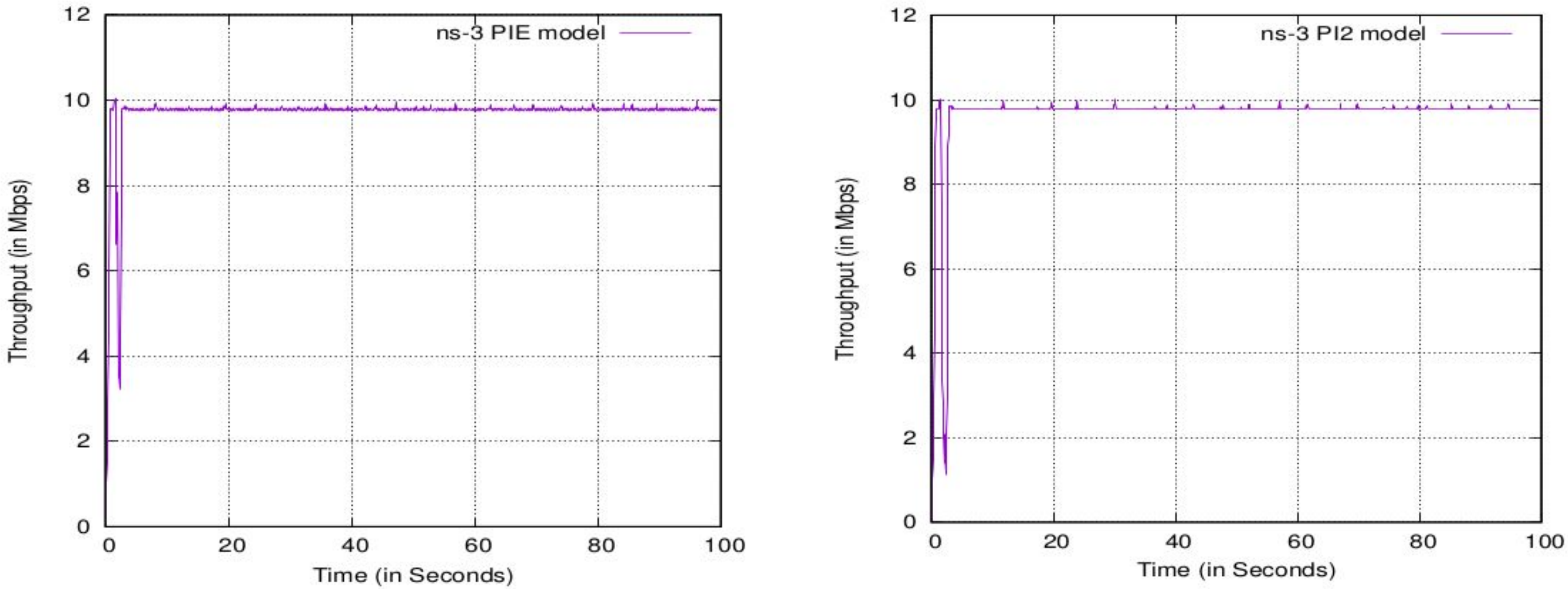


Fig. 4: Link Throughput with Light TCP traffic.

Functional verification: Heavy TCP traffic

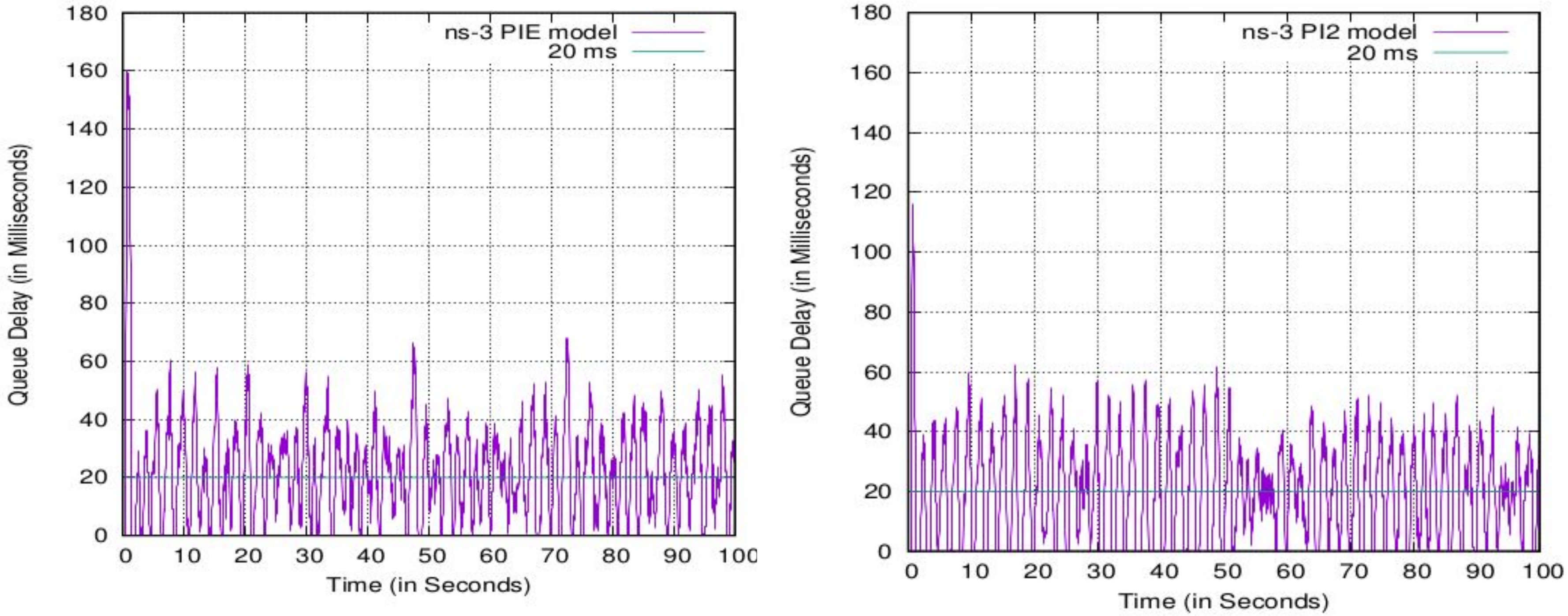
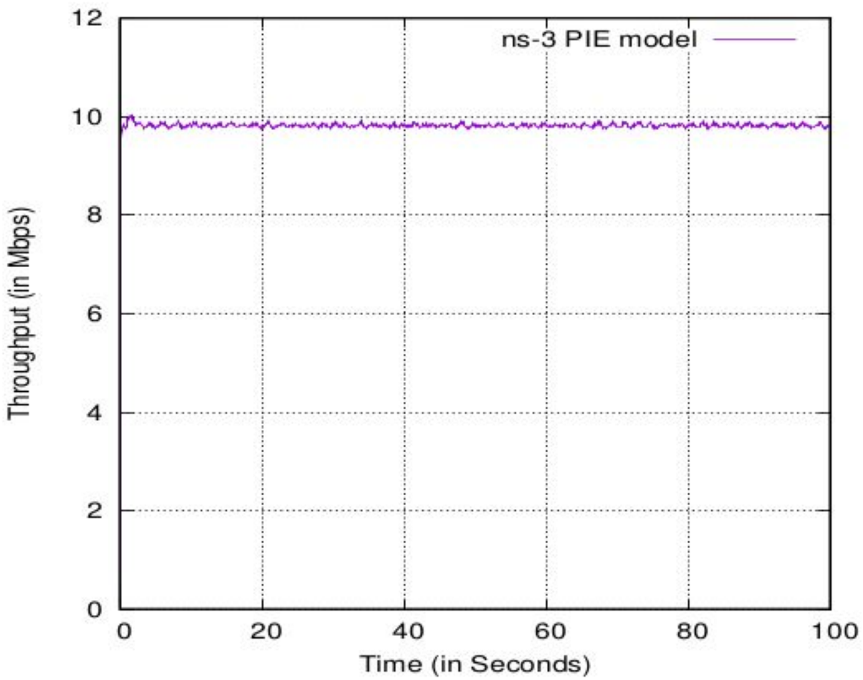


Fig. 5 : Queue Delay with Heavy TCP traffic.

Functional verification: Heavy TCP traffic



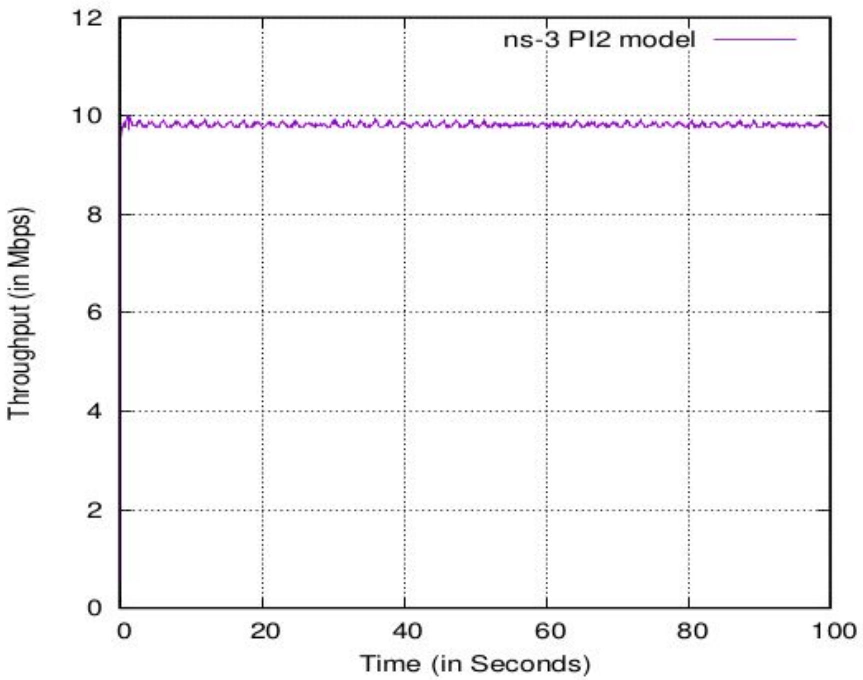


Fig. 6: Link Throughput with Heavy TCP traffic.

Functional verification: Mix TCP and UDP traffic

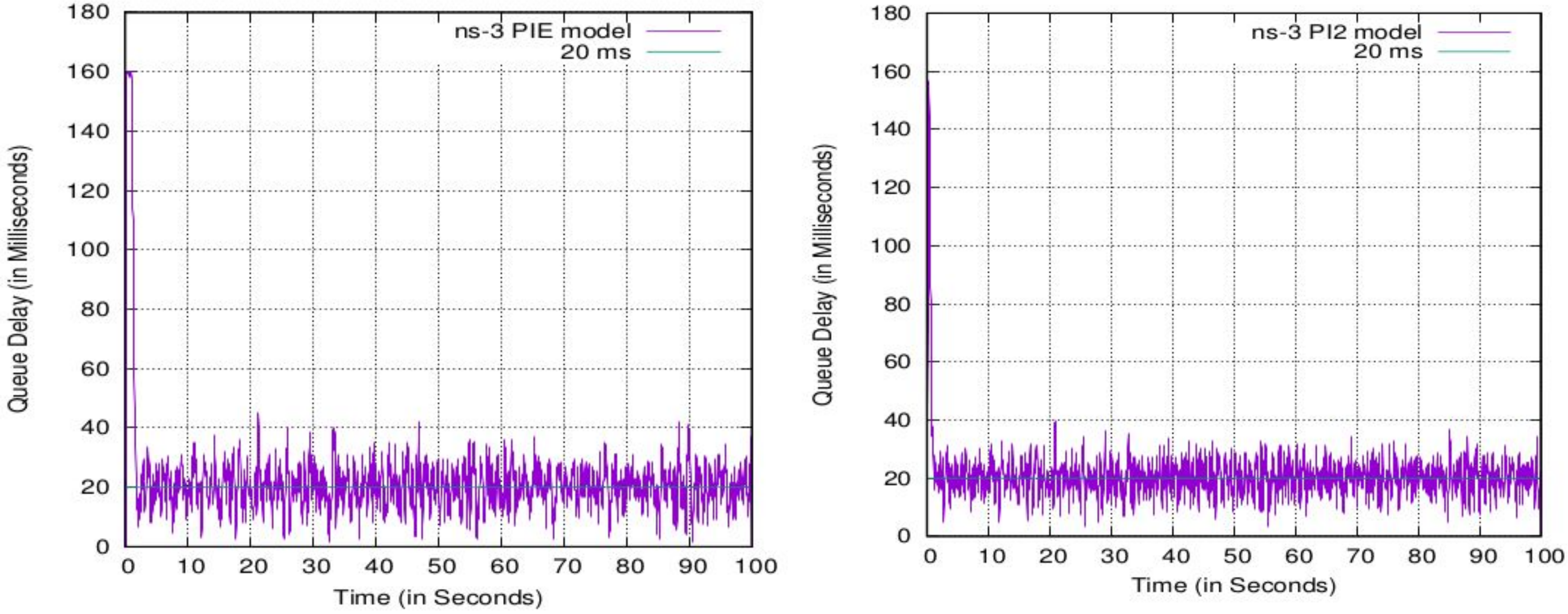


Fig. 7: Queue Delay with mix TCP and UDP traffic.

Functional verification: Mix TCP and UDP traffic

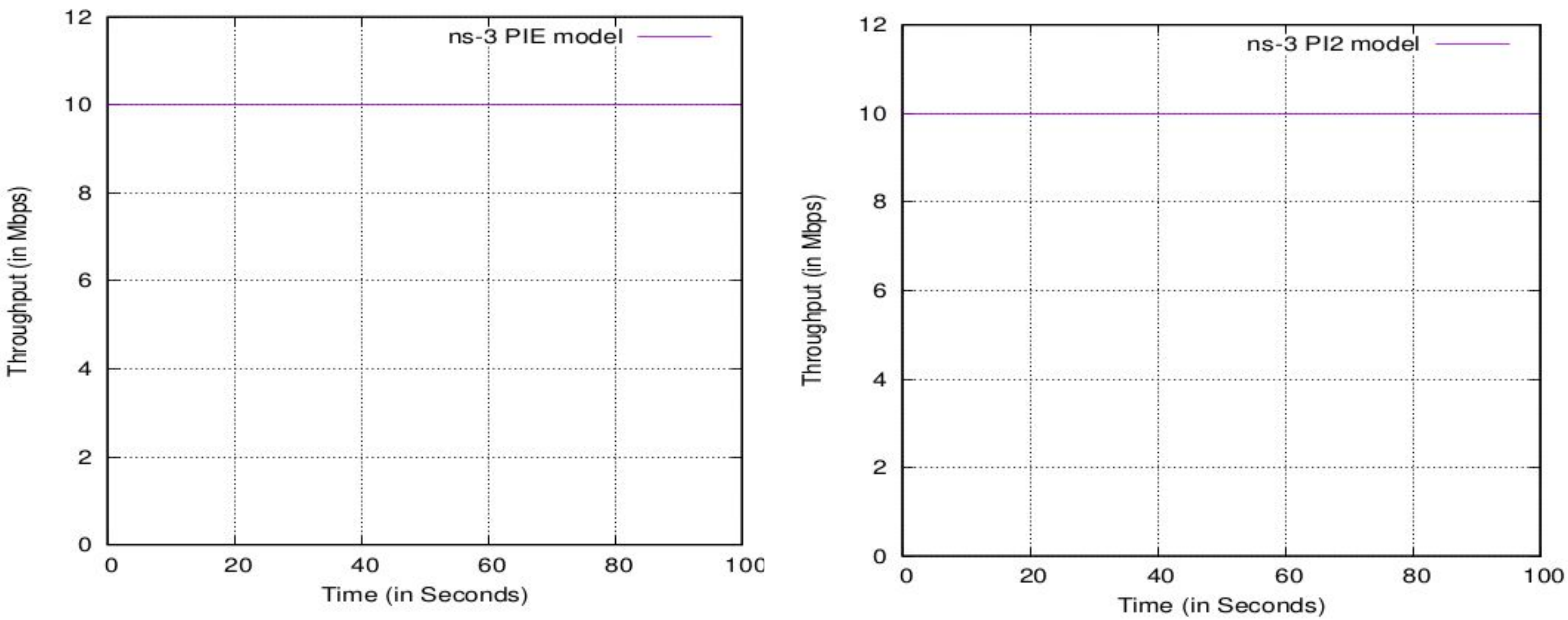
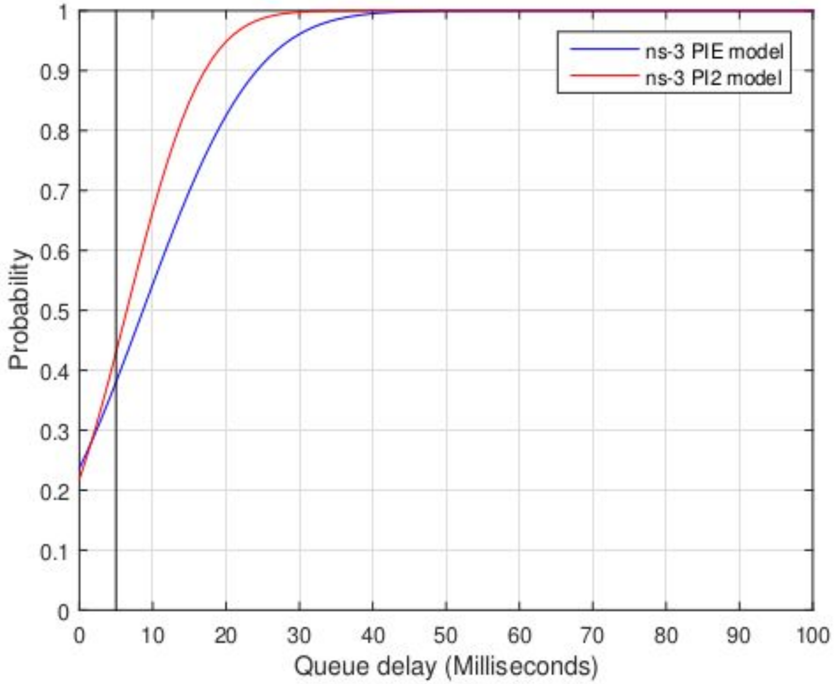
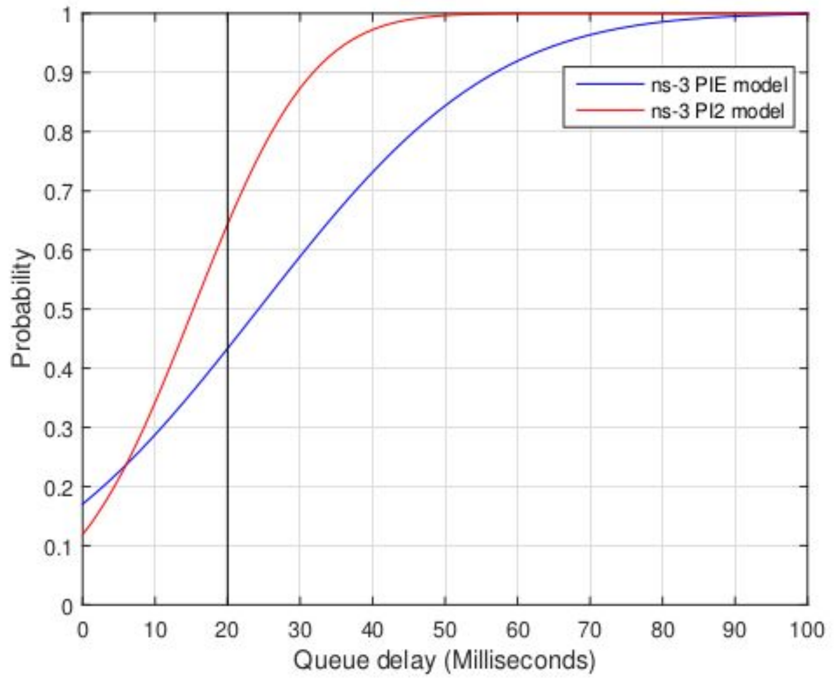


Fig. 8: Link Throughput with mix TCP and UDP traffic.

Functional verification: CDF of Queue Delay



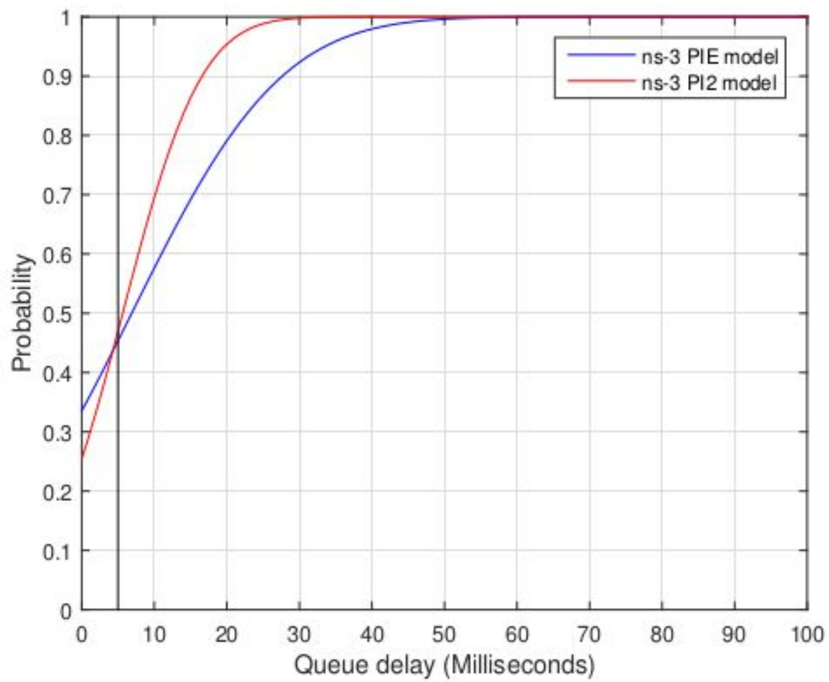
20 TCP flows and target delay = 5ms



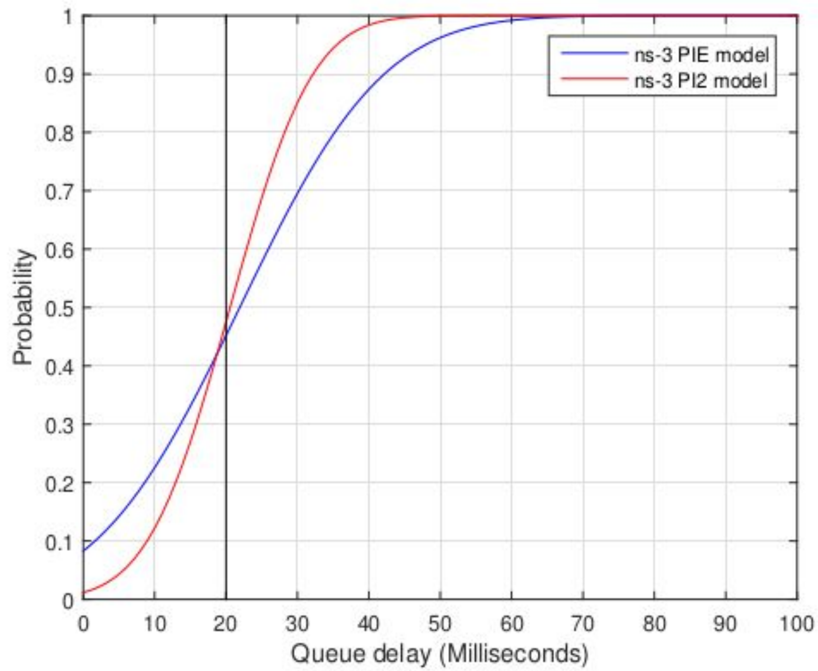
20 TCP flows and target delay = 20ms

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

Functional verification: CDF of Queue Delay



5 TCP + 2 UDP with target delay=5ms



5 TCP + 2 UDP with target delay=20ms

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

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.

Relation to the Future of Internet Transport

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.