PI2

AQM for classic and scalable congestion control

Koen De Schepper

Bob Briscoe

Olga Bondarenko

Inton Tsang

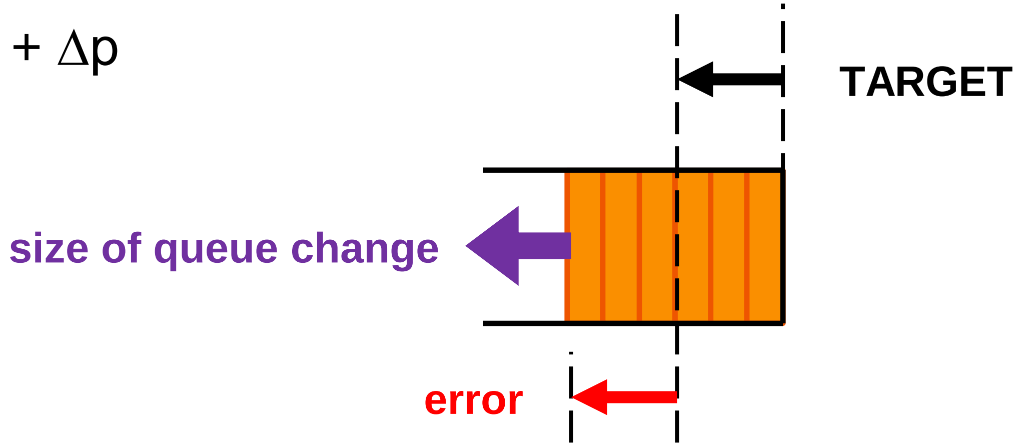
April 4th, 2016

PI recap

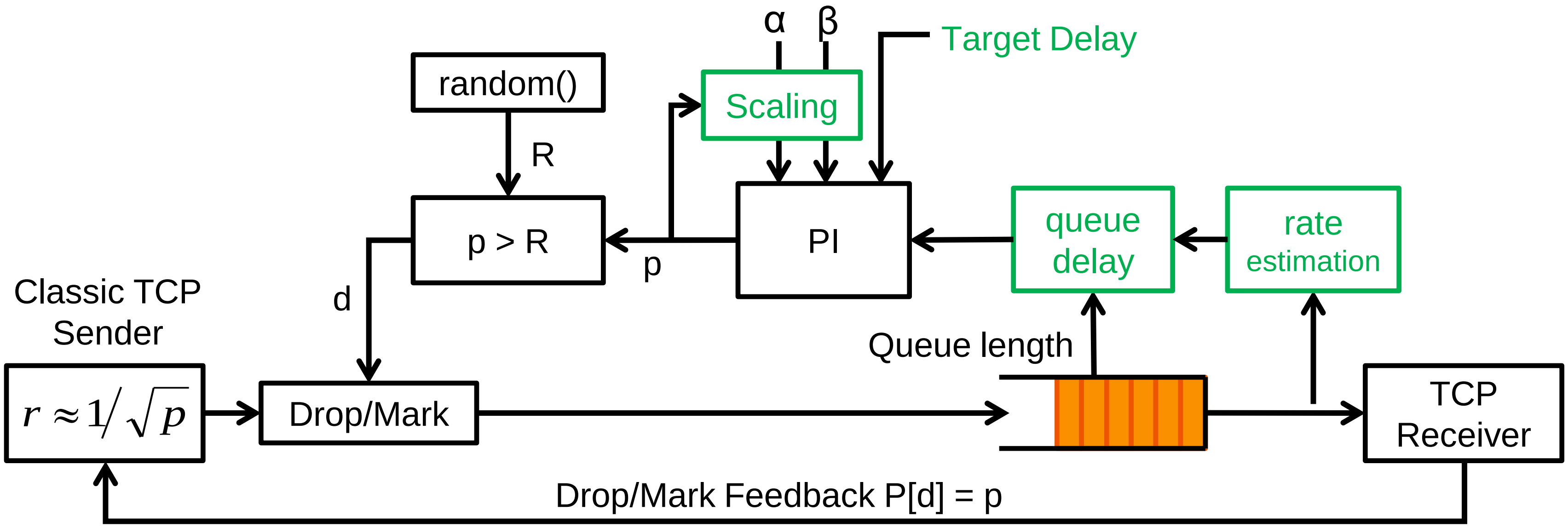
Every T_{update} interval do:

$$\Delta p = \alpha * (current\_queue - TARGET) + \beta * (current\_queue - prev\_queue)$$

$$p = p + \Delta p$$



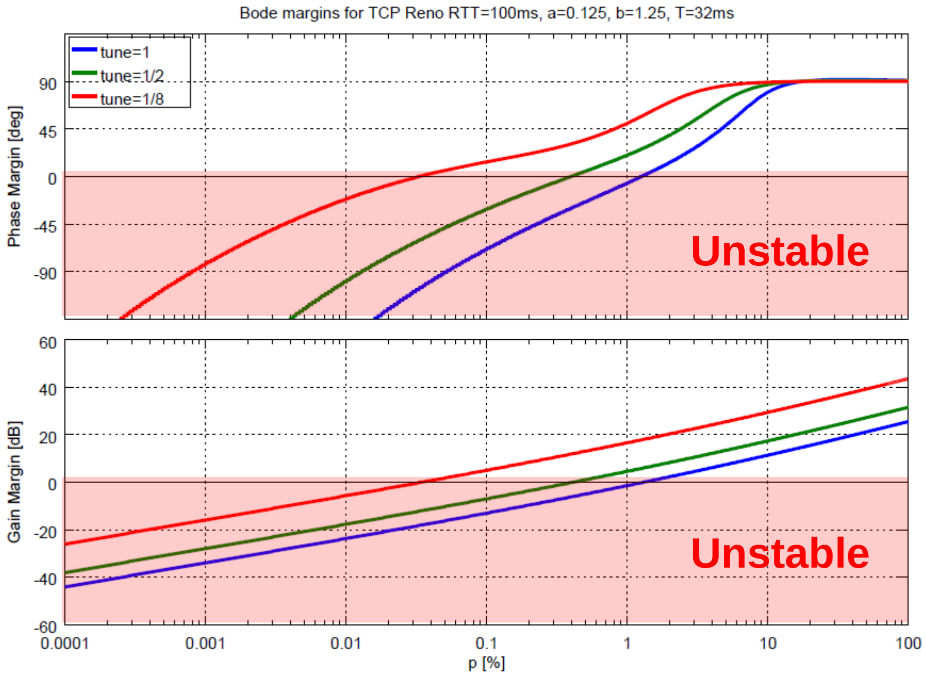
PIE



- Enhancements are: rate estimation, queue delay and gain scaling

Choosing α and β gains

- Higher α and β values give faster response
- To be stable, the phase and gain margins must be > 0



- Gain margin evolves diagonally with p

- Reason is $\sqrt{\ }$ in

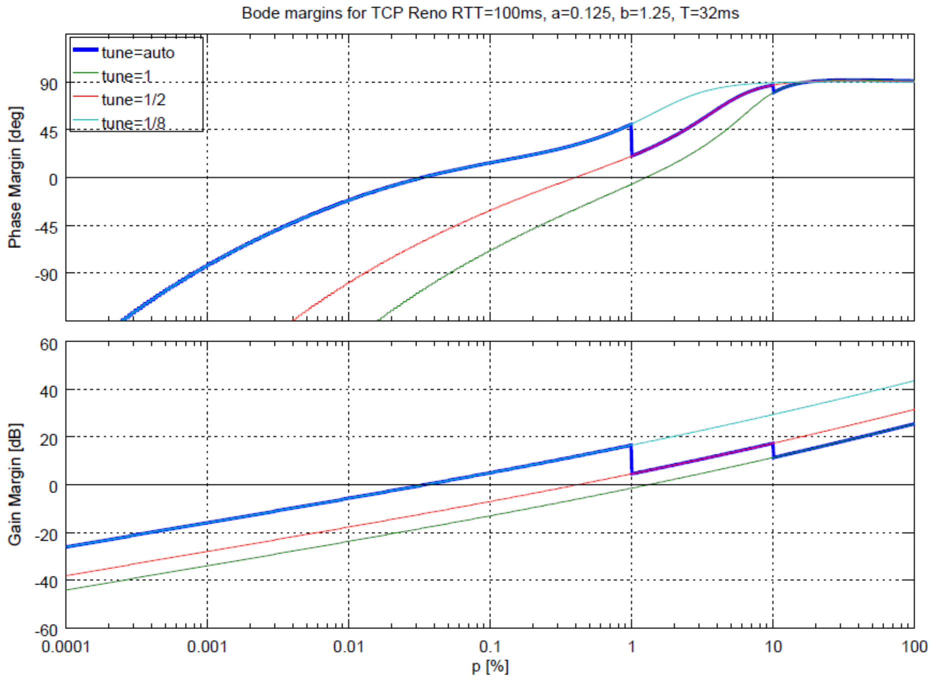
$$r_{reno} = \frac{1.22}{\sqrt{p}.\mathrm{RTT}}$$

PIE solution: α and β scaling

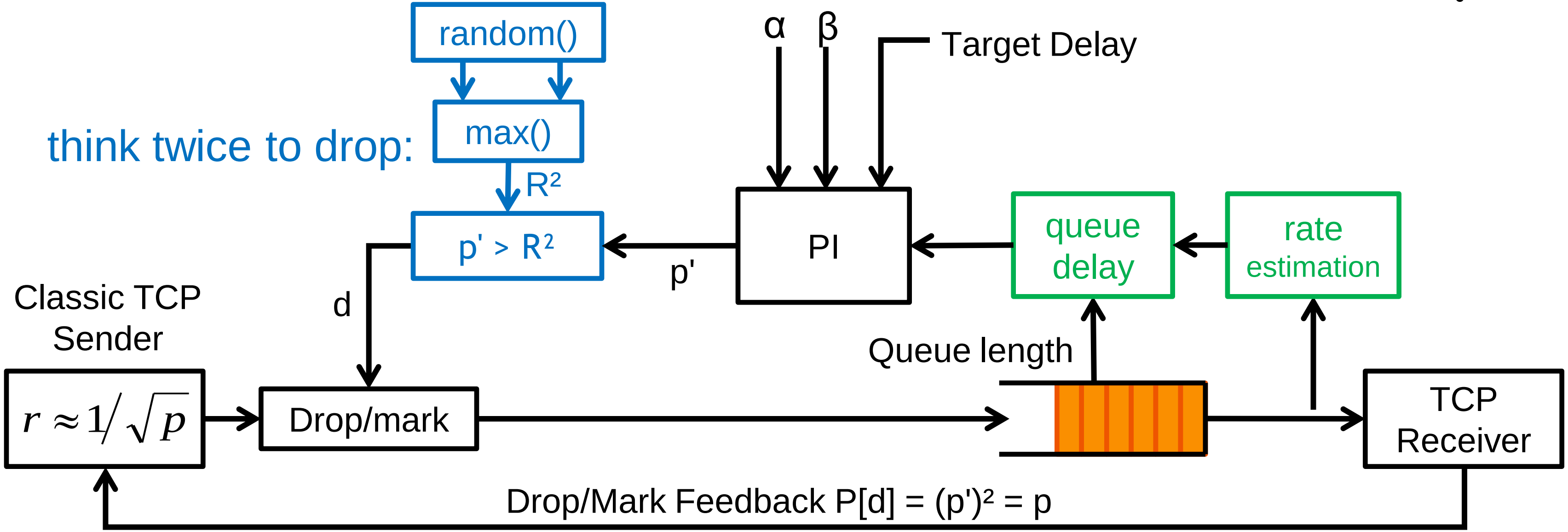
- PIE in Linux has extra internal alpha and beta parameters:

```
if (p<1%)
   at=a/8
   bt=b/8
else if (p<10%)
   at=a/2
   bt=b/2
else
   at=a
   bt=b
```

- Current PIE draft has more if's

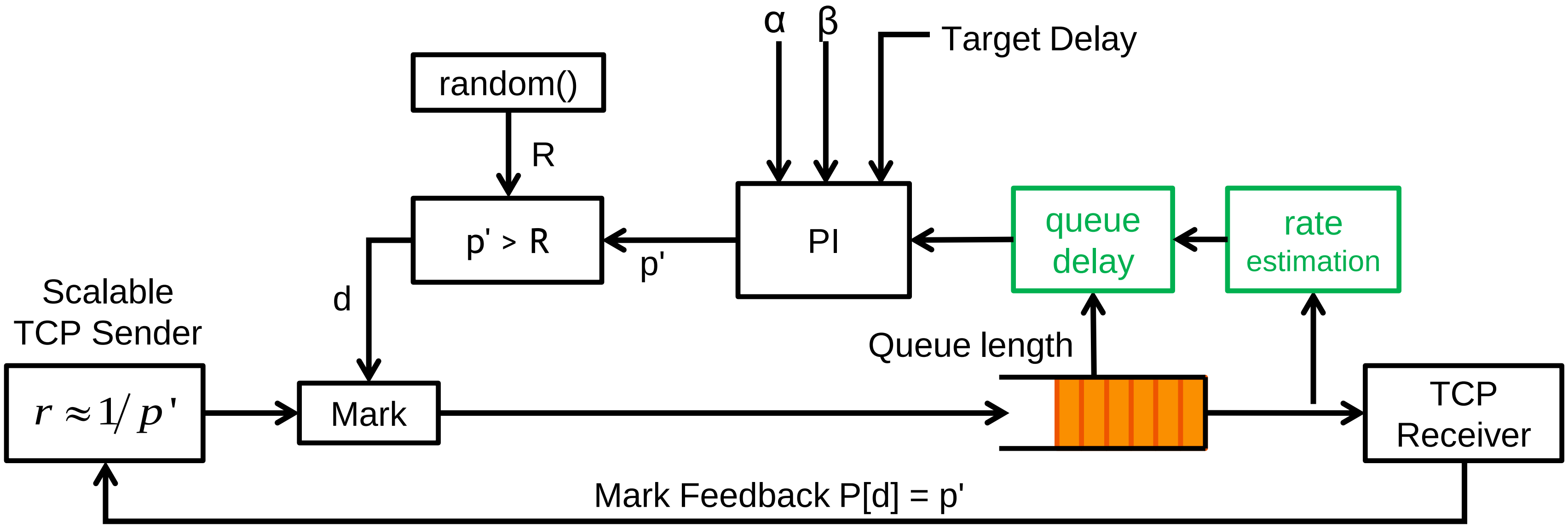


PI2 solution: remove the $\sqrt{\ }$



- Replaces gain scaling with a square: $P[d] = (p')^2 = p$
- PI2 controls p‘ which is actually $\sqrt{p}$ so $r \approx 1/p'$

PI2 also supports scalable TCP



- Scalable TCP needs no scaling, nor squaring
- Can use the same parameters as PI2 for Reno or Cubic

PI2 needs no α and β scaling

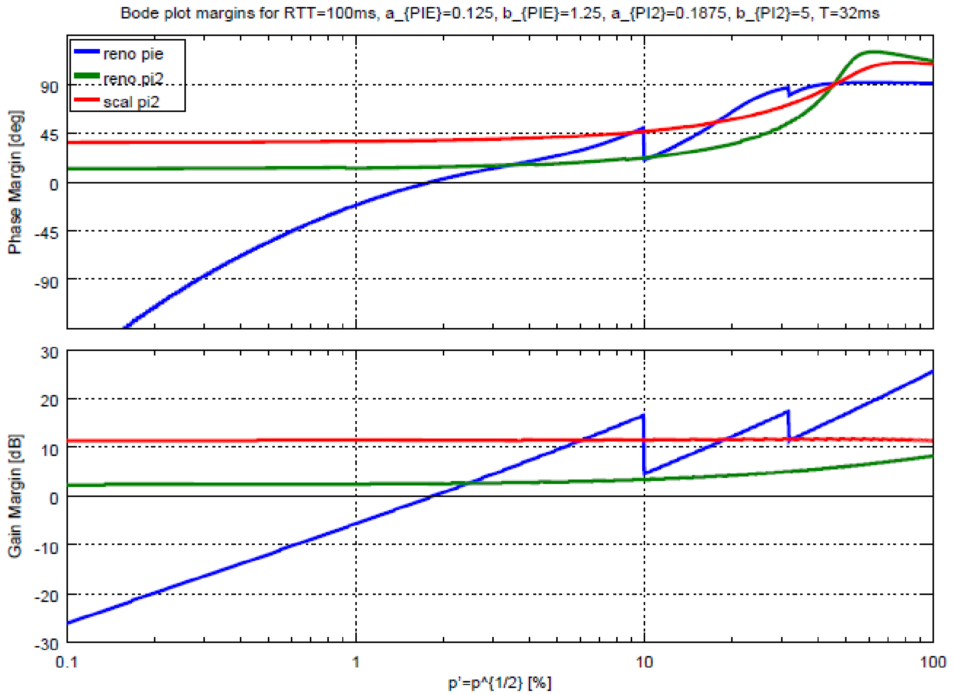
- By squaring at the end, Reno can be controlled like a Scalable TCP
- Models used for:
  - TCP Reno on PI: $$\frac{dW(t)}{dt} = \frac{1}{R(t)} - 0.5\frac{W(t)W(t-R(t))}{R(t-R(t))}p(t-R(t))$$ [1][2]
  - TCP Reno on PI2: $$\frac{dW(t)}{dt} = \frac{1}{R(t)} - 0.5\frac{W(t)W(t-R(t))}{R(t-R(t))}\left(p'(t-R(t))\right)^2$$
  - Scalable TCP on PI2: $$\frac{dW(t)}{dt} = \frac{1}{R(t)} - 0.5\frac{W(t-R(t))}{R(t-R(t))}p'(t-R(t))$$

[1] V. Misra, W.-B. Gong, and D. Towsley, "Fluid-based Analysis of a Network of AQM Routers Supporting TCP Flows with an Application to RED," SIGCOMM Computer Comms.. Review, vol. 30, no. 4, pp. 151–160, Aug. 2000.

[2] C. V. Hollot, V. Misra, D. F. Towsley, and W. Gong, "A Control Theoretic Analysis of RED," in Proc. INFOCOM 2001. 20th Annual Joint Conf. of the IEEE Computer and Communications Societies., vol. 3, 2001, pp. 1510—19.

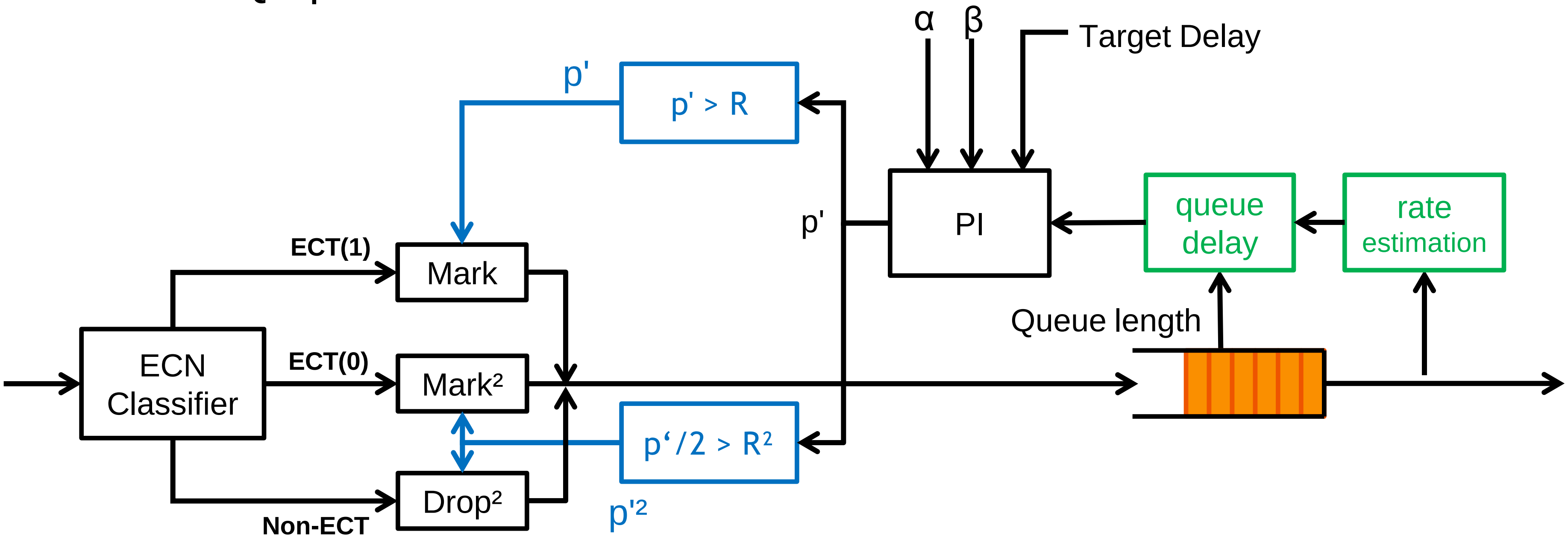
PI2 needs no α and β scaling

- By squaring at the end, Reno can be controlled like a Scalable TCP
- Margins stay above 0 independent from p' getting smaller
- Gain margin evolves flat with p'



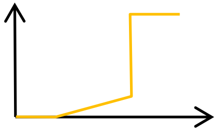
Single Q PI2 experiments

- Lunix implementation
- DualQ option not used here

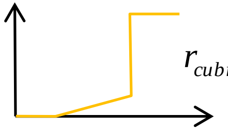


Evaluation with Cubic & DCTCP

Steady state rate has been modeled for existing CC-AQM schemes:



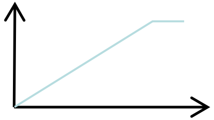
$$r_{reno} = \frac{1.22}{p^{1/2}\mathrm{RTT}}$$



$$r_{cubic} = \frac{1.17}{p^{3/4}\mathrm{RTT}^{1/4}}$$

$$r_{dc} = \frac{2}{p^2 \cdot \mathrm{RTT}}$$

Deviations are DCTCP on non-step threshold:



$$r_{dc\_p} = \frac{2}{p \cdot \mathrm{RTT}}$$

Cubic in Reno mode:

$$r_{creno} = \frac{1.68}{p^{1/2}\mathrm{RTT}}$$

Experiments:

- RTT: 5, 10, 20, 50, 100 ms
- Rate: 4, 12, 40, 200, 400 Mbps

Probability coupling used:

$$p_{creno} = \left(\frac{p_{dc\_p}}{2}\right)^2$$

Dynamic tests

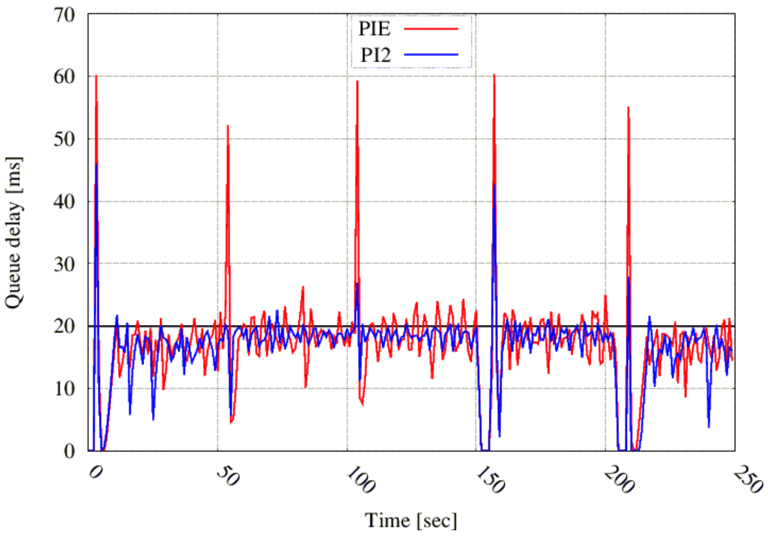
PIE
α=0.125*t
β=1,25*t

PI2
α=0.5
β=10

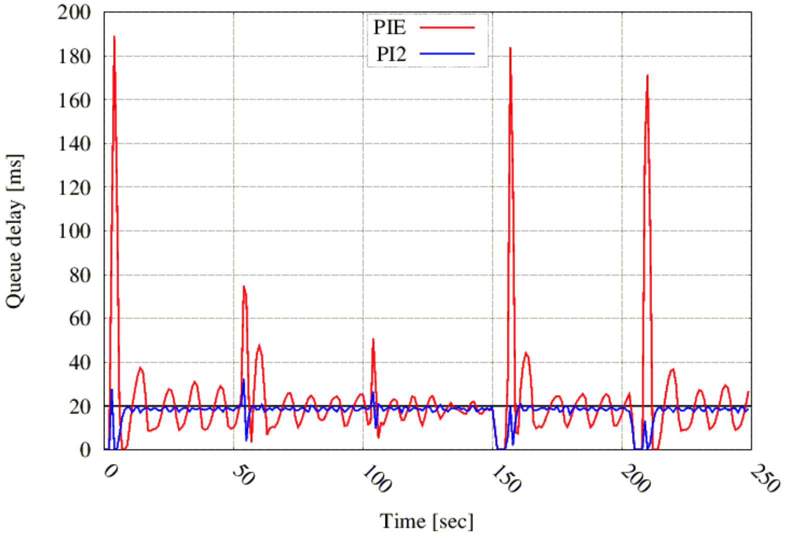
Fig 8 in [3]: Note: plots show average queue delay over 1s intervals

0-50s: 10 TCP flows; 50-100s: 30 TCP flows; 100-150s: 50 TCP flows; 150-200s: 30 TCP flows; 200-250s: 10 TCP flows; RTT=100ms Link=20Mbps

Cubic



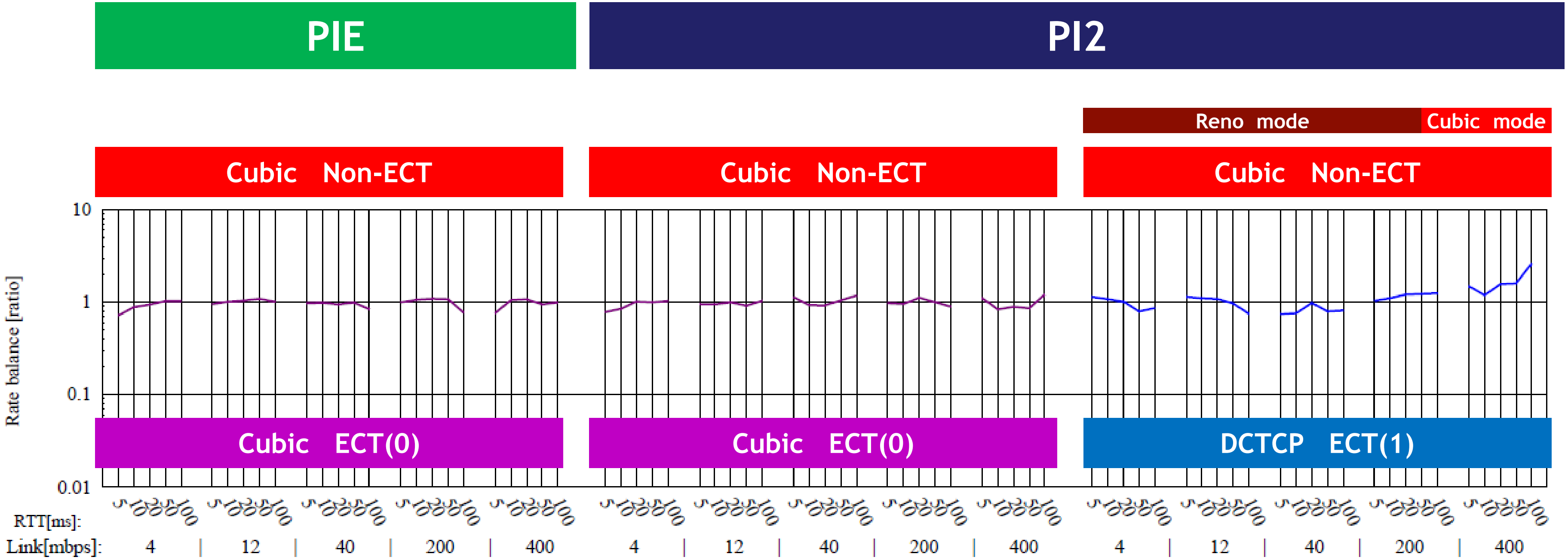
DCTCP



[3] R. Pan et al., “PIE: A lightweight control scheme to address the bufferbloat problem,” in Proc. IEEE Int’l Conf. on High Performance Switching and Routing (HPSR), 2013, pp. 148–155.

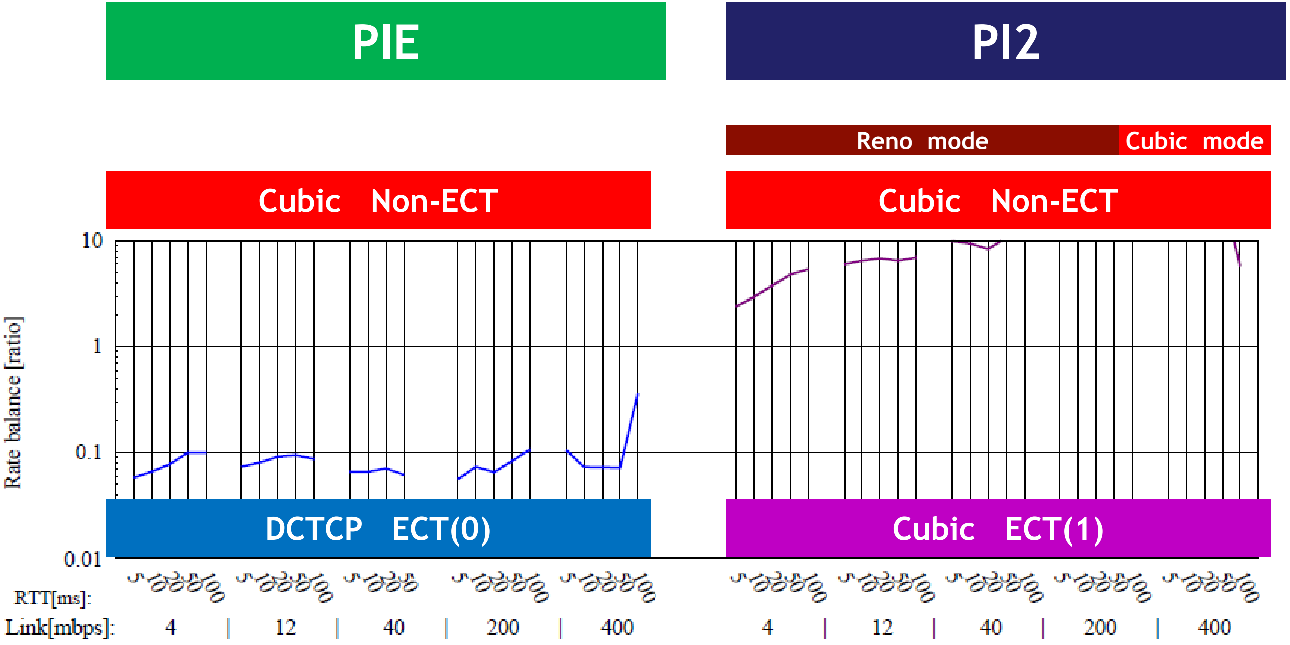
Throughput ratio

one flow each



Throughput ratio

wrong ECN marking; one flow each



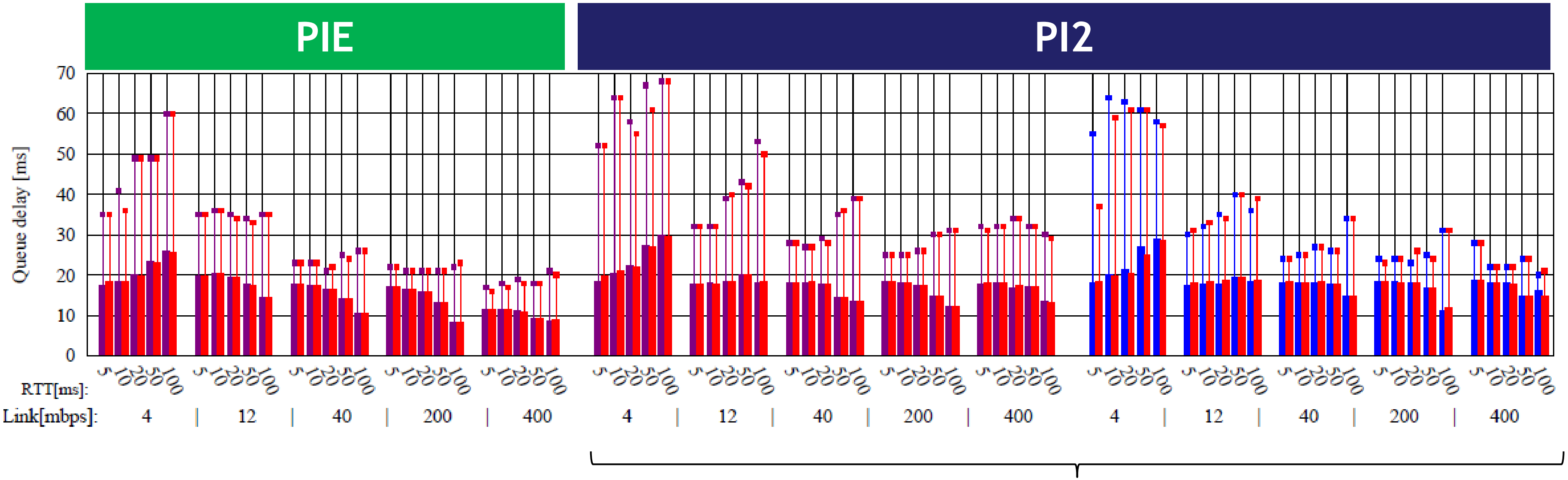
Queuing delay

one flow each

Average Q delay:

- Cubic Non-ECT
- Cubic ECT(0)
- DCTCP ECT(1)

=99th



α=0.125
β=1,25

PI2 using basic α and β PIE parameters → less responsive
Higher gain possible for PI2

DualQ is deployment goal

- L4S (smoothing in the end system) and DualQ with immediate network marking is outperforming any smoothed AQM (such as PIE and PI2)

- PI2 is useful to control Classic Q size, and signal can be applied on both Scalable (L4S) and Classic TCP

- Experiments here without DualQ
  - → no ultra low latency
  - → show impact of one change at a time
  - → focus on throughput fairness

Conclusion

- Scalable (L4S) and Classic TCPs can get equal throughput (tcp-fairness)

- PI2 simplifies Classic queue control and supports Scalable queue / rate control

- PI2 supports higher α and β values

Questions

koen.de_schepper@nokia.com