A Study of Active Queue Management for Congestion Control

Victor Firoiu
Marty Borden

Outline

- ***Introduction***
- ***Feedback Control System Background***
- ***FCS applied to AQM***
- ***Calculating FCS equations***
- ***Simulation verifications***
- ***RED configuration recommendations***
- ***Conclusion***

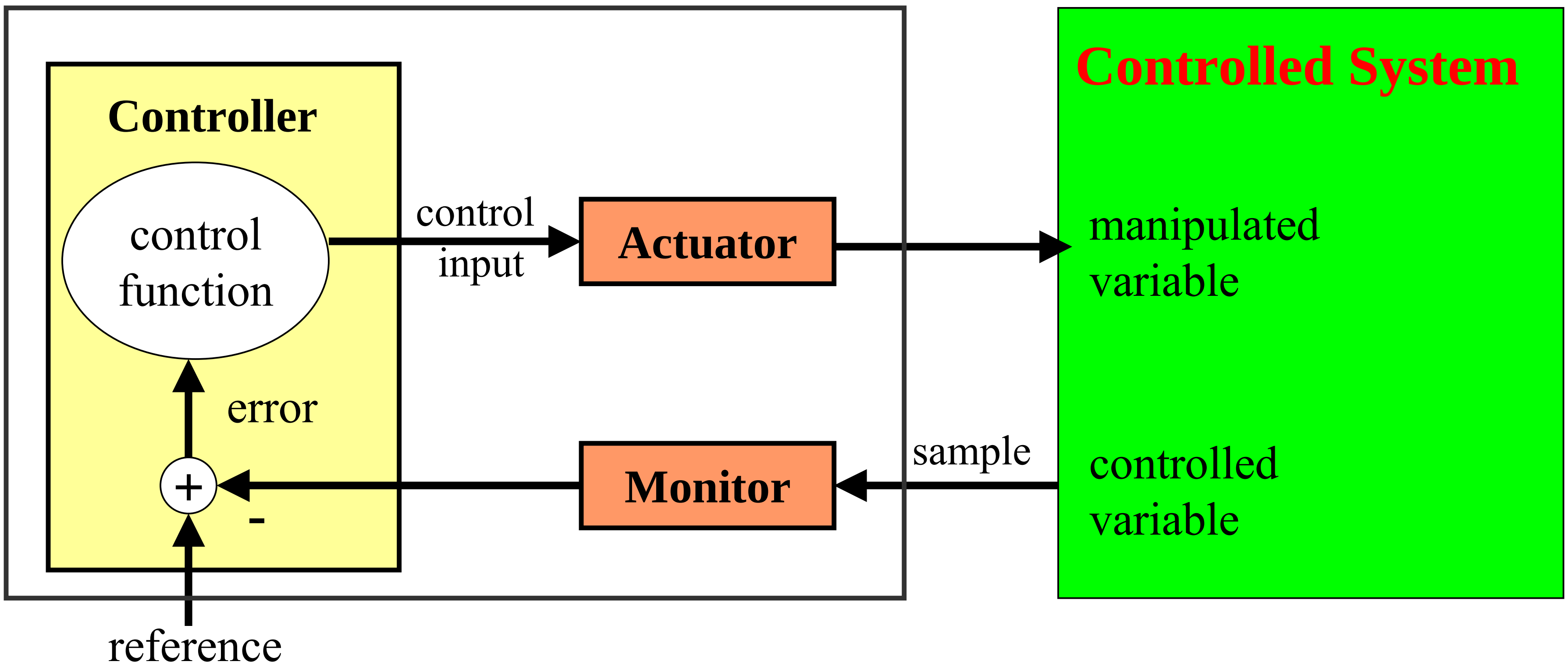
Introduction

- Goal - Determine “best” RED configuration using systematic approach
- Models - queue vs. feedback control system
- Mathematical analysis and fundamental Laws
- Simulation verification of model
- Recommendations
- Future directions

Feedback Control systems

- What is it? – Model where a change in input causes system variables to conform to desired values called the reference
- Why this model ? - Can create a stable and efficient system
- Two basic models - Open vs. Closed loop

Feedback Control (closed loop)



How to apply FCS to AQM

- Try to get two equations to derive steady state behavior – in our case queue function (avg. length of queue) and control function (dependent upon architecture –RED)

Control theory → stability

- Networks as a feedback system
- Distributed & delayed feedback

Model TCP Avg. Queue Size

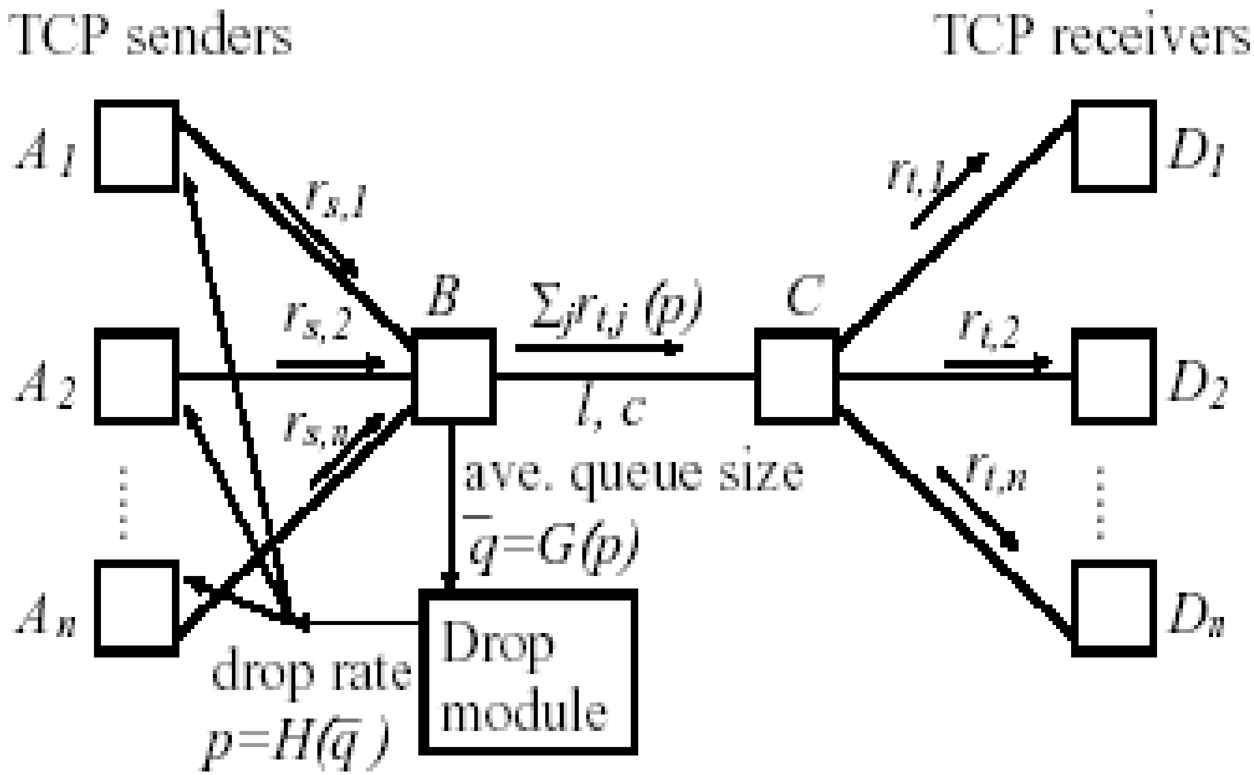


Fig. 1. An n-flow feedback control system

Single flow feedback system

$$\sum_{j=1}^{n} r_{t,j} \leq c$$

$$r_{t,i}(p,\mathrm{R}_i) = \mathrm{T}(p,\mathrm{R}_i)$$

Becomes

$$r_{t,i}(p,\mathrm{R}) \leq c/n,\ 1 \leq i \leq n$$

Finding the Queue "Law"

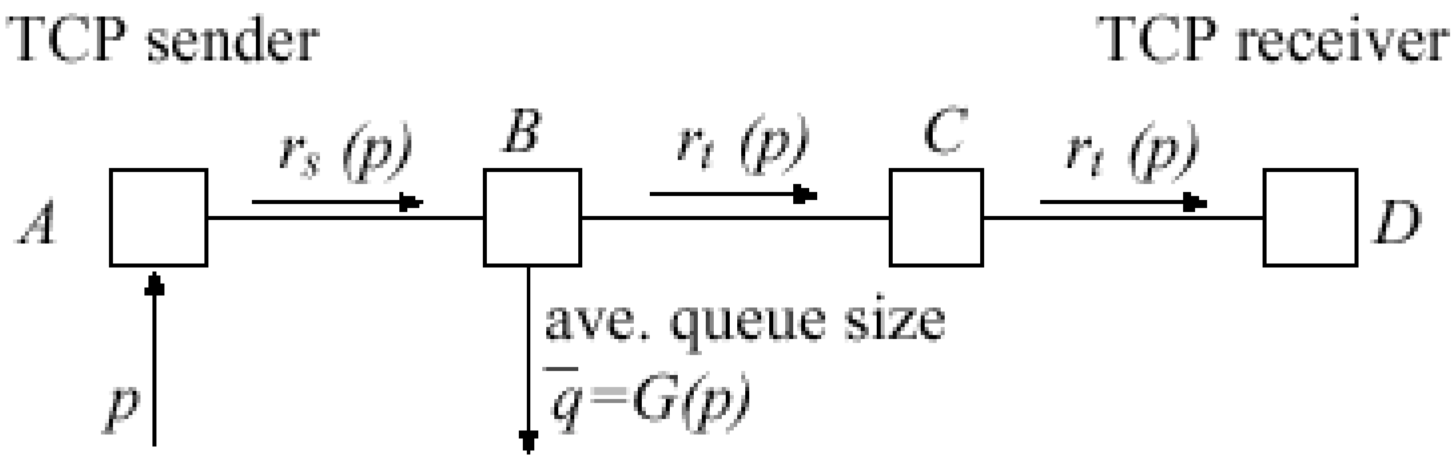


Fig. 3. An open control system with one TCP flow

Non Feedback Queue "Law"

$$R = R_0 + \bar{q}/c$$

$$p_0 = T^{-1}_p (c/n, R_0)$$

$$\bar{q}(p) = \{ \max (B, c (T^{-1}_R (p, c/n) - R_0)), p \le p_0$$

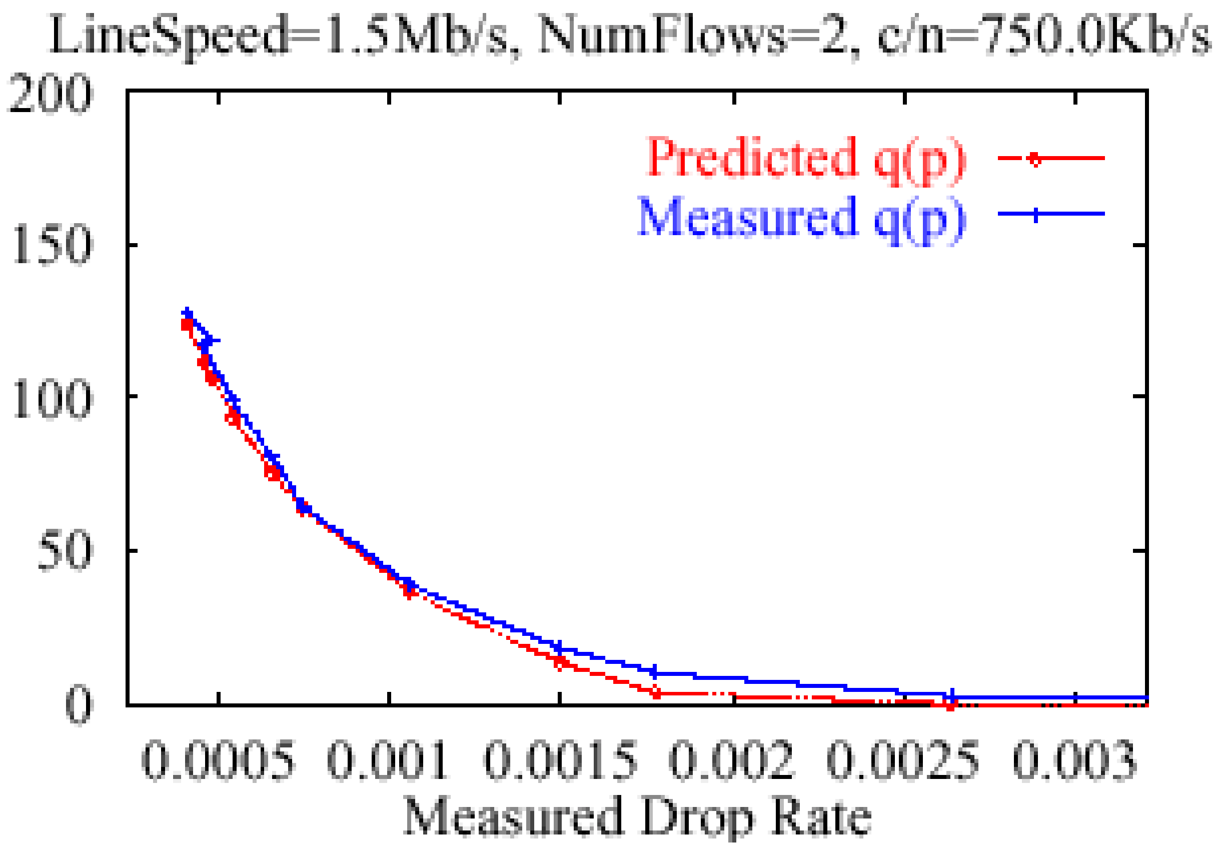
Else 0

$$u(p) = \{ 1, p \le p_0 \quad \text{Else } T(p, R_0) / (c/n)$$

Verification through simulation

- Using NS run multiple simulations varying link capacity, number of flows, and drop probability p
- Flows are “infinite” FTP sessions with fixed RTT
- Buffer is large enough to prevent packet loss due to overflow
- Graph mathematically predicted average queue size vs. simulation (and do the same with link utilization)

One Sample Result



Add in Feedback

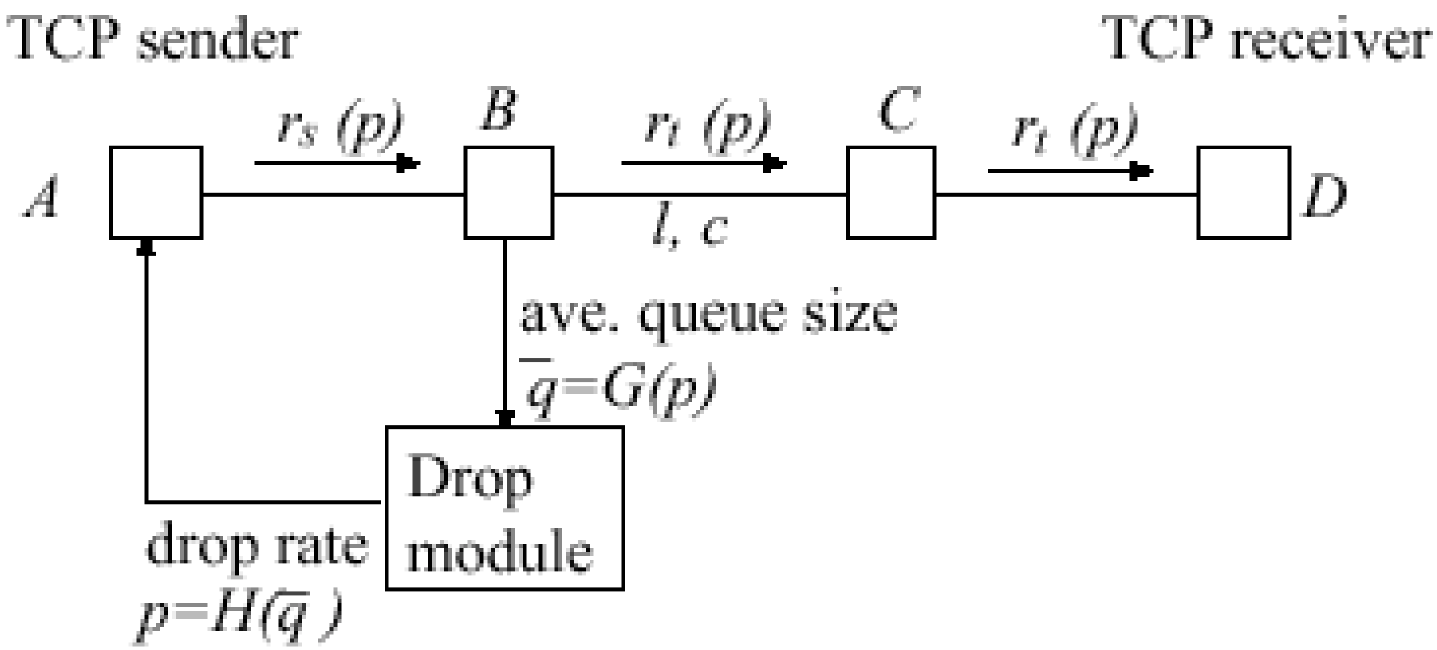
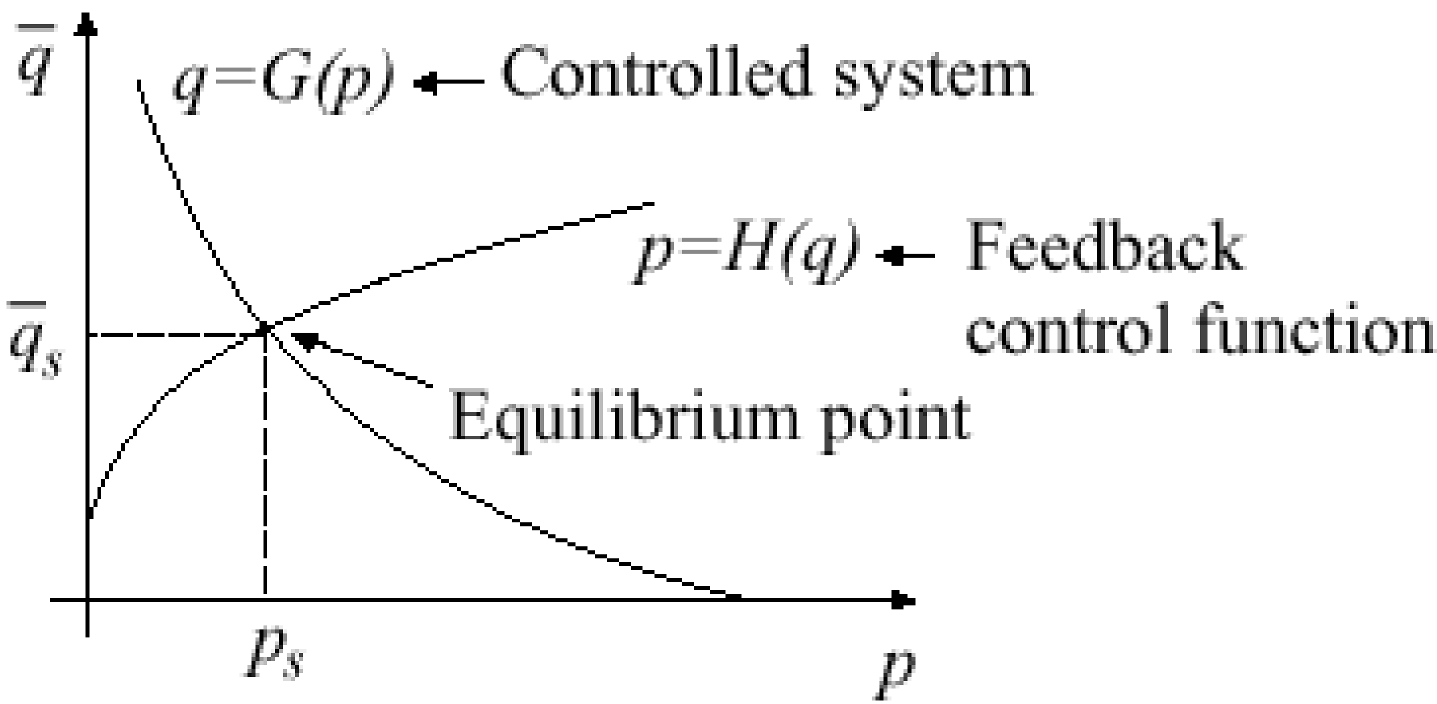


Fig. 2. A single-flow feedback control system

Feedback Control system Equilibrium point



RED as a Control Function

$$p = H(\overline{q}_e) = \begin{cases} 0, & 0 \leq \overline{q}_e < q_{min} \\ \frac{\overline{q}_e - q_{min}}{q_{max} - q_{min}} p_{max}, & q_{min} \leq \overline{q}_e < q_{max} \\ 1, & q_{max} \leq \overline{q}_e \leq B \end{cases}$$

Simulation with G(p) and H($\bar{q}$)

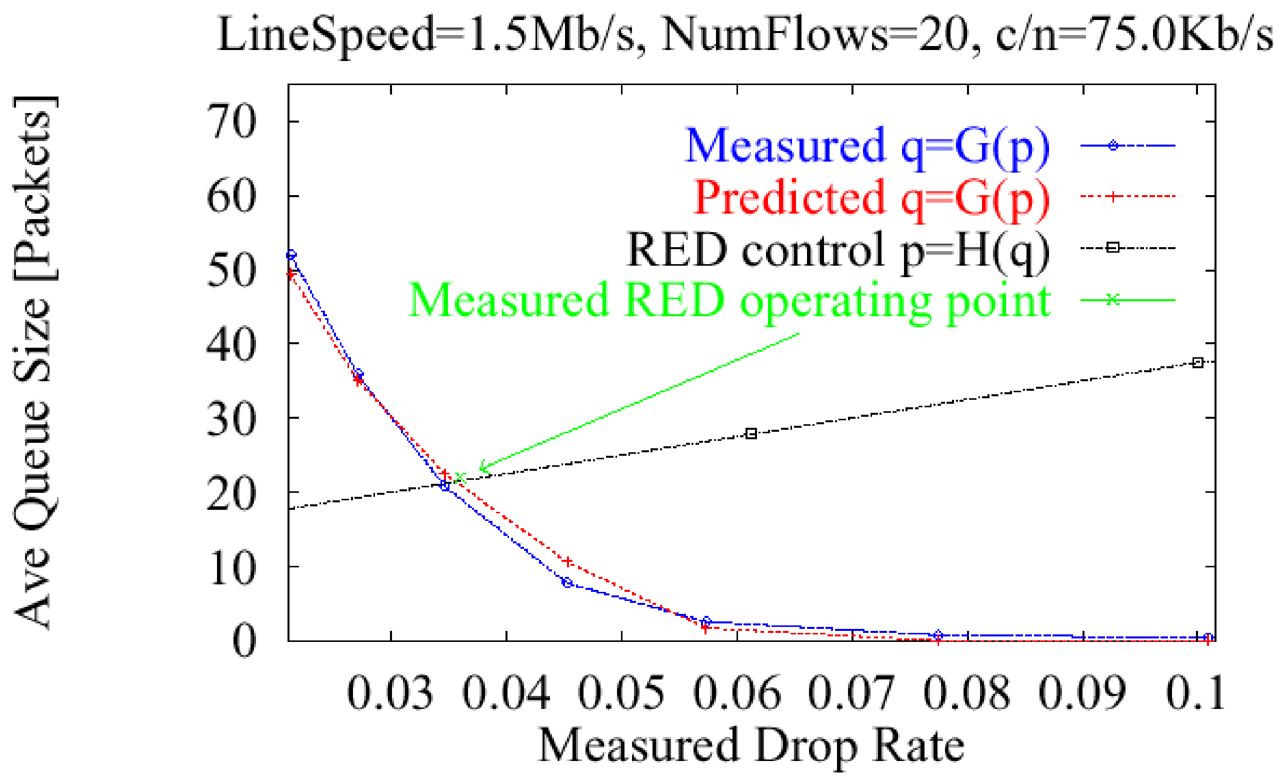
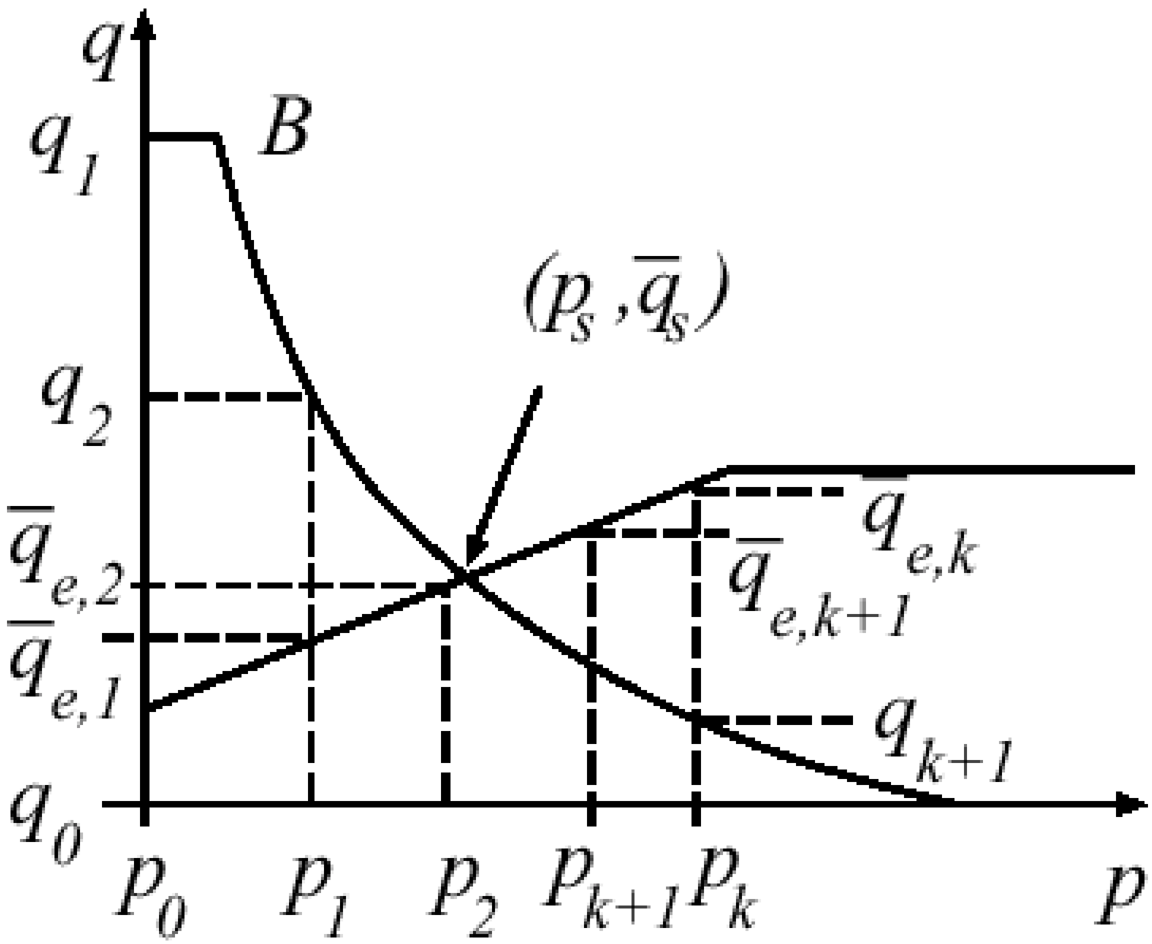


Fig. 8. RED average operating point: measured and predicted

RED convergence point



Stable system results

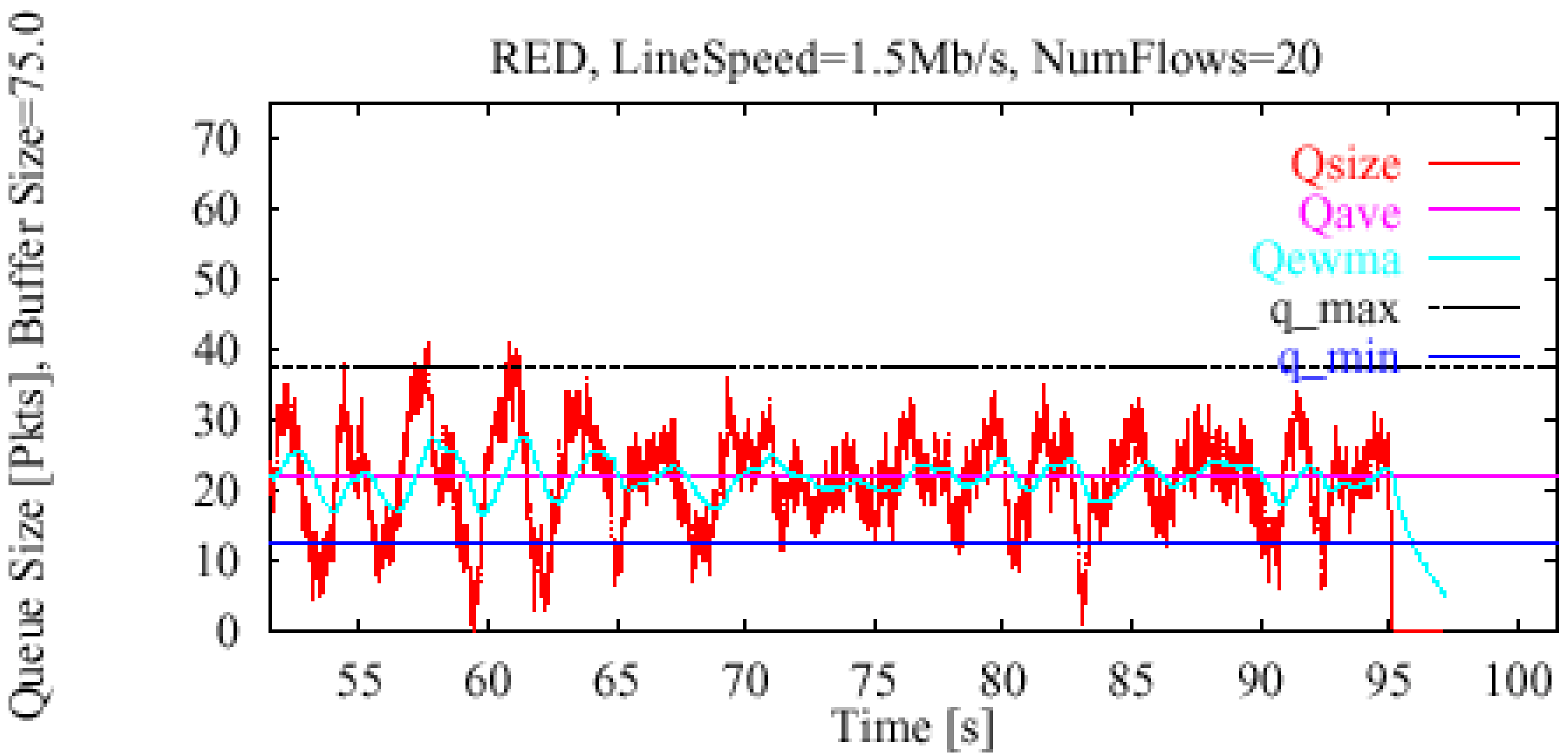


Fig. 10. Instantaneous and average queue size in time, converging case

Unstable results

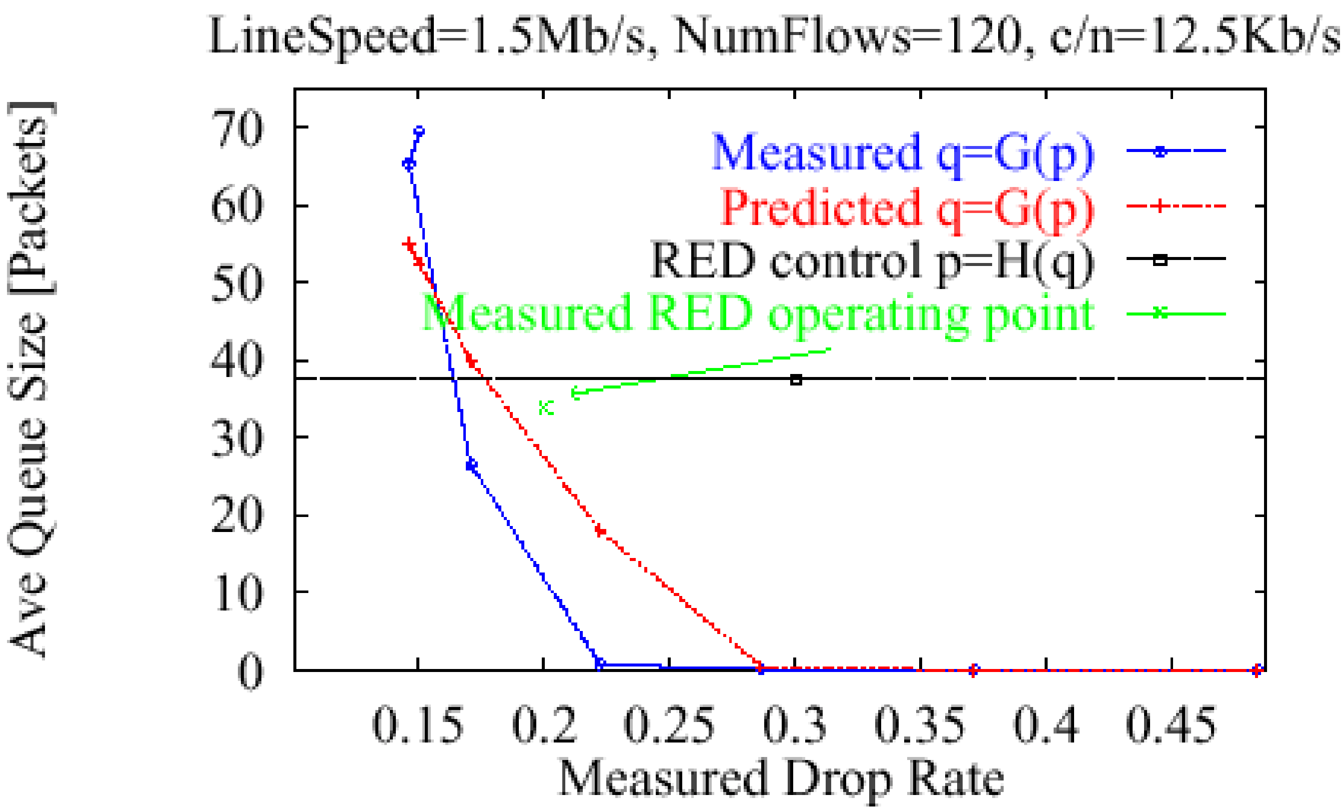


Fig. 11. RED average operating point situated beyond $p_{max} = 0.1$

Unstable results part 2

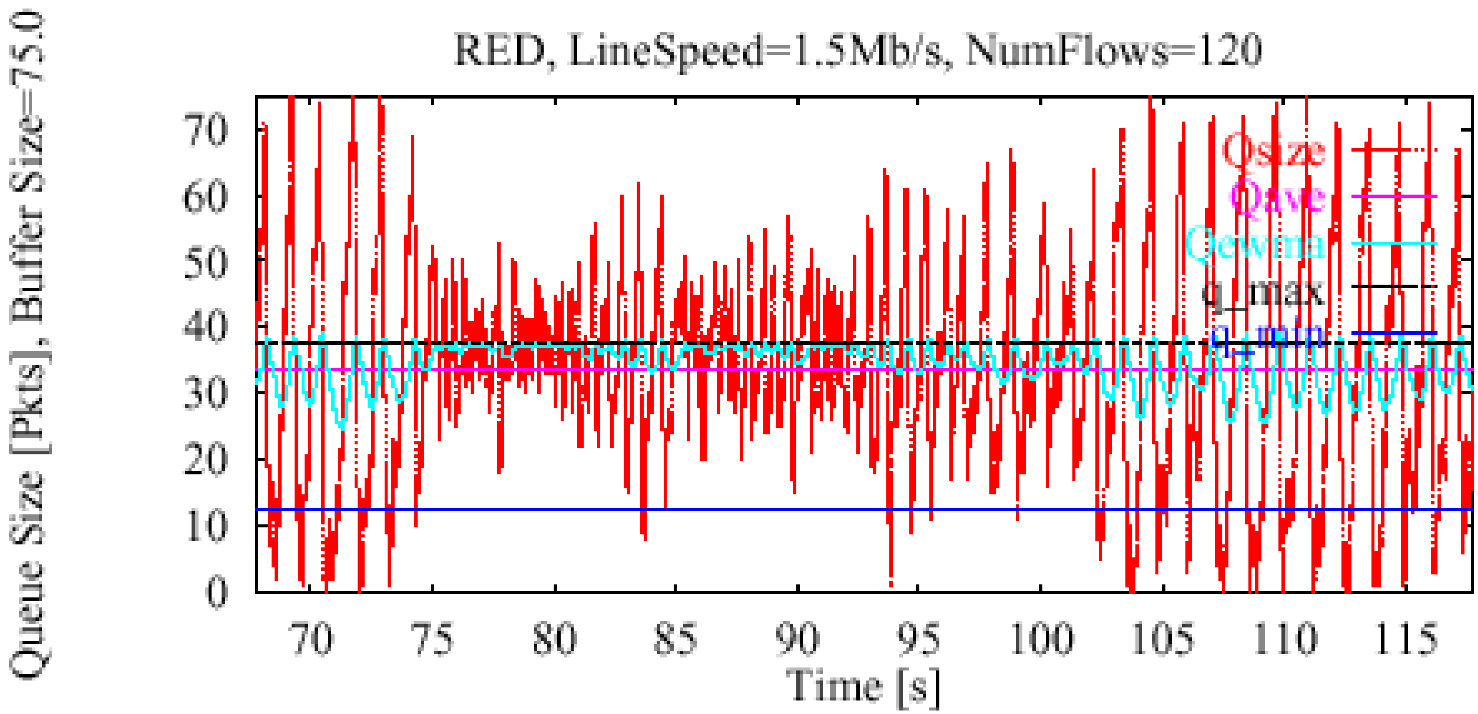


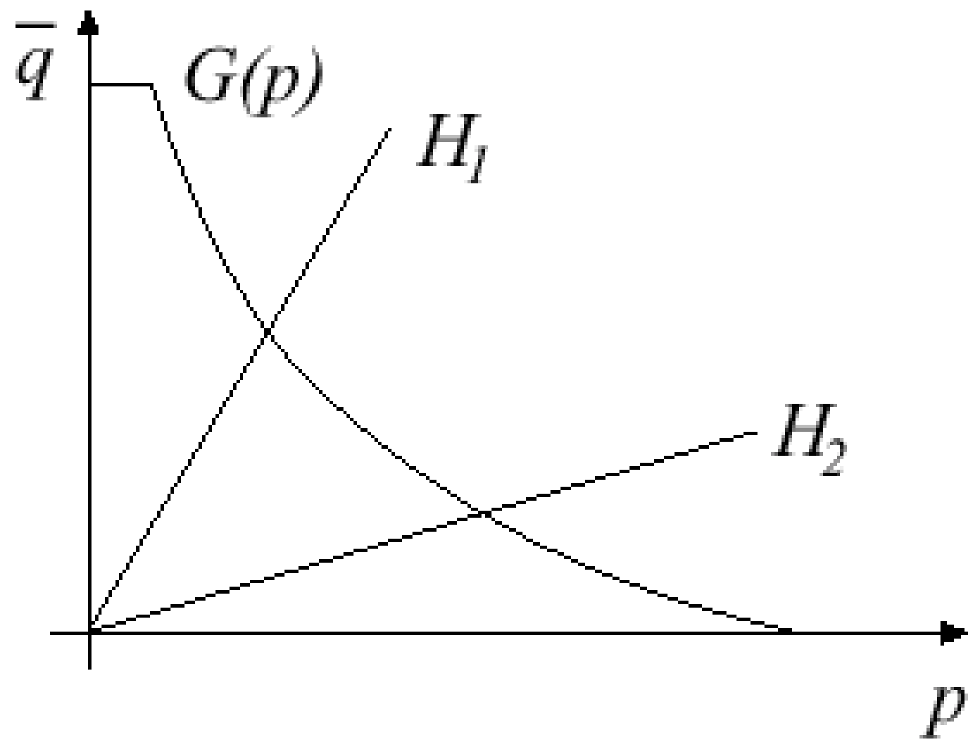
Fig. 13. Instantaneous and average queue size in time, oscillating case

RED configuration Recommendations

- **drop-conservative policy:** low p, high $\bar{q}$
- **delay-conservative policy:** low $\bar{q}$, high p
- Need to estimate:

1. Line speed *c*
2. Min and Max throughput per flow τ or number of flows *n*
3. Min and Max packet size M
4. Min and Max RRT R_0

Sample Control Law policy



Range of Queue Laws

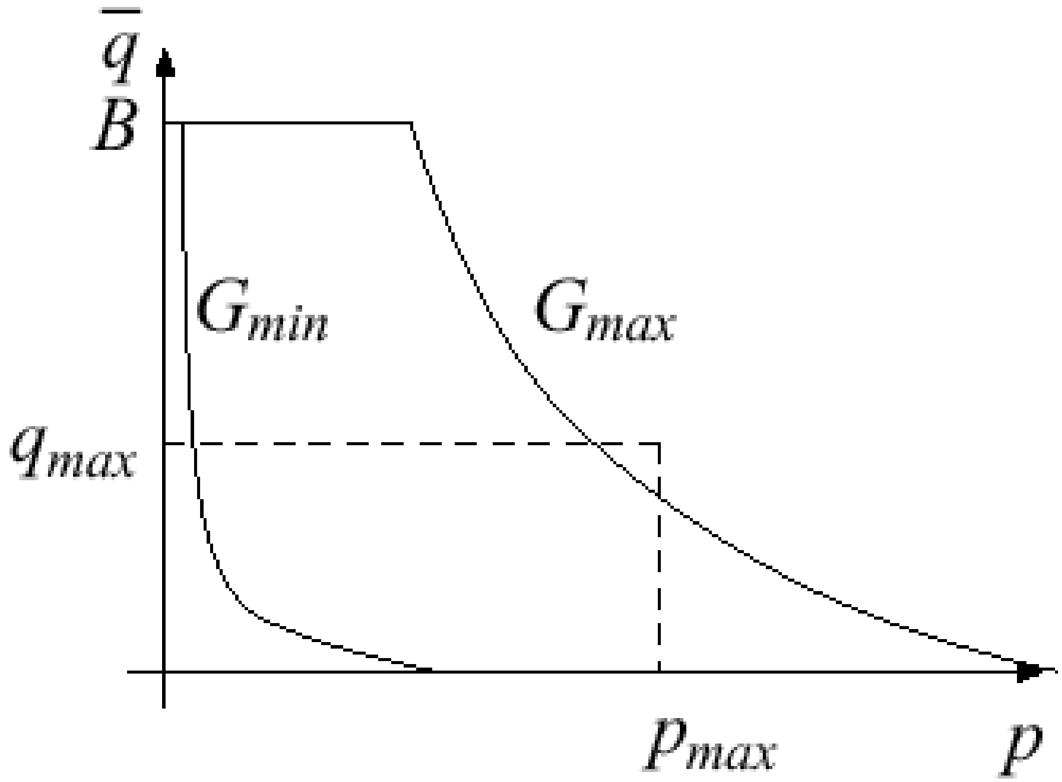


Fig. 15. Range of queue laws for a given queue system

Configuring Estimator of average queue Size

Consists of :

- Queue averaging algorithms
- Averaging interval
- Sampling the queue size

Queue Averaging Algorithm

- Low- pass filter on current queue size
- Moving average to filter out bursts
- Exponential weighting decreasing with age
- Estimate is computed over samples from the previous I time period – recommendations for I to follow

$$\text{Average weight} = w = 1 - a^{\delta/I}$$

Averaging Interval I

- Should provide good estimate of long term average assuming number of flows is constant
- Should adapt as fast as possible to change in traffic conditions

I = P is recommended

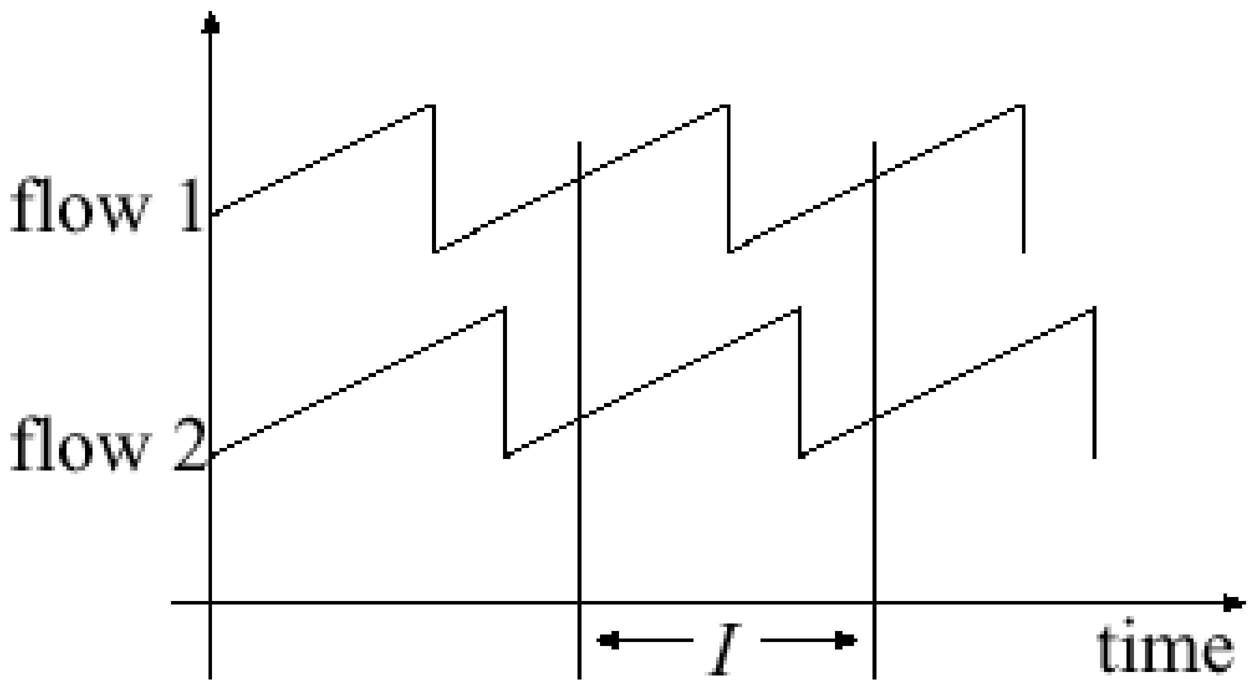


Fig. 18. Averaging two TCP flows

Sampling the Queue size

- Queue size acts like a step function
- Changes every RTT with adjustments made from information received
- "Ideal" sampling rate is once every RTT
- Recommend sampling = minimum RRT

Conclusions

- Feedback control model validated through simulations
- Found instability points and recommended settings to avoid them
- Also developed recommended RED queue size estimator settings
- Many issues still to look at in future

Thoughts

- Nice idea using model from a different discipline to analyze networks
- Good simulations to validate predicted data
- Many assumptions made to make math and model work which may make it invalid
- Limited traffic patterns and type of traffic also make the model's value suspect

Questions?