Stability Analysis of TCP/RED Communication Algorithms

Ljiljana Trajković
Simon Fraser University, Vancouver, Canada
ljilja@cs.sfu.ca

http://www.ensc.sfu.ca/~ljilja

Collaborators

Mingjian Liu and Hui Zhang
School of Engineering Science
Simon Fraser University, Vancouver, Canada

Alfredo Marciello and Mario di Bernardo
University of Naples Federico II, Naples, Italy

Xi Chen, Siu-Chung Wong, and Chi K. Tse
Department of Electronic and Information Engineering
The Hong Kong Polytechnic University, Hong Kong

Roadmap

- Introduction
- TCP/RED congestion control algorithms: an overview
- Discrete-time dynamical models of TCP Reno with RED
- Discontinuity-induced bifurcations in TCP/RED
- Stability analysis of RED gateway with multiple TCP Reno connections
- Stability study of TCP/RED system using detrended fluctuation analysis
- References

Motivation

- Modeling TCP Reno with RED is important to:
  - examine the interactions between TCP and RED
  - understand and predict the dynamical network behavior
  - analyze the impact of system parameters
  - investigate bifurcations and complex behavior
  - investigate stability of the TCP/RED system

TCP: Transmission Control Protocol

RED: Random Early Detection Gateways for Congestion Avoidance

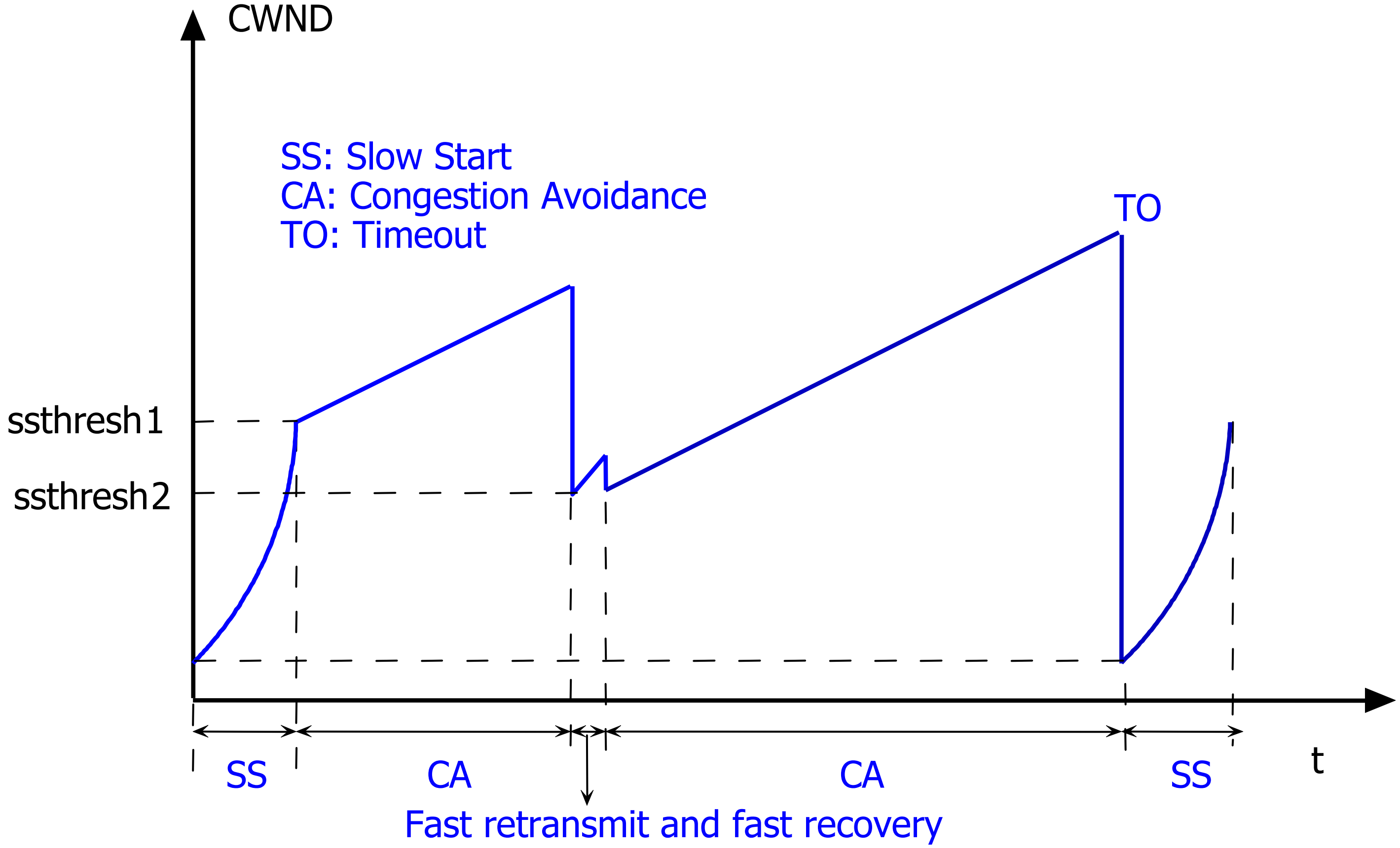
Roadmap

- Introduction
- TCP/RED congestion control algorithms: an overview
- Discrete-time dynamical models of TCP Reno with RED
- Discontinuity-induced bifurcations in TCP/RED
- Stability analysis of RED gateway with multiple TCP Reno connections
- Stability study of TCP/RED system using detrended fluctuation analysis
- References

TCP

- Several flavors of TCP:
  - Tahoe: 4.3 BSD Tahoe (~ 1988)
    - slow start, congestion avoidance, and fast retransmit (RFC 793, RFC 2001)
  - Reno: 4.3 BSD Reno (~ 1990)
    - slow start, congestion avoidance, fast retransmit, and fast recovery (RFC 2001, RFC 2581)
  - NewReno (~ 1996)
    - new fast recovery algorithm (RFC 2582)
  - SACK (~ 1996, RFC 2018)

TCP Reno


CWND
SS: Slow Start
CA: Congestion Avoidance
TO: Timeout
TO
ssthresh1
ssthresh2
t
SS
CA
CA
SS
Fast retransmit and fast recovery

TCP Reno: slow start and congestion avoidance

- Slow start:
  - *cwnd* = *IW* (1 or 2 packets)
  - when *cwnd* < *ssthresh*

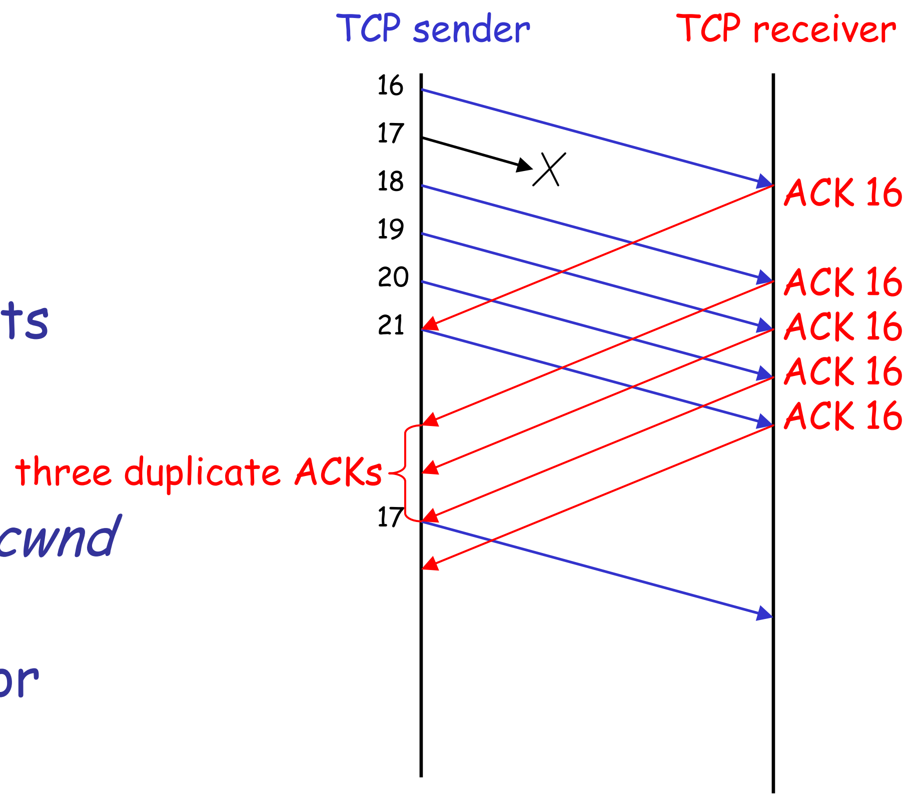
    cwnd = *cwnd* + 1 for each received *ACK*
- Congestion avoidance:
  - when *cwnd* > *ssthresh*

    cwnd = *cwnd* + 1/*cwnd* for each *ACK*

cwnd : congestion window size
IW : initial window size
ssthresh : slow start threshold
ACK : acknowledgement
RTT : round trip time

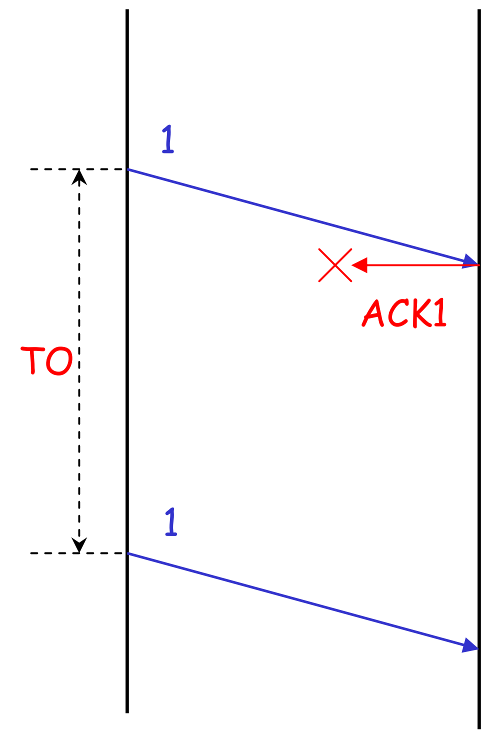
TCP Reno: fast retransmit and fast recovery

- three duplicate *ACKs* are received
- retransmit the packet
- *ssthresh* = *cwnd*/2, *cwnd* = *ssthresh* + 3 packets
- *cwnd* = *cwnd* + 1, for each additional duplicate ACK
- transmit the new data, if *cwnd* allows
- *cwnd* = *ssthresh*, if ACK for new data is received



TCP Reno: timeout

- TCP maintains a retransmission timer
- The duration of the timer is called retransmission timeout
- Timeout occurs when the ACK for the delivered data is not received before the retransmission timer expires
- TCP sender retransmits the lost packet
- *ssthresh* = *cwnd*/2
  cwnd = 1 or 2 packets



AQM: Active Queue Management

- AQM (RFC 2309):
  - reduces bursty packet drops in routers
  - provides lower-delay interactive service
  - avoids the "lock-out" problem
  - reacts to the incipient congestion before buffers overflow
- AQM algorithms:
  - RED (RFC 2309)
  - ARED, CHOKe, BLUE, …

RED

- Random Early Detection Gateways for Congestion Avoidance
  - Proposed by S. Floyd and V. Jacobson, LBN, 1993:

    S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Trans. Networking*,
    vol. 1, no. 4, pp. 397–413, Aug. 1993.
- Main concept:
  - drop packets before the queue becomes full

RED variables and parameters

- Main variables and parameters:
  - average queue size: $\overline{q}_{k+1}$
  - instantaneous queue size: q_{k+1}
  - drop probability: p_{k+1}
  - queue weight: w_q
  - maximum drop probability: $p_{\max}$
  - queue thresholds: $q_{\min}$ and $q_{\max}$

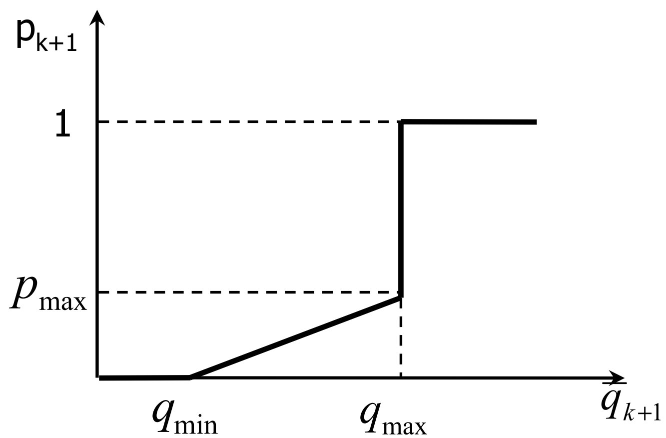
RED algorithm

Calculate:

- average queue size for each packet arrival

$$\bar{q}_{k+1} = (1 - w_q) \cdot \bar{q}_k + w_q \cdot q_{k+1}$$

- drop probability



RED algorithm: drop probability

- if $(q_{\min} < \bar{q}_{k+1} < q_{\max})$

$$p_{k+1} = \frac{\bar{q}_{k+1} - q_{\min}}{q_{\max} - q_{\min}} p_{\max}$$

- else if $(\bar{q}_{k+1} \geq q_{\max})$

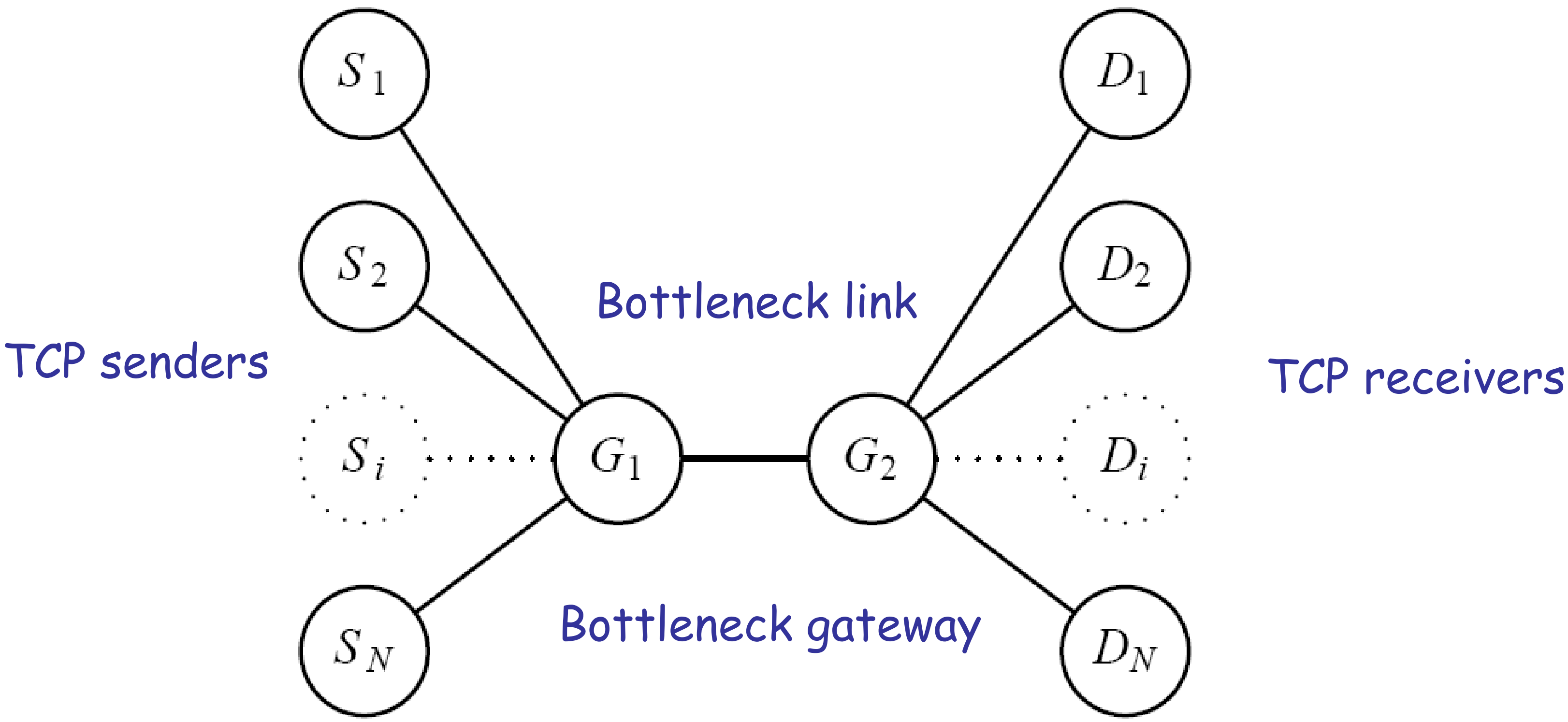
$$p_{k+1} = 1$$

- else $(\bar{q}_{k+1} \leq q_{\min})$

$$p_{k+1} = 0$$

- mark or drop the arriving packet with probability p_{k+1}

Network model



TCP window congestion control algorithm

Sender sends *W* packets at a time

Window size = *W*

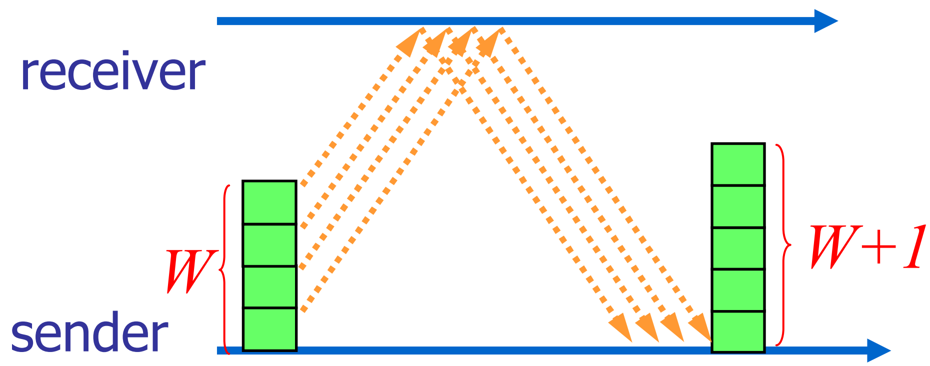
- Additive increase (AI):
  if no loss, window size increases by one per round trip time
- Multiplicative decrease (MD):
  on detection of loss, window size decreases by half

TCP window congestion control algorithm

Sender sends *W* packets at a time

Window size = *W*

- **Additive increase (AI):** if no loss, window size increases by one per round trip time
- **Multiplicative decrease (MD):** on detection of loss, window size decreases by half

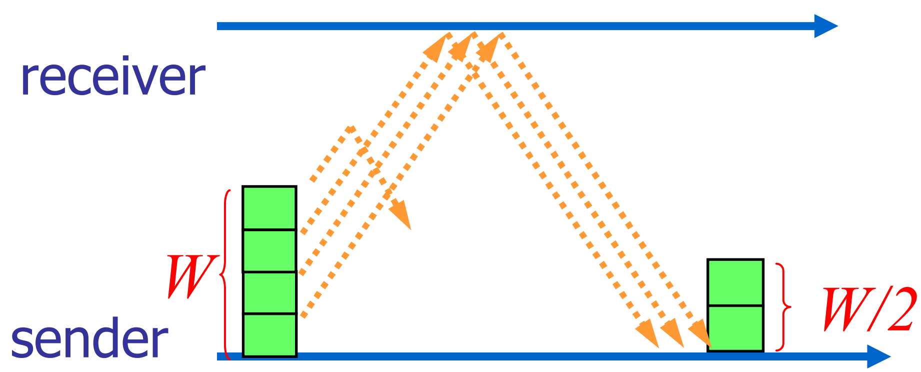


TCP window congestion control algorithm

Sender sends *W* packets at a time

Window size = *W*

- Additive increase (AI): if no loss, window size increases by one per round trip time
- Multiplicative decrease (MD): on detection of loss, window size decreases by half



RED algorithm

Average queue length: $x_k = (1-\alpha)x_{k-1} + \alpha q_k$

α: queue averaging weight $0<\alpha<1$

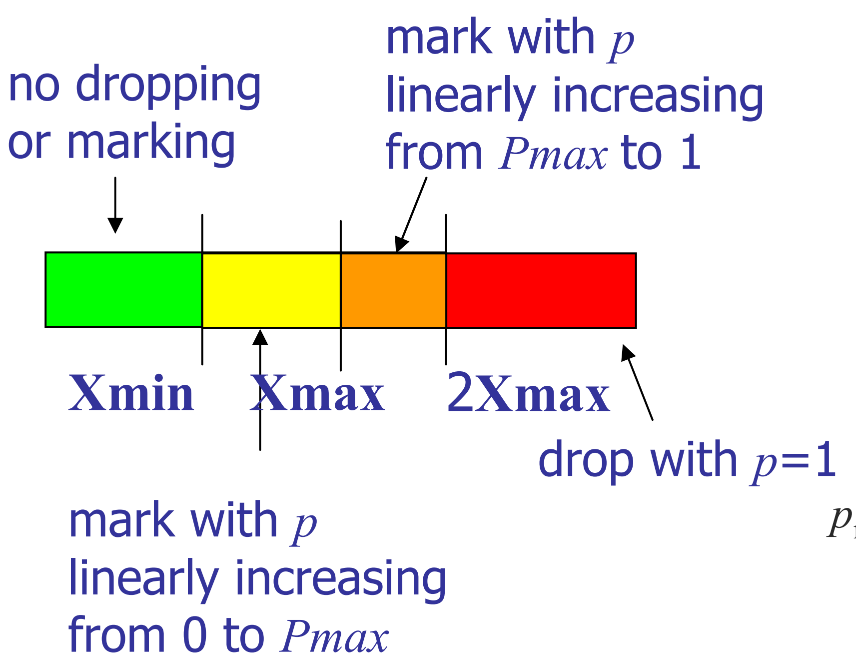
q_k: :current queue size

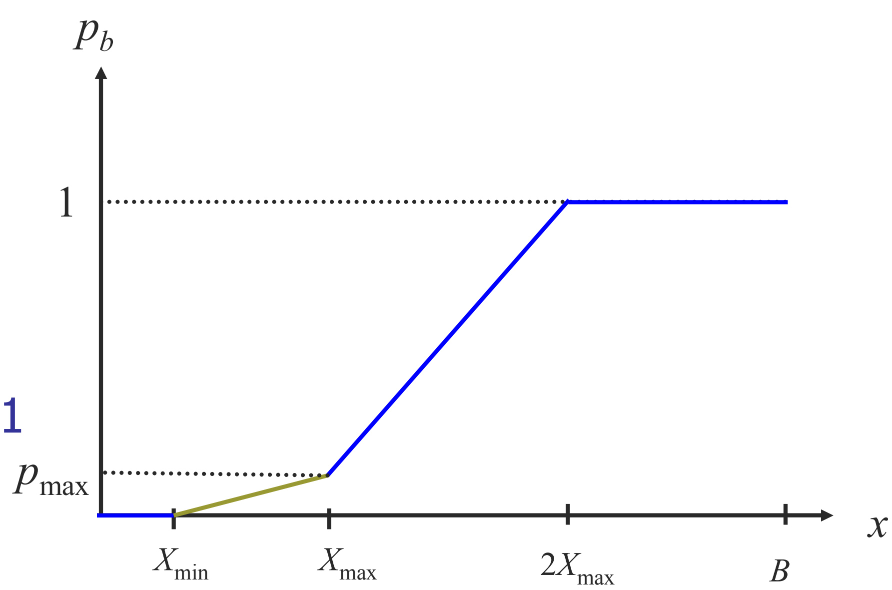
Marking/dropping probability:

$$p_b = \begin{cases} 0 & 0 \le x_k < X_{\min} \\ \dfrac{x_k - X_{\min}}{X_{\max} - X_{\min}} p_{\max} & X_{\min} \le x_k \le X_{\max} \\ p_{\max} - \dfrac{x_k - X_{\max}}{X_{\max} - X_{\min}}(1 - p_{\max}) & X_{\max} < x_k \le 2X_{\max} \\ 1 & 2X_{\max} < x_k \le B \end{cases}$$

$$p_k = \frac{p_b}{1 - c_m p_b}$$

RED marking/dropping probability

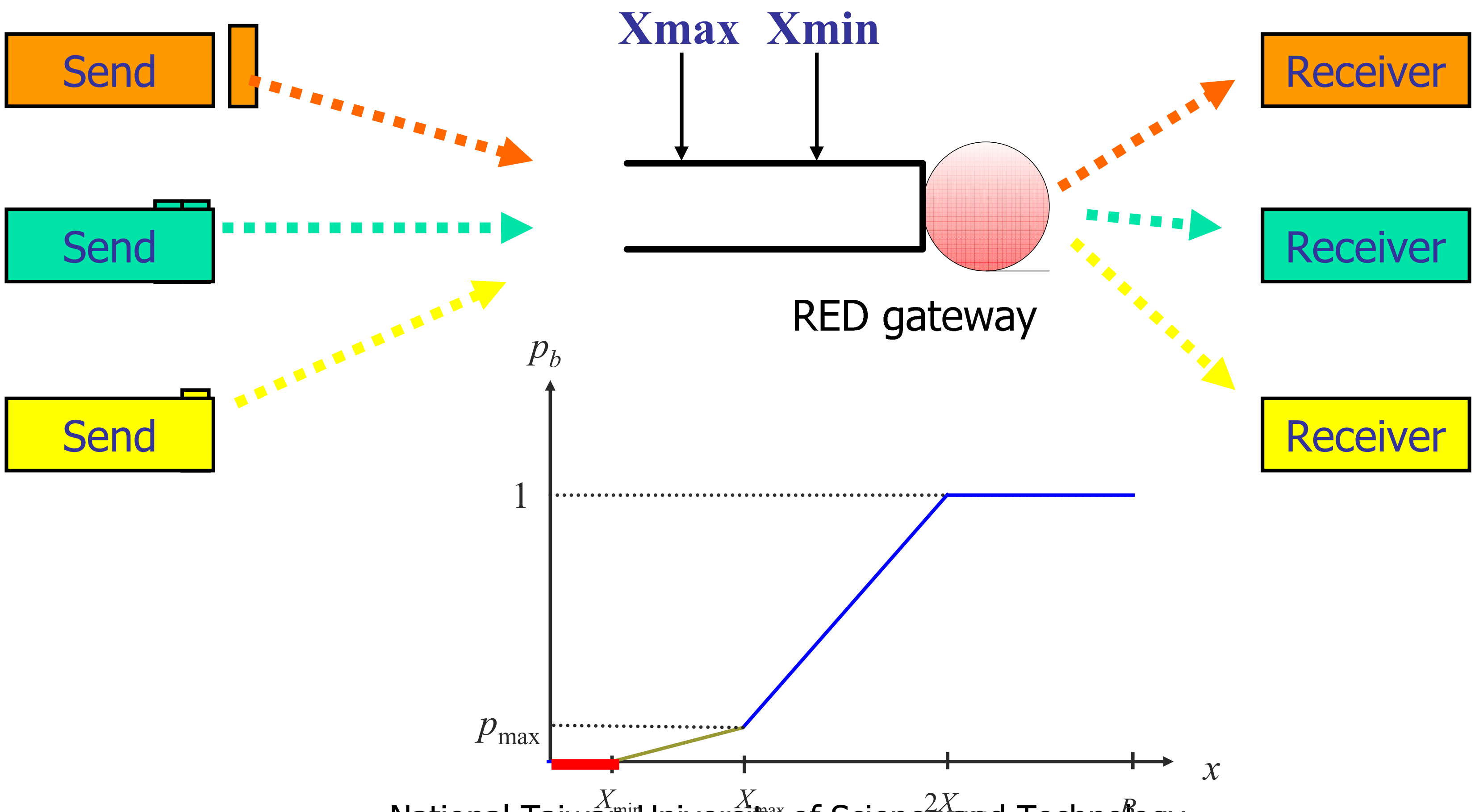




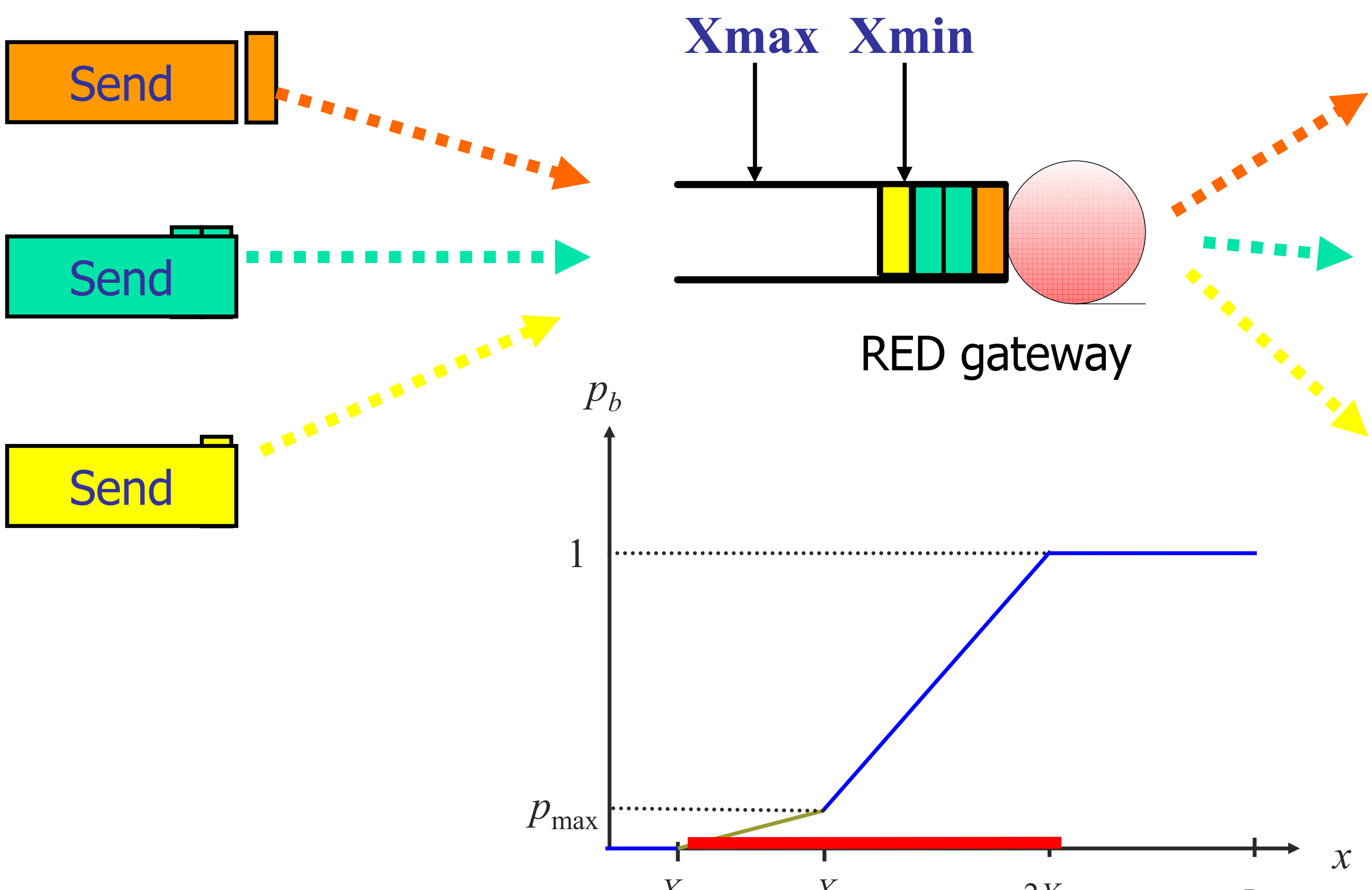
average queue length

drop probability *p*

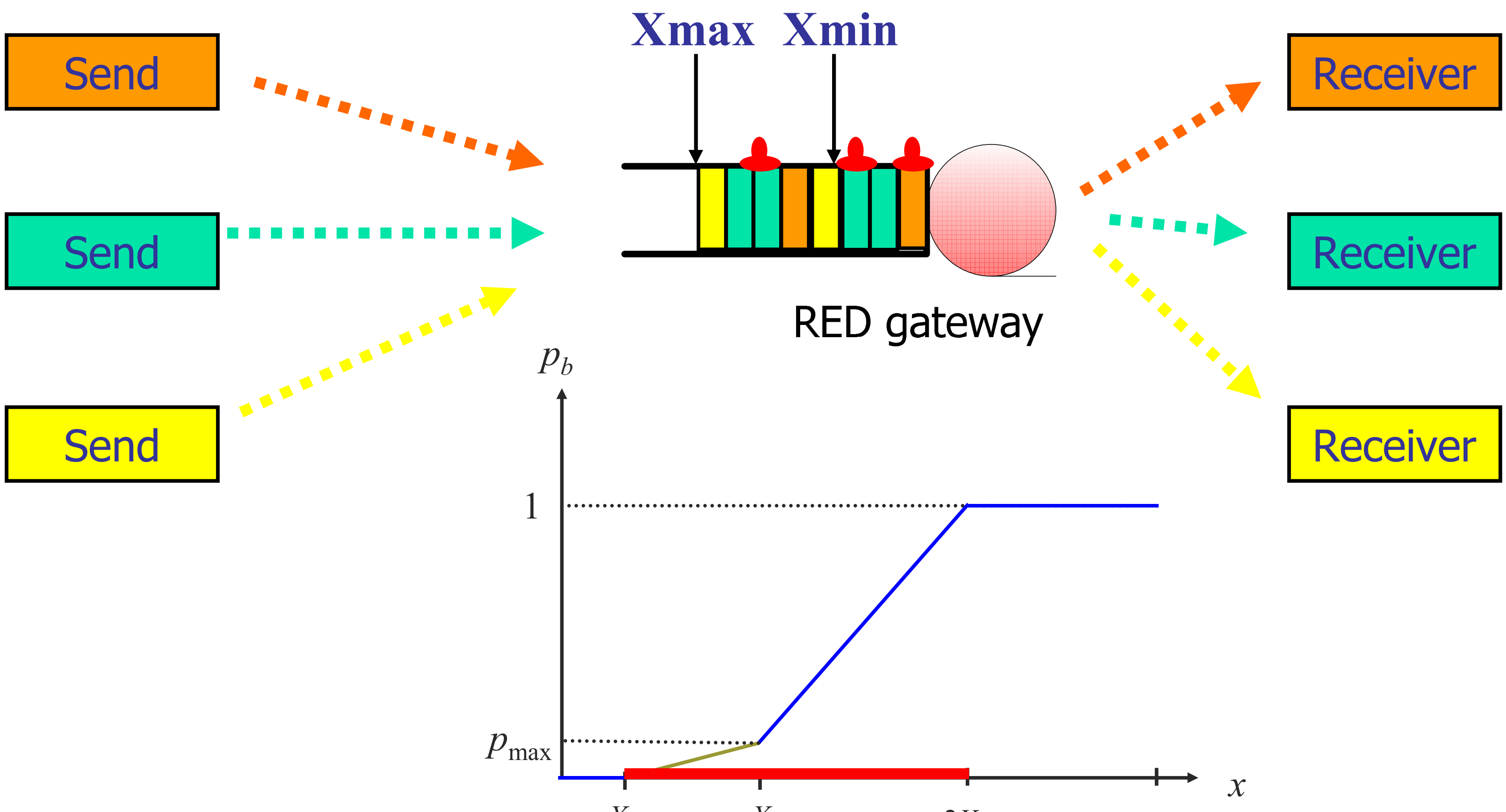
RED gateway: small queue length



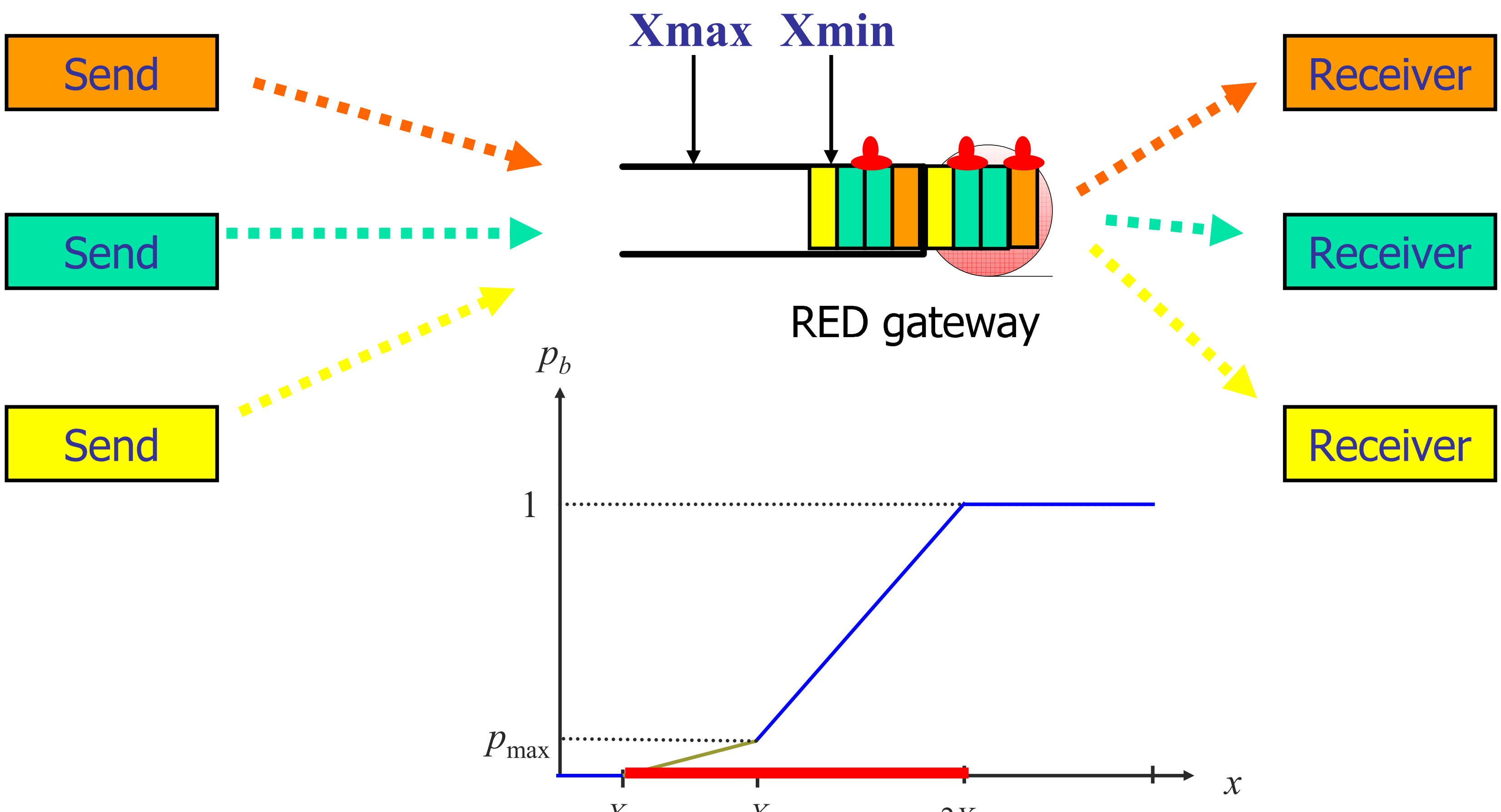
RED gateway: small queue length



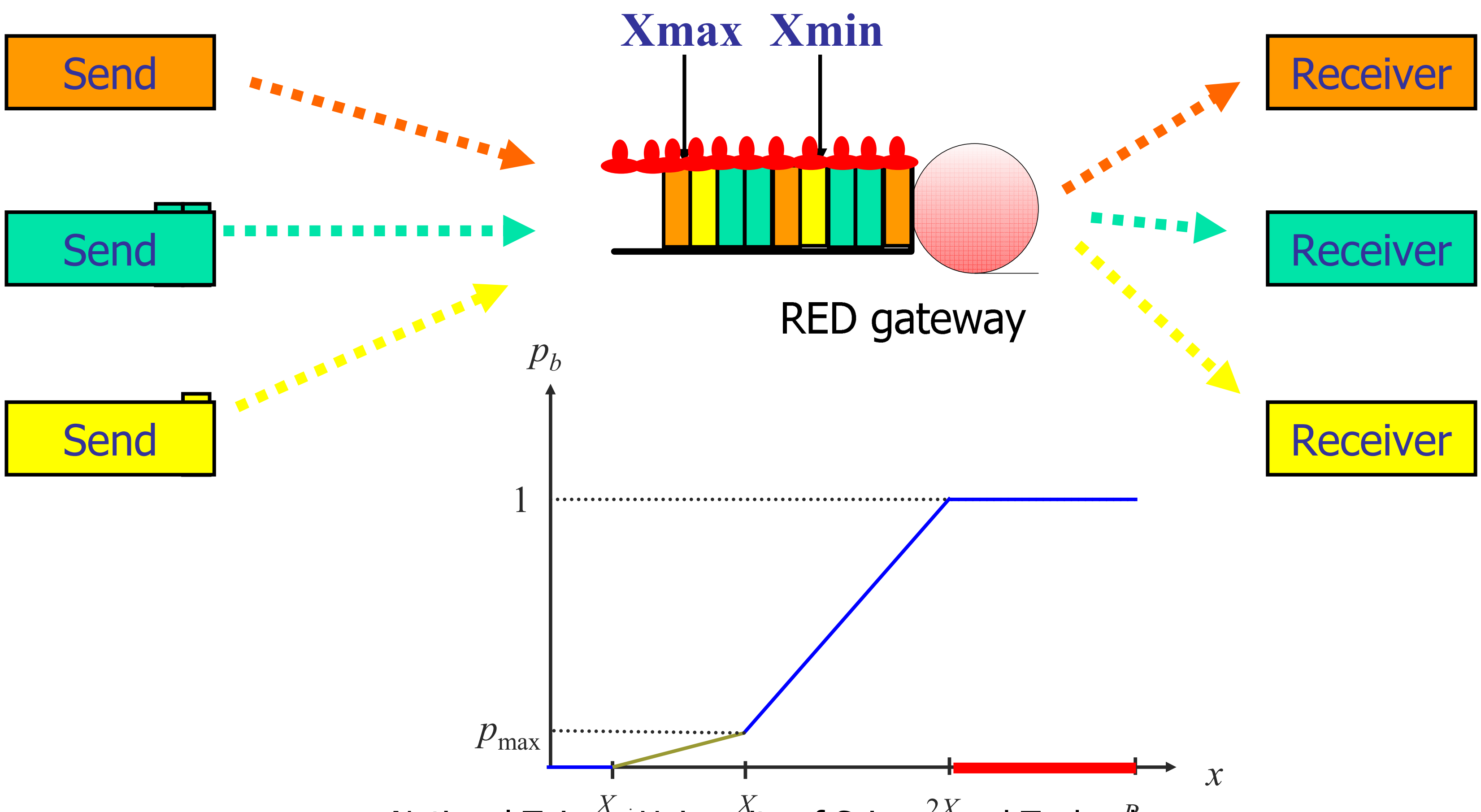
RED gateway: target queue length



RED gateway: target queue length



RED gateway: large queue length



Roadmap

- Introduction
- TCP/RED congestion control algorithms: an overview
- Discrete-time dynamical models of TCP Reno with RED
- Discontinuity-induced bifurcations in TCP/RED
- Stability analysis of RED gateway with multiple TCP Reno connections
- Stability study of TCP/RED system using detrended fluctuation analysis
- References

Modeling methodology

- Categories of TCP models:
  - averaged and discrete-time models
  - short-lived and long-lived TCP connections
- TCP/RED model:
  - discrete-time model with a long-lived connection
- State variables:
  - window size (TCP)
  - average queue size (RED)

TCP/RED model

- Key properties of the proposed TCP/RED model:
  - slow start, congestion avoidance, fast retransmit, and fast recovery (simplified)
  - Timeout:

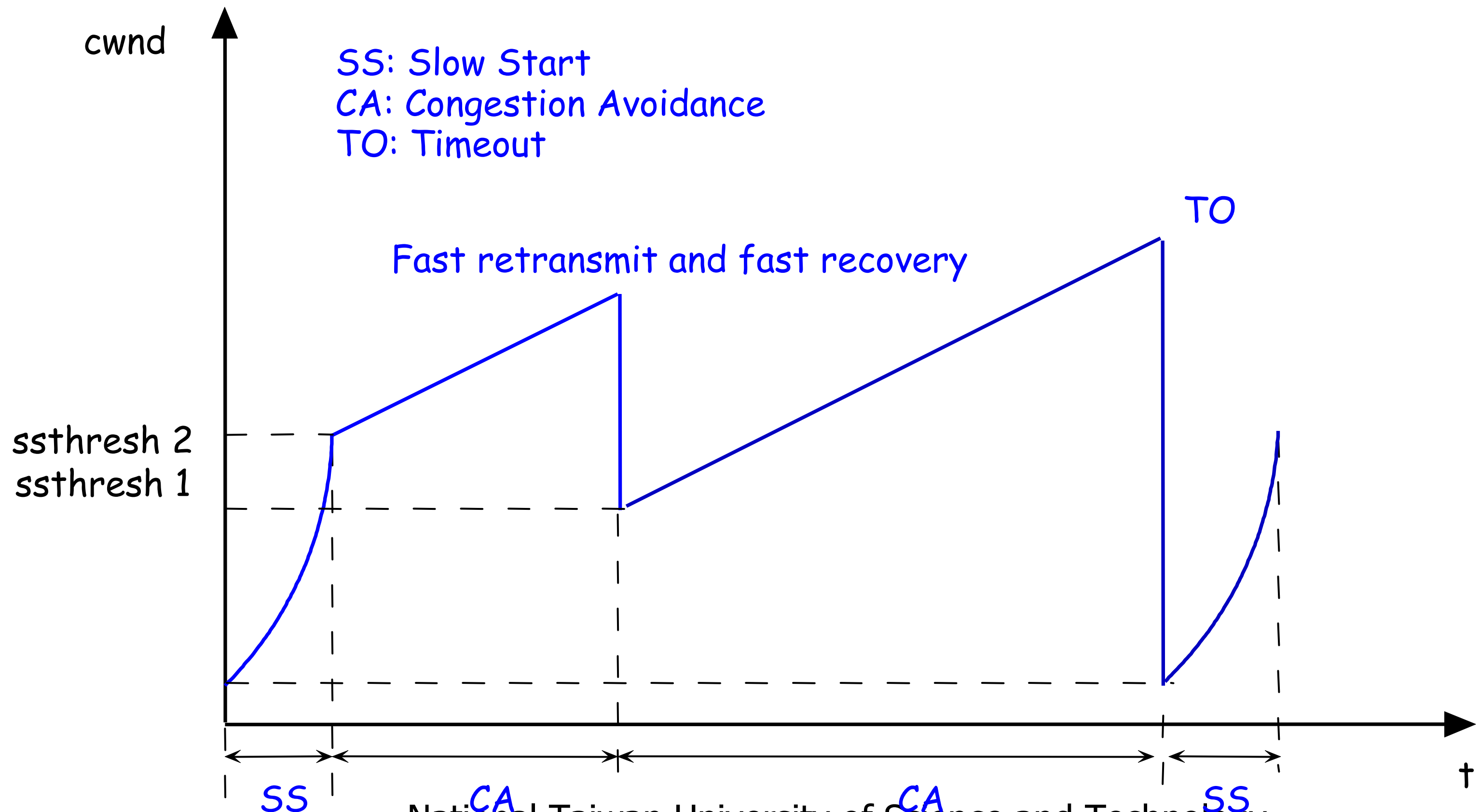
    J. Padhye, V. Firoiu, and D. F. Towsley, "Modeling TCP Reno performance: a simple model and its empirical validation," *IEEE/ACM Trans. Networking*, vol. 8, no. 2, pp. 133–145, Apr. 2000.
  - Captures the basic RED algorithm

Assumptions

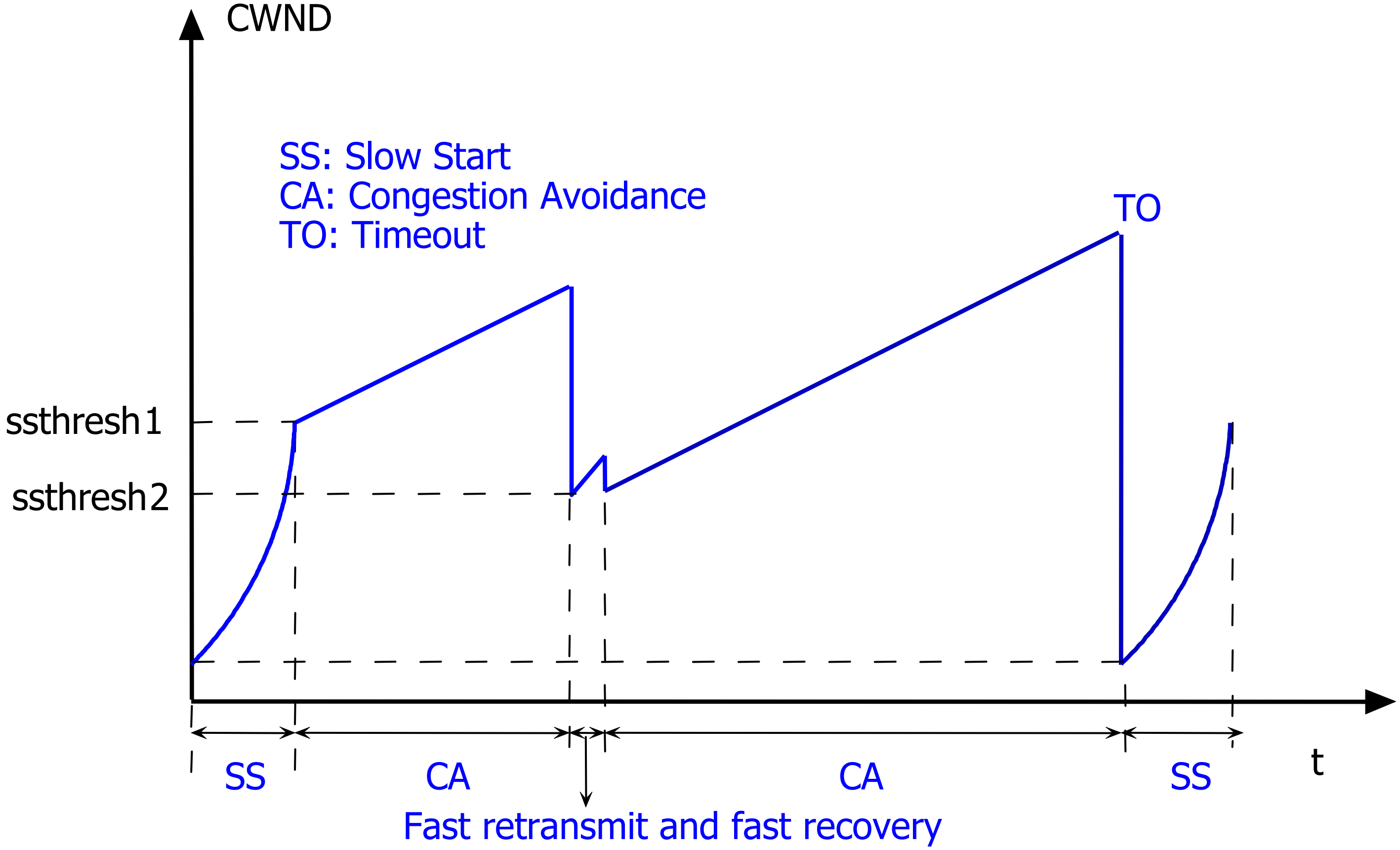
- long-lived TCP connection
- constant propagation delay between the source and the destination
- constant packet size
- ACK packets are never lost
- timeout occurs only due to packet loss
- the system is sampled at the end of every RTT interval

TCP/RED model simplifications

- Simplified fast recovery



TCP Reno: fast recovery



TCP/RED model simplifications

- TO = 5 RTT

  V. Firoiu and M. Borden, "A study of active queue management for congestion control," in *Proc. of IEEE INFOCOM 2000*, vol. 3, pp. 1435-1444, Tel-Aviv, Israel, Mar. 2000.

- RED: parameter count is not used

$$\text{if } (q_{\min} < \bar{q} < q_{\max})$$

$$p_b = p_{\max} \times \frac{\bar{q} - q_{\min}}{q_{\max} - q_{\min}}$$

$$p_a = \frac{p_b}{1 - count \times p_b}$$

$$\xrightarrow{p_a = p_b}$$

$$\text{if } (q_{\min} < \bar{q} < q_{\max})$$

$$p_a = p_{\max} \times \frac{\bar{q} - q_{\min}}{q_{\max} - q_{\min}}$$

Simple S-TCP/RED model

- M-model, a discrete nonlinear dynamical model of TCP Reno with RED:
  - P. Ranjan, E. H. Abed, and R. J. La, "Nonlinear instabilities in TCP-RED," in *Proc. IEEE INFOCOM 2002*, New York, NY, USA, June 2002, vol. 1, pp. 249–258 and *IEEE/ACM Trans. on Networking*, vol. 12, no. 6, pp. 1079–1092, Dec. 2004.
- One state variable: average queue size
- The proposed TCP/RED model is:
  - simple and intuitively derived
  - able to capture detailed dynamical behavior of TCP/RED systems
  - has been verified via ns-2 simulations

Simple S-TCP/RED model: state variable and parameters

- Variables:
  - $\overline{q}_{k+1}$: average queue size in round k+1
  - $\overline{q}_k$: average queue size in round k
  - w_q: queue weight in RED
  - N: number of TCP connections
  - K: constant = $\sqrt{3/2}$
  - p_k: drop probability in round k
  - C: capacity of the link between the two routers
  - d: round-trip propagation delay
  - M: packet size
  - *rwnd*: receiver's advertised window size

Simple S-TCP/RED model: case 1

- Drop probability: $p_k \neq 0$

$$q_{k+1} = q_k + B(p_k) \cdot RTT_{k+1} \cdot N - \frac{C \cdot RTT_{k+1}}{M}$$

$$= q_k + \frac{K}{\sqrt{p_k} \cdot RTT_{k+1}} \cdot RTT_{k+1} \cdot N - \frac{C}{M}(d + \frac{q_k \cdot M}{C})$$

$$= \frac{K \cdot N}{\sqrt{p_k}} - \frac{C \cdot d}{M}$$

where:

$B(p_k)$: TCP sending rate

$B(p_k) \cdot RTT_{k+1} \cdot N$: the number of incoming packets

$C \cdot \frac{RTT_{k+1}}{M}$: the number of outgoing packets

Simple S-TCP/RED model: case 1

The average queue size is:

$$\overline{q}_{k+1} = (1 - w_q) \cdot \overline{q}_k + w_q \cdot q_{k+1}$$

hence

$$\overline{q}_{k+1} = (1 - w_q) \cdot \overline{q}_k + w_q \cdot \max\left(\frac{N \cdot K}{\sqrt{p_k}} - \frac{C \cdot d}{M}, 0\right)$$

Simple S-TCP/RED model: case 2

- Drop probability: $p_k = 0$

$$q_{k+1} = q_k + B(p_k)\cdot RTT_{k+1}\cdot N - \frac{C\cdot RTT_{k+1}}{M}$$

$$= q_k + \frac{rwnd}{RTT_{k+1}}\cdot RTT_{k+1}\cdot N - \frac{C}{M}(d + \frac{q_k \cdot M}{C})$$

$$= rwnd \cdot N - \frac{C\cdot d}{M}$$

The average queue size is:

hence $$\bar{q}_{k+1} = (1-w_q)\cdot \bar{q}_k + w_q \cdot q_{k+1}$$

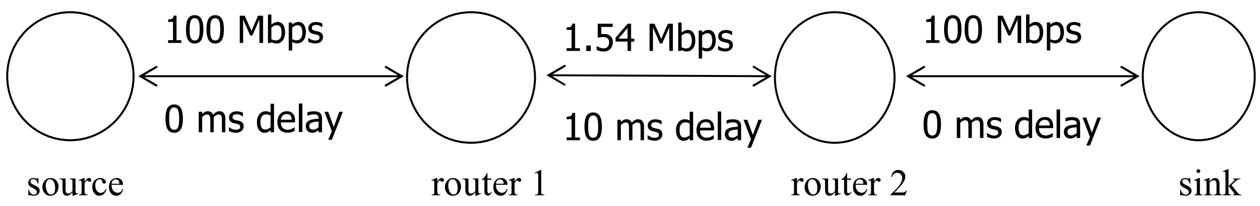
$$\bar{q}_{k+1} = (1-w_q)\cdot \bar{q}_k + w_q \cdot (rwnd\cdot N - \frac{C\cdot d}{M})$$

Simple S-TCP/RED model

- Dynamical model of TCP/RED:

$$\bar{q}_{k+1} = \begin{cases} (1-w_q)\cdot \bar{q}_k + w_q \cdot \max\left(\dfrac{N\cdot K}{\sqrt{p_k}} - \dfrac{C\cdot d}{M}, 0\right) \quad if\ p_k \neq 0 \\ (1-w_q)\cdot \bar{q}_k + w_q \cdot \left(rwnd \cdot N - \dfrac{C\cdot d}{M}\right) \quad if\ p_k = 0 \end{cases}$$

Validation: simulation scenario



- source to router1:
  - link capacity: 100 Mbps with 0 ms delay
- router 1 to router 2: the only bottleneck in the network
  - link capacity: 1.54 Mbps with 10 ms delay
- router 2 to sink:
  - link capacity: 100 Mbps with 0 ms delay

RED: default parameters

- RED parameters:

  S. Floyd, "RED: Discussions of Setting Parameters," Nov. 1997: http://www.icir.org/floyd/REDparameters.txt

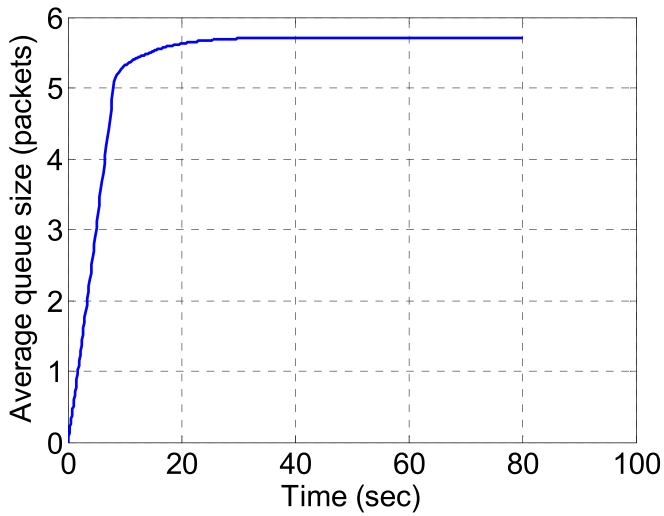
Queue weight (w_q)	0.002
Maximum drop probability (p_{max})	0.1
Minimum queue threshold (q_{min})	5 (packets)
Maximum queue threshold (q_{max})	15 (packets)
Packet size (M)	4,000 (bytes)

TCP/RED model validation

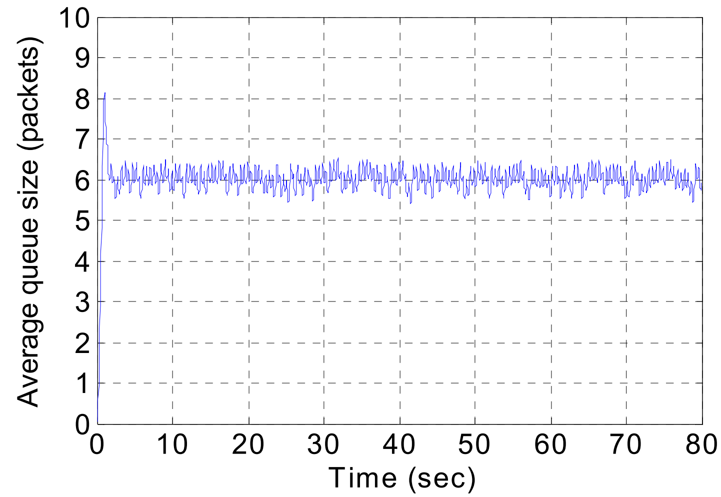
- Waveforms of the state variable with default parameters:
  - average queue size

- Validation for various values of the system parameters:
  - queue weight: w_q
  - maximum drop probability: p_{max}
  - queue thresholds: q_{min} and q_{max} , $q_{max}/q_{min} = 3$

Average queue size: waveforms



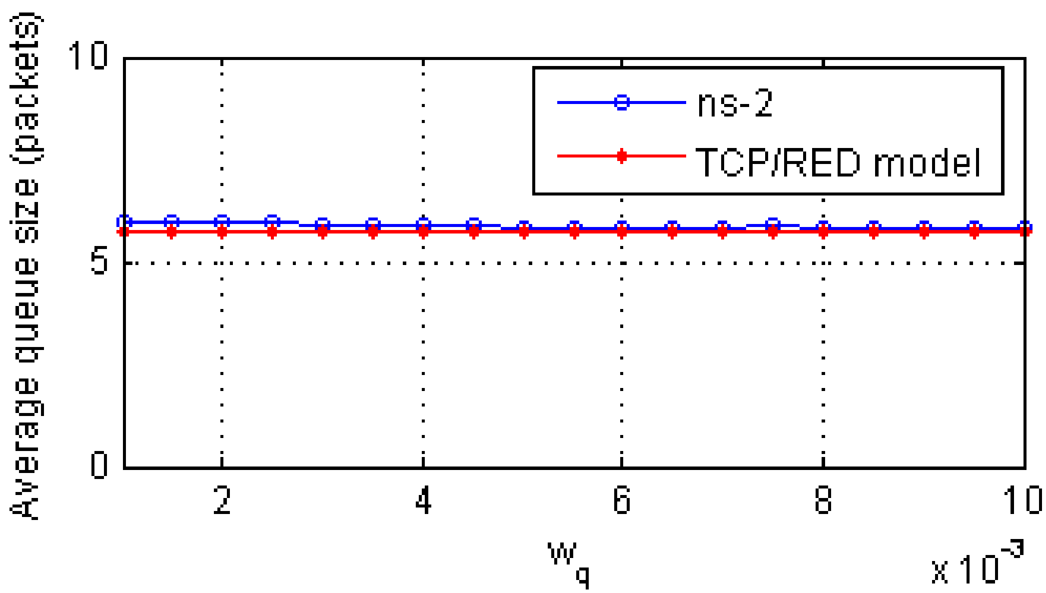
TCP/RED model



ns-2

Model validation: w_q

- average queue size during steady state:



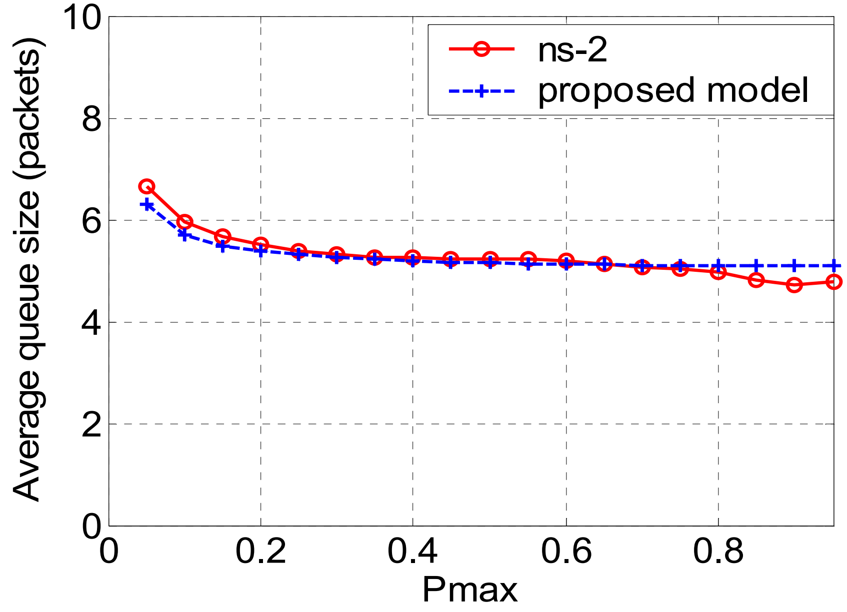
Model validation: w_q

- Comparison of system variables:

Parameter	Average RTT (msec)		Sending rate (packets/sec)		Drop rate (%)	
weight (w_q)	TCP/RED model	ns-2	TCP/RED model	ns-2	TCP/RED model	ns-2
0.001	164.6	36.1	385.6950	384.710	0.0421	0.543
0.002	143.8	36.0	385.7317	384.767	0.0356	0.546
0.004	137.1	36.2	385.3205	384.789	0.0486	0.556
0.006	135.2	35.8	385.4833	384.726	0.0486	0.556
0.008	134.7	35.8	385.5207	384.676	0.0486	0.549
0.01	134.6	35.7	385.5913	384.700	0.0483	0.546

Model validation: p_{max}

- average queue size during steady state:



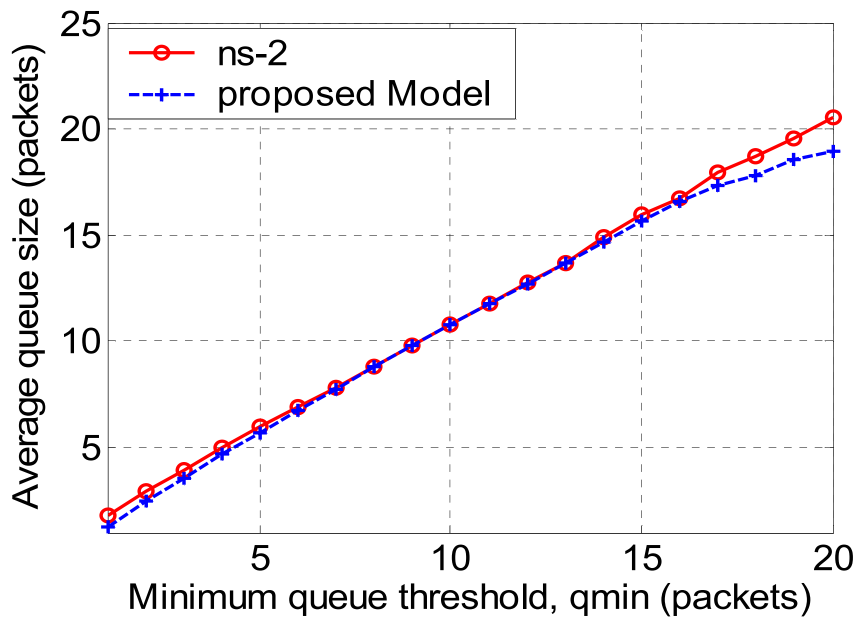
Model validation: p_{max}

- Comparison of system variables:

Parameter	Average RTT (msec)		Sending rate (packets/sec)		Drop rate (%)	
p_{max}	TCP/RED model	ns-2	TCP/RED model	ns-2	TCP/RED model	ns-2
0.05	161.7	38.1	385.7815	384.700	0.0323	0.510
0.1	143.8	36.0	385.6950	384.767	0.0421	0.546
0.25	131.7	34.5	385.4579	384.726	0.0518	0.585
0.5	126.9	34.0	385.5830	379.367	0.0551	0.613
0.75	125.9	35.1	385.3998	357.550	0.0572	0.647

Model validation: q_{min} and q_{max}

- average queue size during steady state:



Model validation: q_{min} and q_{max}

- Comparison of system variables:

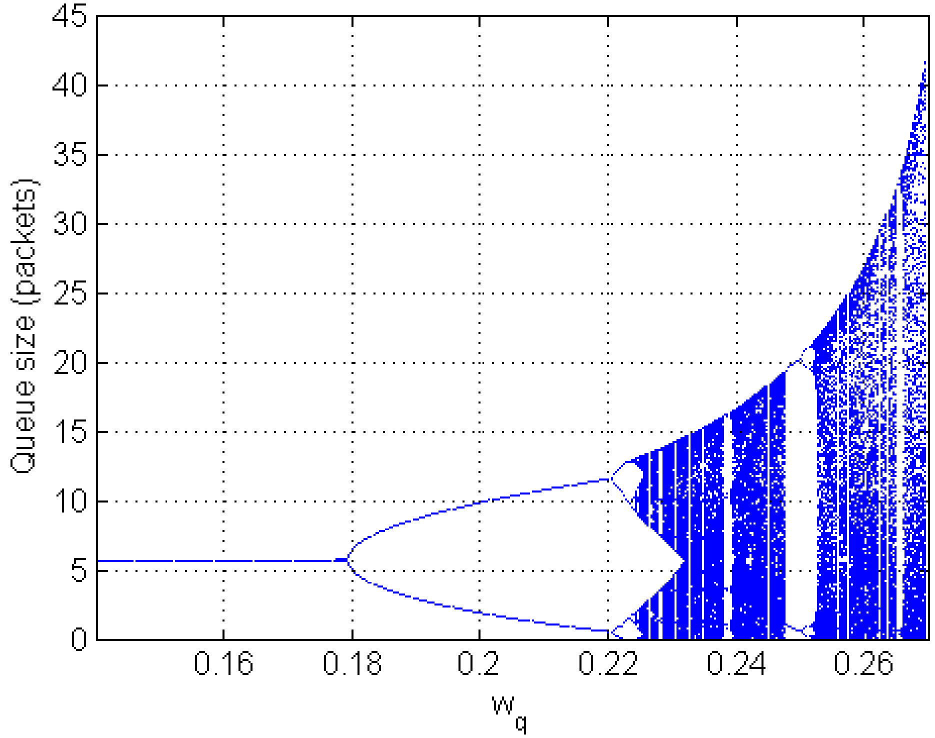
	Average RTT (msec)		Sending rate (packets/sec)		Drop rate (%)	
q_{min} (packets)	TCP/RED model	ns-2	TCP/RED model	ns-2	TCP/RED model	ns-2
3	103.6	31.1	385.2706	382.437	0.0875	0.709
5	143.8	36.0	385.6950	384.767	0.4210	0.546
10	238.8	48.1	385.7833	384.850	0.1300	0.331
15	307.9	60.3	386.9869	384.830	0.3216	0.224
20	343.8	73.0	387.9538	384.950	0	0.159

TCP/RED: model evaluation

- Waveforms of the average queue size:
  - match the ns-2 simulation results
- Sending rate:
  - reasonable agreement with ns-2 simulation results
- Average RTT and drop rate:
  - disagreement with ns-2 simulation results

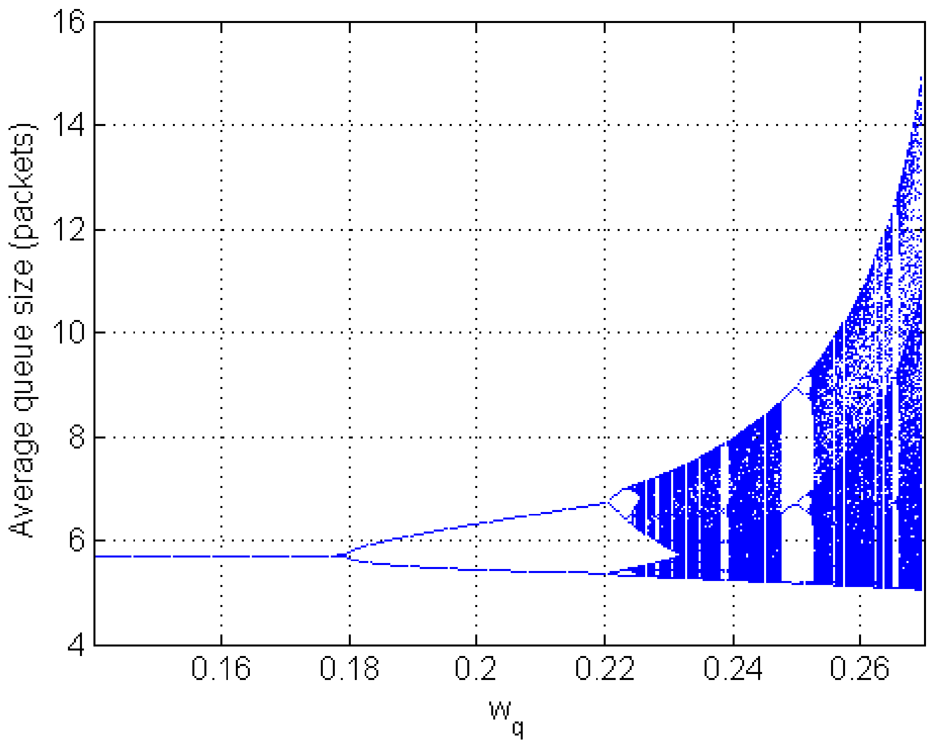
Queue size vs. w_q

- $p_{max} = 0.1$, $q_{min} = 5$, $q_{max} = 15$



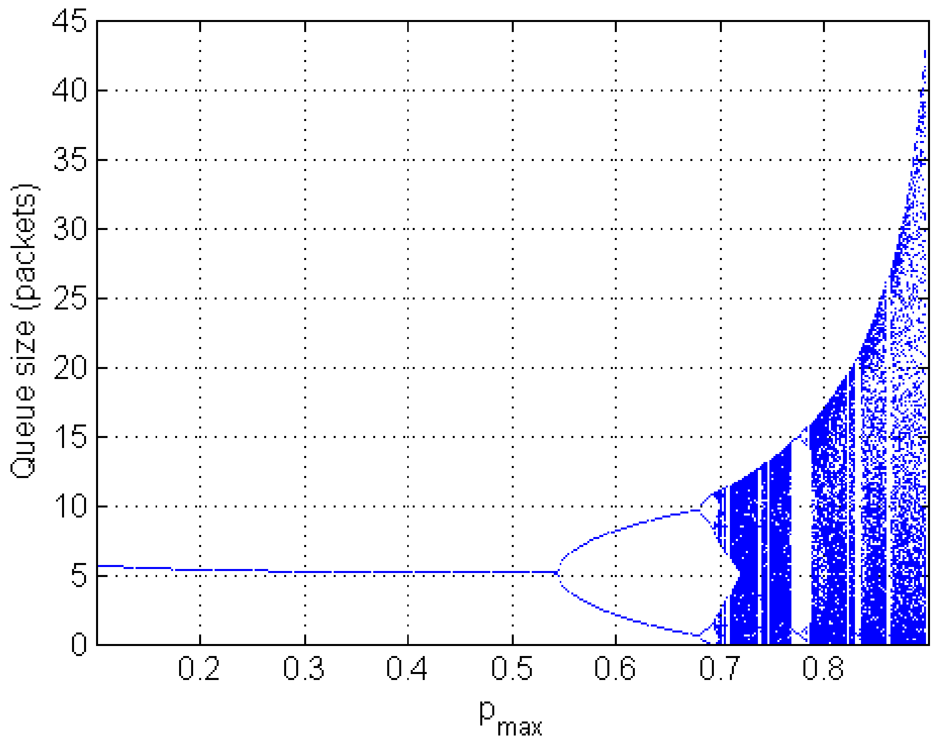
Average queue size vs. w_q

- $p_{max} = 0.1$, $q_{min} = 5$, $q_{max} = 15$



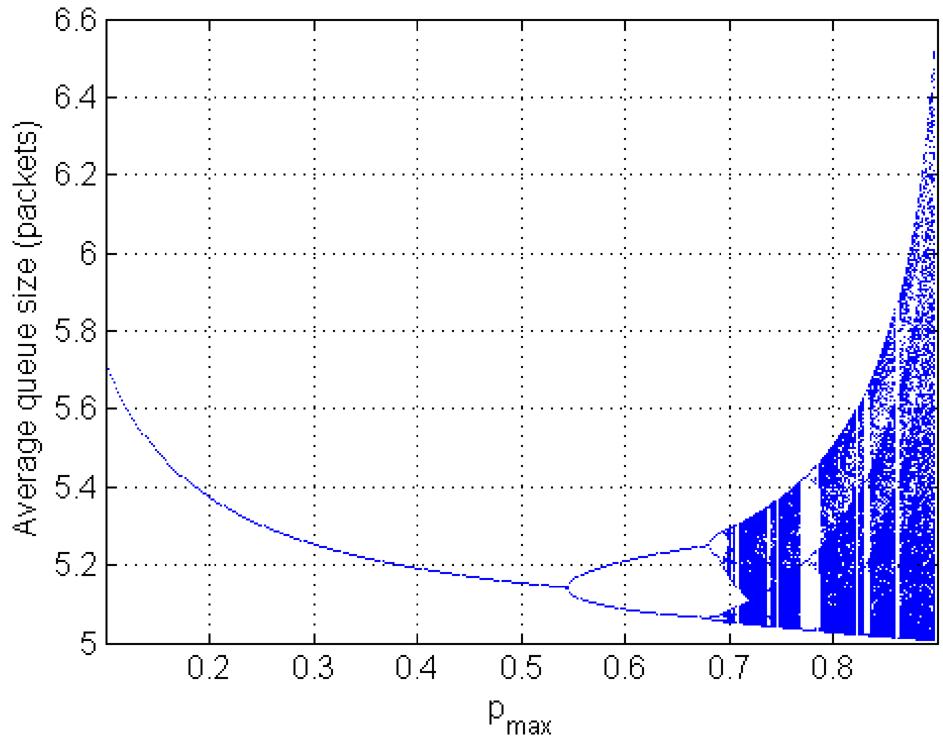
Queue size vs. p_{max}

- $w_{qx} = 0.04$, $q_{min} = 5$, $q_{max} = 15$



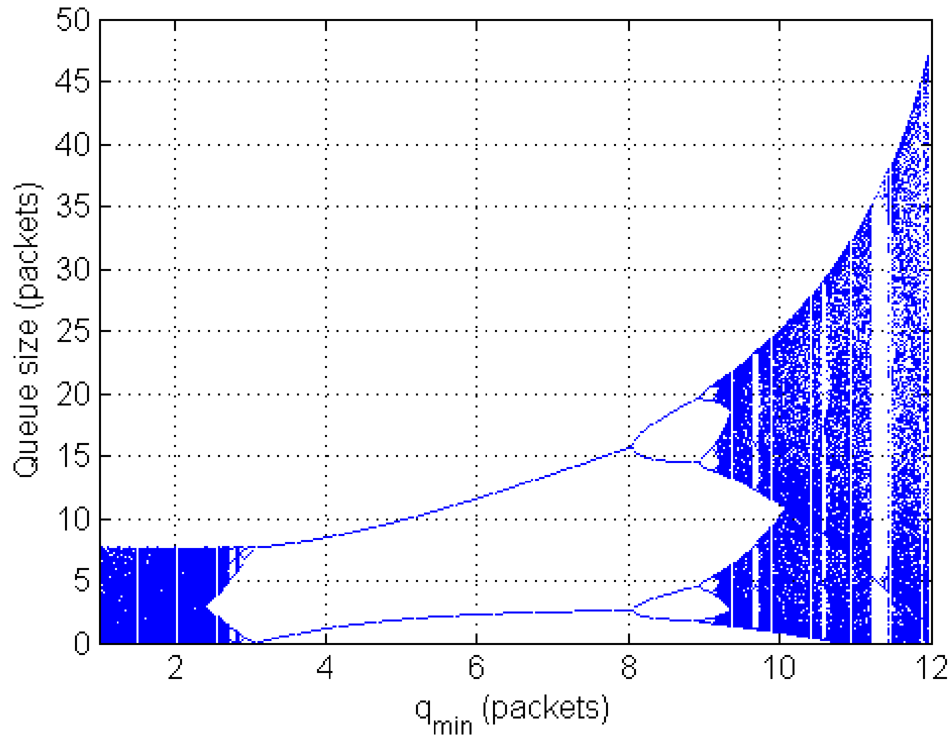
Average queue size vs. p_{max}

- $w_q = 0.04$, $q_{min} = 5$, $q_{max} = 15$



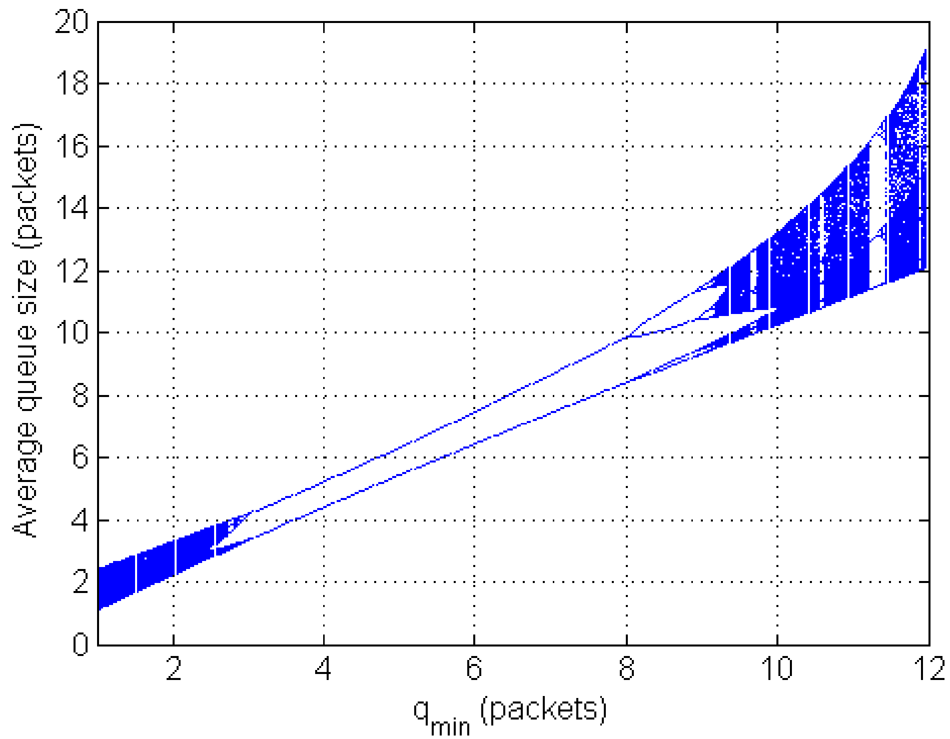
Queue size vs. q_{min}/q_{max}

- $w_q = 0.2$, $p_{max} = 0.1$, $q_{max} = 3 \times q_{min}$



Average queue size vs. q_{min}/q_{max}

- $w_q = 0.2$, $p_{max} = 0.1$, $q_{max} = 3 \times q_{min}$



S-TCP/RED model

- S-model: a discrete nonlinear dynamical model of TCP Reno with RED
- Two state variables:
  - window size
  - average queue size
- The proposed TCP/RED model is:
  - simple and intuitively derived
  - able to capture detailed dynamical behavior of TCP/RED systems
  - has been verified via ns-2 simulations

S-TCP/RED model: state variable and parameters

- q_{k+1}: instantaneous queue size in round k+1
- $\overline{q}_{k+1}$: average queue size in round k+1
- W_{k+1}: current TCP window size in round k+1
- w_q: queue weight in RED
- p_k: drop probability in round k
- RTT_{k+1}: round-trip time at k+1
- C: capacity of the link between the two routers
- M: packet size
- d: round-trip propagation delay
- ssthesh: slow start threshold
- rwnd: receiver's advertised window size

S-TCP/RED model: no packet loss, details

- current queue size:

$$q_{k+1} = q_k + W_{k+1} - C \cdot \frac{RTT_{k+1}}{M}$$

$$= q_k + W_{k+1} - \frac{C}{M}\left(d + \frac{q_k M}{C}\right)$$

$$= W_{k+1} - \frac{C \cdot d}{M}$$

- average queue size:

$$\overline{q}_{k+1} = (1 - w_q) \cdot \overline{q}_k + w_q \cdot \max\left(W_{k+1} - \frac{C \cdot d}{M}, 0\right)$$

S-TCP/RED model: no packet loss

- drop probability: $p_k W_k < 0.5$
- window size:

$$W_{k+1} = \begin{cases} \min(2W_k, ssthresh) & if\ W_k < ssthresh \\ \min(W_k + 1, rwnd) & if\ W_k \geq ssthresh \end{cases}$$

- average queue size:

$$\overline{q}_{k+1} = (1 - w_q)^{W_{k+1}} \overline{q}_k + (1 - (1 - w_q)^{W_{k+1}}) \cdot \max(W_{k+1} - \frac{C \cdot d}{M}, 0)$$

S-TCP/RED model: one packet loss

- drop probability: $0.5 \le p_k W_k < 1.5$
- window size: $W_{k+1} = \frac{1}{2} W_k$
- average queue size:

$$\overline{q}_{k+1} = (1-w_q)^{W_{k+1}} \overline{q}_k + (1-(1-w_q)^{W_{k+1}}) \cdot \max(W_{k+1} - \frac{C \cdot d}{M}, 0)$$

S-TCP/RED model: two packet losses

- drop probability: $p_k W_k \geq 1.5$
- window size: $W_{k+1} = 0$
- average queue size: $\overline{q}_{k+1} = \overline{q}_k$

Roadmap

- Introduction
- TCP/RED congestion control algorithms: an overview
- Discrete-time dynamical models of TCP Reno with RED
- Discontinuity-induced bifurcations in TCP/RED
- Stability analysis of RED gateway with multiple TCP Reno connections
- Stability study of TCP/RED system using detrended fluctuation analysis
- References

RED: default parameters

- RED parameters:

  S. Floyd, "RED: Discussions of Setting Parameters," Nov. 1997: http://www.icir.org/floyd/REDparameters.txt

Queue weight (w_q)	0.002
Maximum drop probability (p_{max})	0.1
Minimum queue threshold (q_{min})	5 (packets)
Maximum queue threshold (q_{max})	15 (packets)
Packet size (M)	4,000 (bytes)

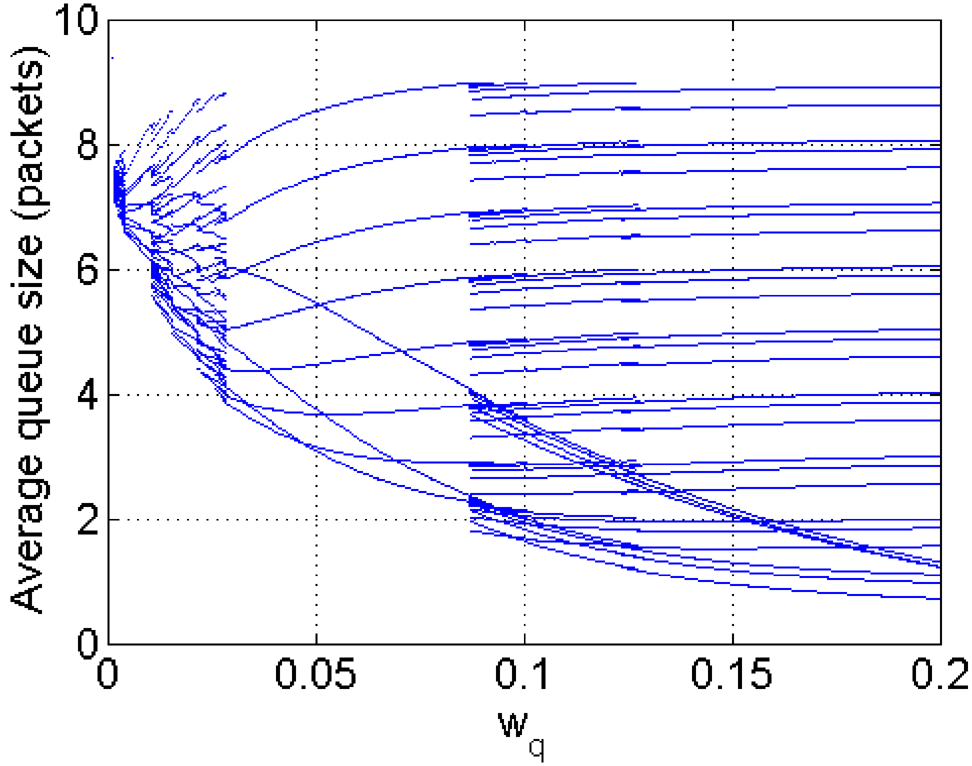
TCP/RED: bifurcation and chaos

- Bifurcation diagrams for various values of the system parameters:
  - queue weight: w_q
  - maximum drop probability: p_{max}
  - queue thresholds: q_{min} and q_{max} ($q_{max}/q_{min} = 3$)
  - round-trip propagation delay: d

Queue weight (w_q)	0.002
Maximum drop probability (p_{max})	0.1
Minimum queue threshold (q_{min})	5 (packets)
Maximum queue threshold (q_{max})	15 (packets)
Packet size (M)	4,000 (bytes)

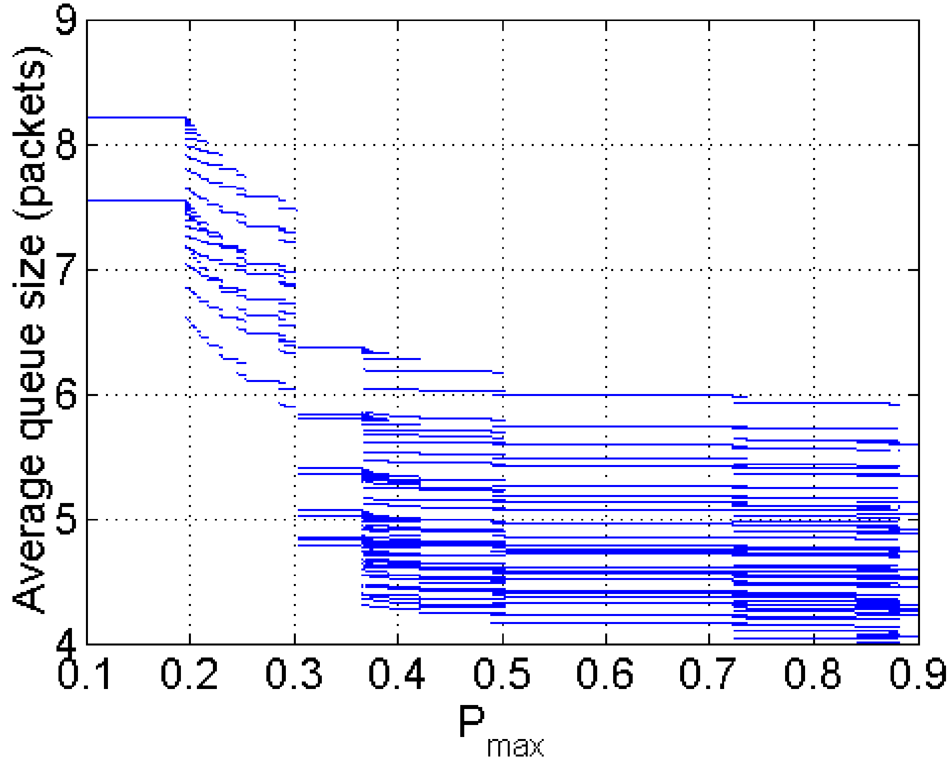
Average queue size vs. w_q

- p_{max} = 0.1, q_{min} = 5, q_{max} = 15, and sstresh = 80



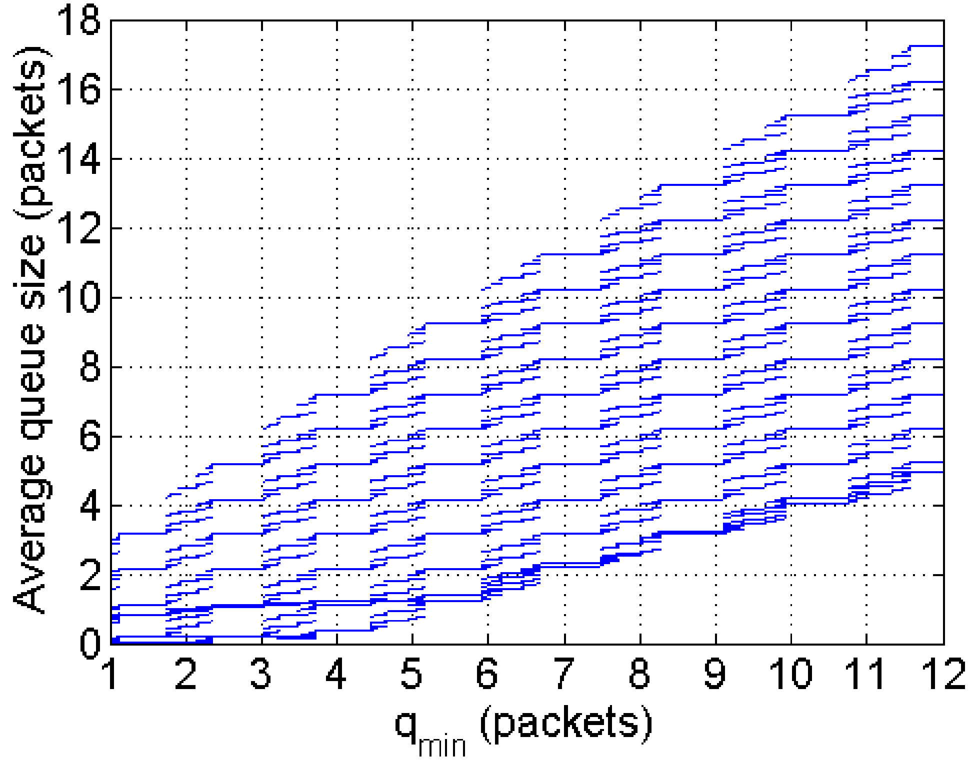
Average queue size vs. p_{max}

- $w_q = 0.01$, $q_{min} = 5$, $q_{max} = 15$, and ssthresh = 20



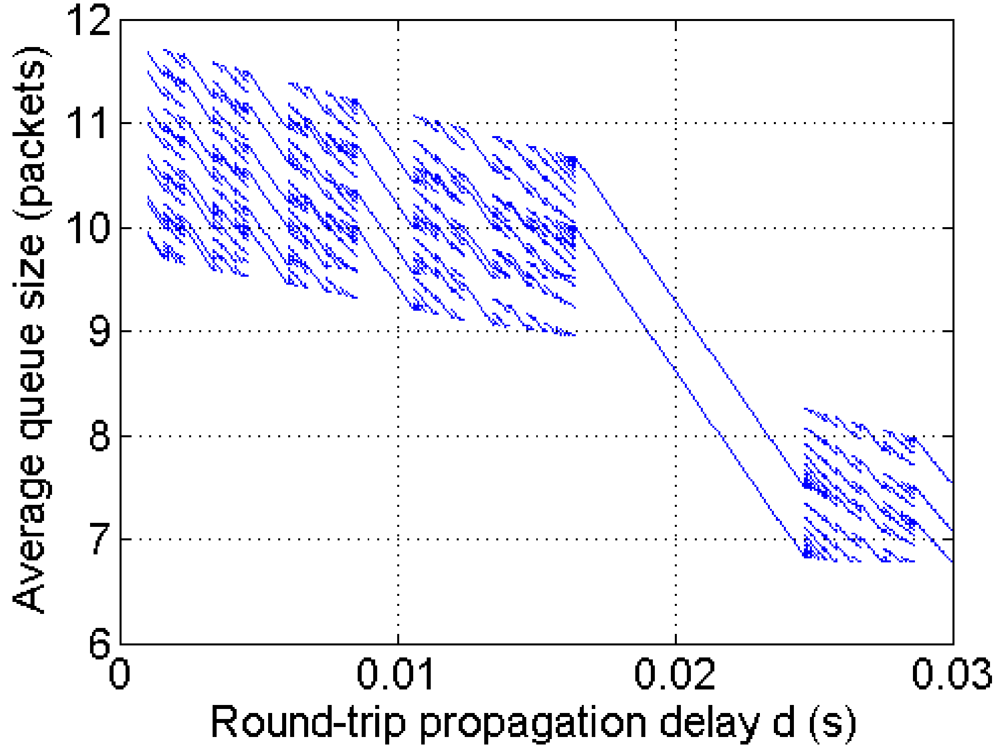
Average queue size vs. q_{min}/q_{max}

- $w_q = 0.01$, $p_{max} = 0.1$, $q_{max} = 3\ q_{min}$, and ssthresh = 20



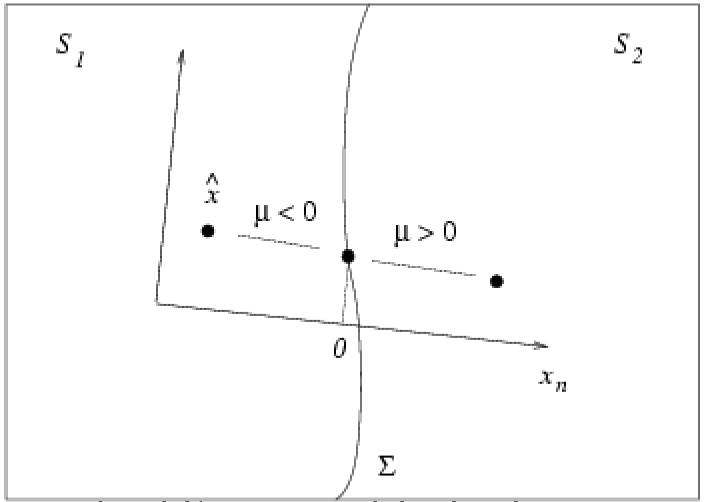
Average queue size vs. d

- $w_q = 0.01$, $p_{max} = 0.1$, $q_{min} = 5$, $q_{max} = 15$, and ssthresh = 20



An analytical explanation

- Nonsmooth systems may exhibit discontinuity-induced bifurcations: a class of bifurcations unique to their nonsmooth nature
- These phenomena occur when a fixed point, cycle, or aperiodic attractor interacts nontrivially with one of the phase space boundaries where the system is discontinuous



Discontinuity-induced bifurcations: classification

- Standard:
  - SN (smooth saddle-node)
  - PD (smooth period-doubling)
- C-bifurcations or discontinuity-induced bifurcations
  - PWS maps: border collisions of fixed points
  - PWS flows: discontinuous bifurcations of equilibriums
  - Grazing bifurcations of periodic orbits
  - Sliding bifurcations

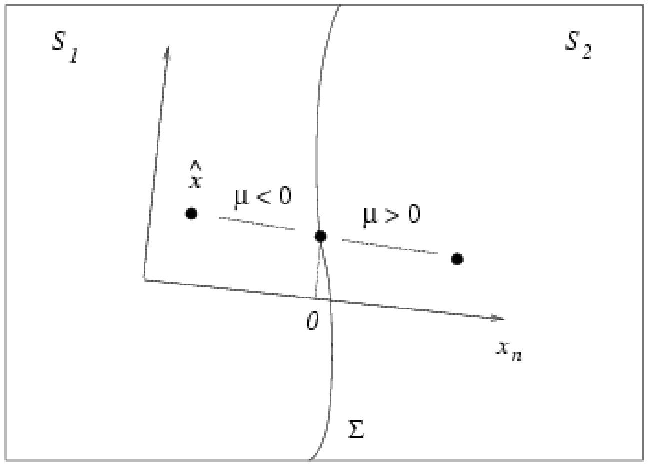
Border collisions in PWS maps

- Consider a map of the form:

$$x_{k+1} = \begin{cases} F_1(x_k, p), & H(x_k) < 0 \\ F_2(x_k, p), & H(x_k) > 0 \end{cases}$$

- A fixed point is undergoing a border-collision bifurcation at $p=0$ if:

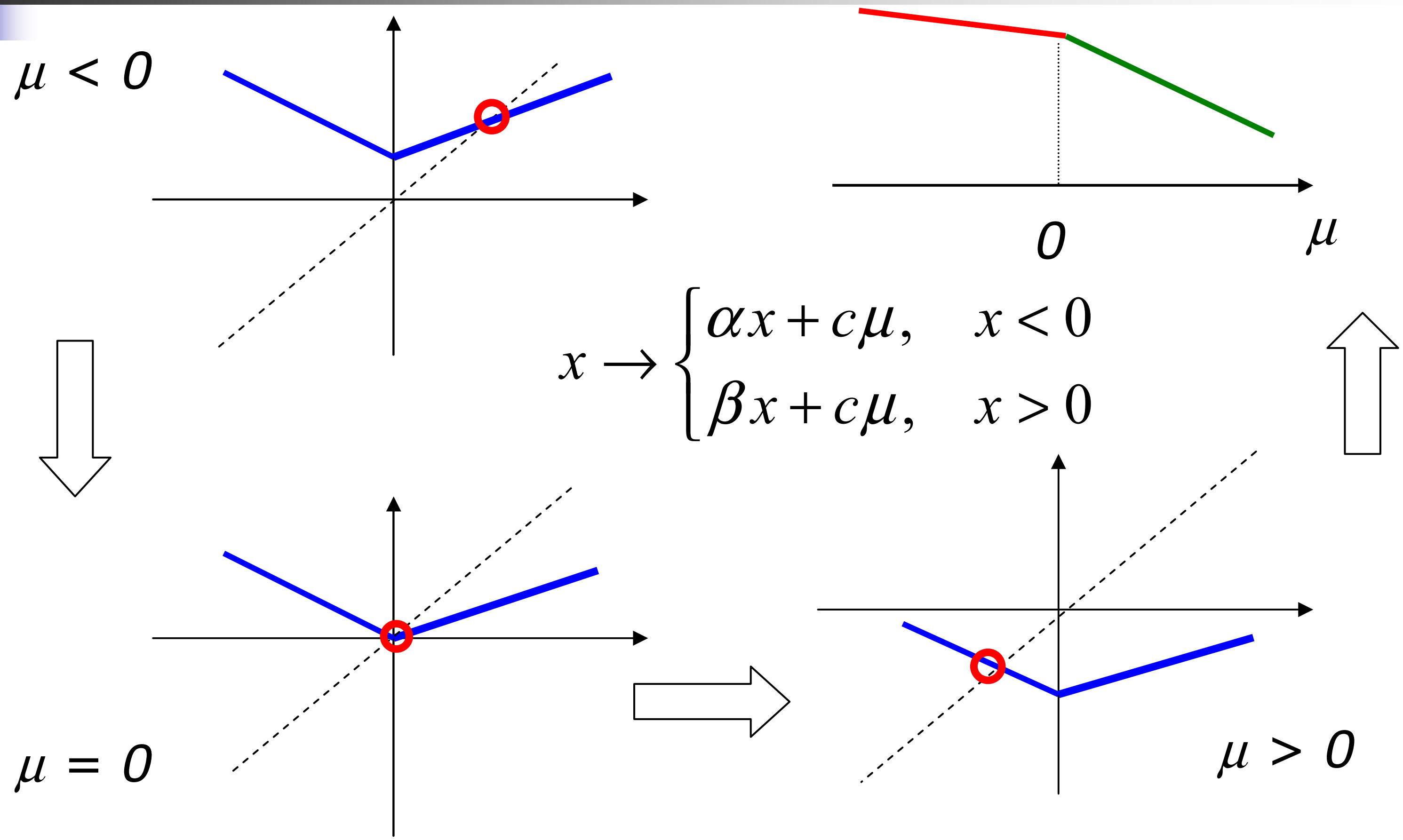
- $\mu \in (-\varepsilon,0) \Rightarrow x^* \in S_1$
- $\mu \in (0,\varepsilon) \Rightarrow x^* \in S_2$
- $\mu = 0 \Rightarrow x^* \in \Sigma$
- $DF_1 \neq DF_2$ on Σ



Classifying border collisions

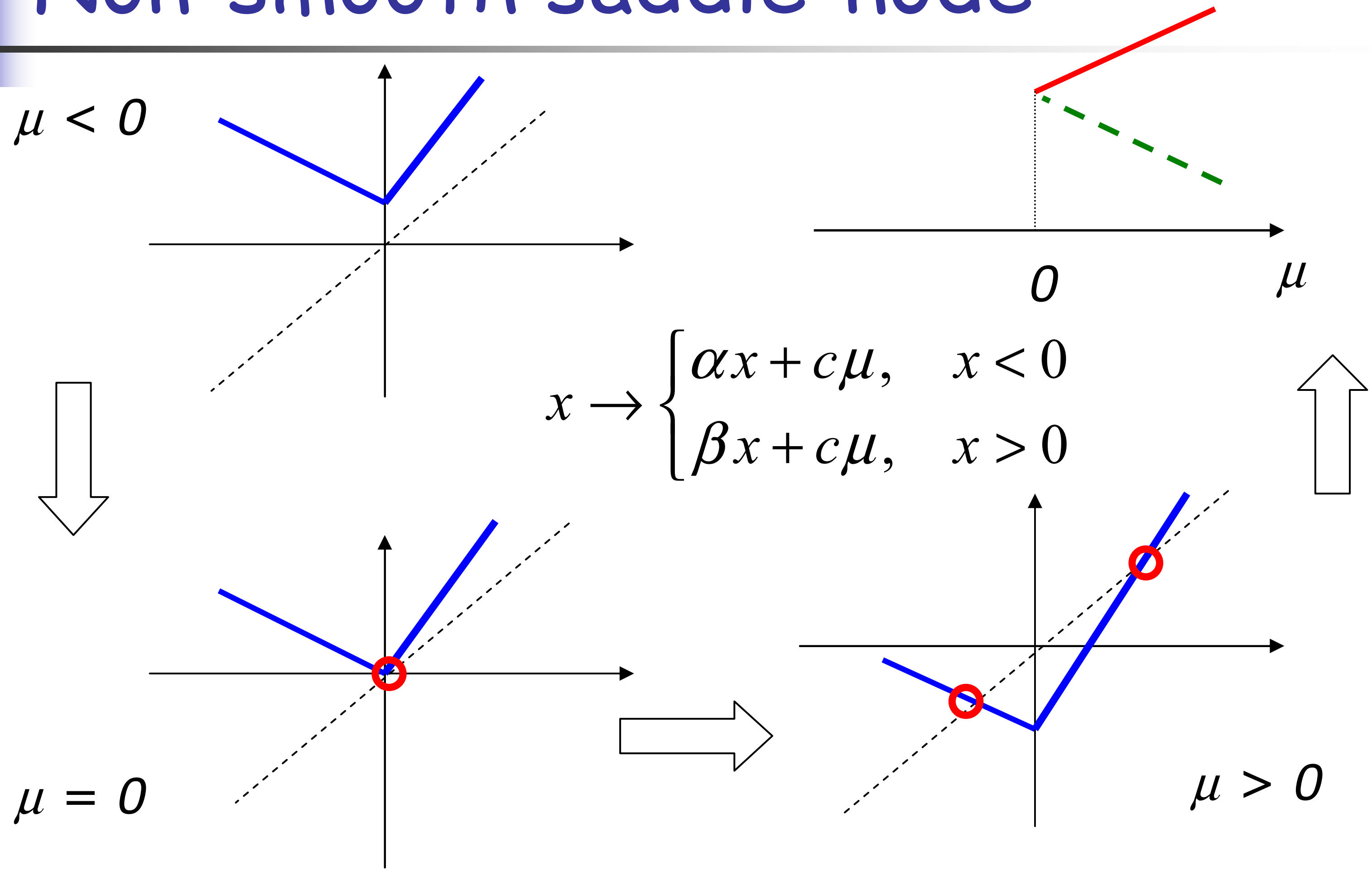
- Several scenarios are possible when a border-collision occurs
- They can be classified by observing the map eigenvalues on both sides of the boundary
- The phenomenon can be illustrated by a very simple 1D map where the eigenvalues are the slopes of the map on both sides of the boundary

Persistence



$$x \rightarrow \begin{cases} \alpha x + c\mu, & x < 0 \\ \beta x + c\mu, & x > 0 \end{cases}$$

Non-smooth saddle-node



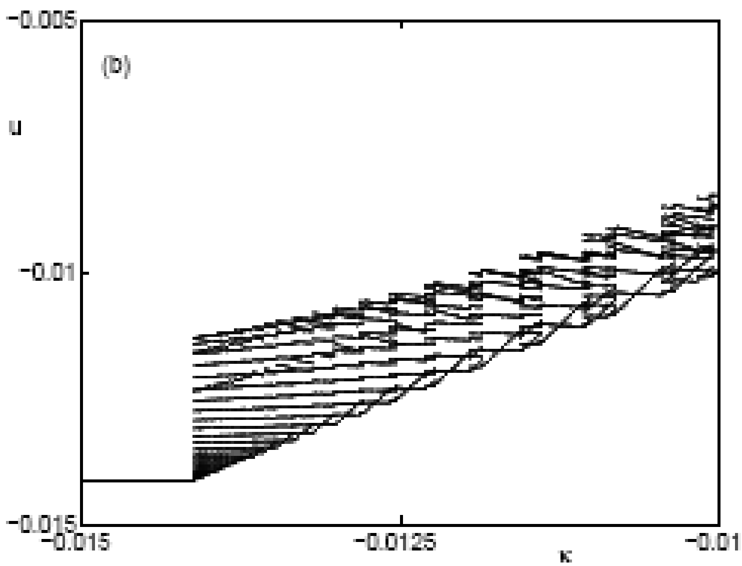
$$x \to \begin{cases} \alpha x + c\mu, & x < 0 \\ \beta x + c\mu, & x > 0 \end{cases}$$

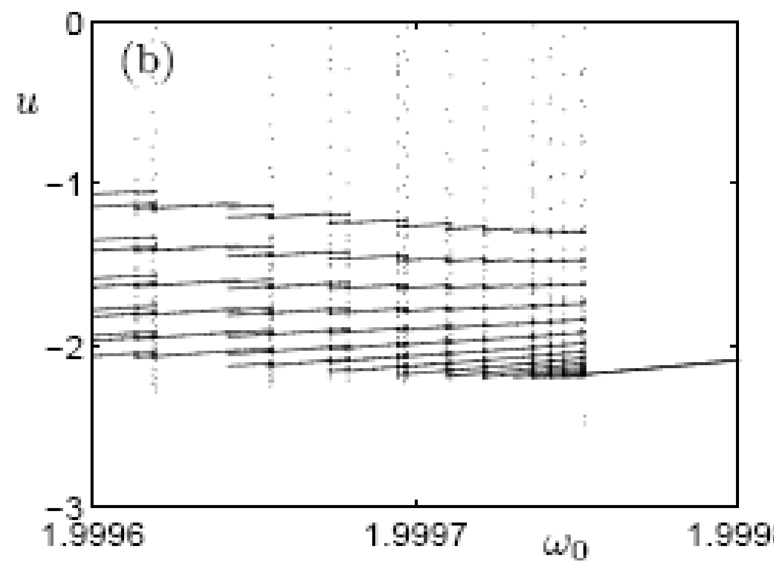
Border-collisions in the TCP/RED model

- The analysis has focussed mostly on continuous maps
- Recently proposed: further bifurcations are possible when the map is piecewise with a gap
- Complete classification method is available only for the one-dimensional case
- The TCP/RED case is a 2D map with a gap: its dynamics resemble closely those observed in very different systems: the impact oscillator considered by Budd and Piiroinen, 2006
- They might be explained in terms of border-collision bifurcations of 2D discontinuous maps

Numerical evidence

Cascades of corner-impact bifurcations in a forced impact oscillator show a striking resemblance to the phenomena detected in the TCP/RED model. They were explained in terms of border-collisions of local maps with a gap.





C. J. Budd and P. Piiroinen, "Corner bifurcations in nonsmoothly forced impact oscillators," to appear in *Physica D*, 2005.

TCP/RED models: conclusions

- We developed discrete-time models for TCP Reno with RED
- TCP/RED models include:
  - slow start, congestion avoidance, fast retransmit, timeout, elements of fast recovery, and RED
- Proposed models were validated by comparing its performance with ns-2 simulation results
- They capture the main features of the dynamical behavior of TCP Reno with RED
- These models were used to study bifurcation and chaos in TPC/RED systems with a single connection

TCP/RED bifurcations: conclusions

- Discrete-time one and two-dimensional models capture the main features of the dynamical behavior of TCP/RED communication algorithms
- These models were used to study bifurcations and chaos in TPC/RED systems with a single connection
- Bifurcations diagrams were characterized by period-adding cascades and devil staircases
- The observed behavior can be explained in terms of a novel class of piecewise-smooth maps with a gap

Roadmap

- Introduction
- TCP/RED congestion control algorithms: an overview
- Discrete-time dynamical models of TCP Reno with RED
- Discontinuity-induced bifurcations in TCP/RED
- Stability analysis of RED gateway with multiple TCP Reno connections
- Stability study of TCP/RED system using detrended fluctuation analysis
- References

Fluid-flow model of TCP-RED

Window size: $$\frac{dw(t)}{dt} = \frac{1}{r(t)} - \frac{w(t)}{2} \cdot \frac{w(t-r(t))}{r(t-r(t))} p(t-r(t))$$

additive increase | multiplicative decrease | loss arrival rate

Instantaneous queue length: $$\frac{dq(t)}{dt} = N\frac{w(t)}{r(t)} - C$$

incoming traffic | outgoing traffic

Round trip time: $$r(t) = \frac{q(t)}{C} + R_0$$

queuing delay | propagation delay

Fluid-flow model of TCP-RED

Average queue length:

$$\frac{dx}{dt} = \frac{\ln(1-\alpha)}{\delta}\left(x(t) - q(t) \right)$$

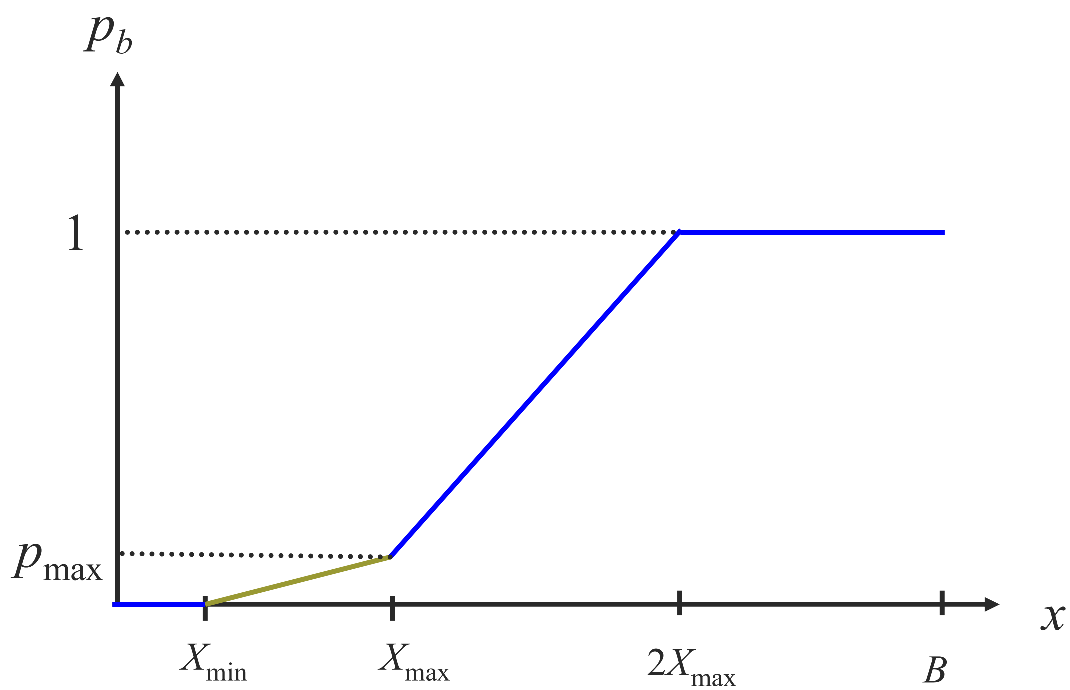
α : queue averaging weight

δ : sampling rate ~1/C

Marking/dropping probability:

$$p_b(t) = \begin{cases} 0 & 0 \le x(t) < X_{\min} \\ \dfrac{x(t) - X_{\min}}{X_{\max} - X_{\min}} p_{\max} & X_{\min} \le x(t) \le X_{\max} \\ p_{\max} - \dfrac{x(t) - X_{\max}}{X_{\max} - X_{\min}} (1 - p_{\max}) & X_{\max} < x(t) \le 2X_{\max} \\ 1 & X_{\max} < x(t) \le B \end{cases}$$

Marking/dropping probability



Steady-state solution and target queue length

Let $N(t) \equiv N$, $C(t) \equiv C$, and $p(t) = \frac{x(t) - X_{\min}}{X_{\max} - X_{\min}} p_{\max}$

Equilibrium point (q_0, r_0, w_0, p_0) :

$$q_0 = \frac{X_{\max} + X_{\min}}{\phi}$$

$$r_0 = \frac{q_0}{C} + R_0$$

$$w_0 = \frac{Cr_0}{N}$$

$$p_0 = p_{\max} \frac{1 + (1 - \phi)\frac{X_{\min}}{X_{\max}}}{\phi(1 - \frac{X_{\min}}{X_{\max}})} = 2\left(\frac{N}{Cr_0}\right)^2$$

Linearization and characteristic equation

$$
\begin{cases}
\delta \dot{w}(t) = \dfrac{-N}{r_0{}^2 C}(\delta w(t) + \delta w(t - r_0)) + \dfrac{-1}{r_0{}^2 C}(\delta q(t) - \delta q(t - r_0)) + \dfrac{-r_0 C^2}{2N^2}\delta p(t - r_0) \\
\delta \dot{q}(t) = \dfrac{N}{r_0}\delta w(t) - \dfrac{1}{r_0}\delta q(t) \\
\delta \dot{p}(t) = C\ln(1-\alpha)(\delta p(t) - \beta \delta q(t))
\end{cases}
$$

where:

$$
\begin{cases}
\delta w = w - w_0 \\
\delta q = q - q_0 \\
\delta p = p - p_0
\end{cases}
$$

$$\beta = \frac{p_{\max}}{X_{\max} - X_{\min}}$$

Characteristic equation in Laplace domain:

$$\lambda^3 + \left(\frac{1}{r_0} + \frac{N}{r_0{}^2 C} - \alpha_1 C\right)\lambda^2 + \left(\frac{2N}{r_0{}^3 C} - \frac{\alpha_1 C}{r_0} - \frac{\alpha_1 N}{r_0{}^2}\right)\lambda - \frac{2\alpha_1 N}{r_0{}^3} + \left(\frac{N}{r_0{}^2 C}\lambda^2 - \frac{N\alpha_1}{r_0{}^2}\lambda - \frac{C^3\alpha_1\beta}{2N}\right)e^{-\lambda r_0} = 0$$

$$\text{where}: \alpha_1 = \ln(1-\alpha)$$

Characteristic equation

With Padé(1,1) approximation:

$$e^{-\lambda r_0} \approx \frac{1-\frac{r_0\lambda}{2}}{1+\frac{r_0\lambda}{2}}$$

we obtain:

$$\lambda^4 + a_1\lambda^3 + a_2\lambda^2 + a_3\lambda + a_4 = 0$$

where:

$$\begin{cases} a_1 = (\frac{3}{\mathrm{K}} - \alpha_1)C \\ a_2 = (\frac{6N}{\mathrm{K}^3} + \frac{2}{\mathrm{K}^2} - \frac{3\alpha_1}{\mathrm{K}})C^2 \\ a_3 = (\frac{4N}{\mathrm{K}^4} - \frac{6N\alpha_1}{\mathrm{K}^3} - \frac{2\alpha_1}{\mathrm{K}^2} + \frac{2\alpha_1 N}{\Phi \mathrm{K}^2})C^3 \\ a_4 = -4N\alpha_1(\frac{1}{\mathrm{K}^4} - \frac{\alpha_1}{\Phi \mathrm{K}})C^4 \\ K = Cr_0 \\ \Phi = X_{\max} - X_{\min} \end{cases}$$

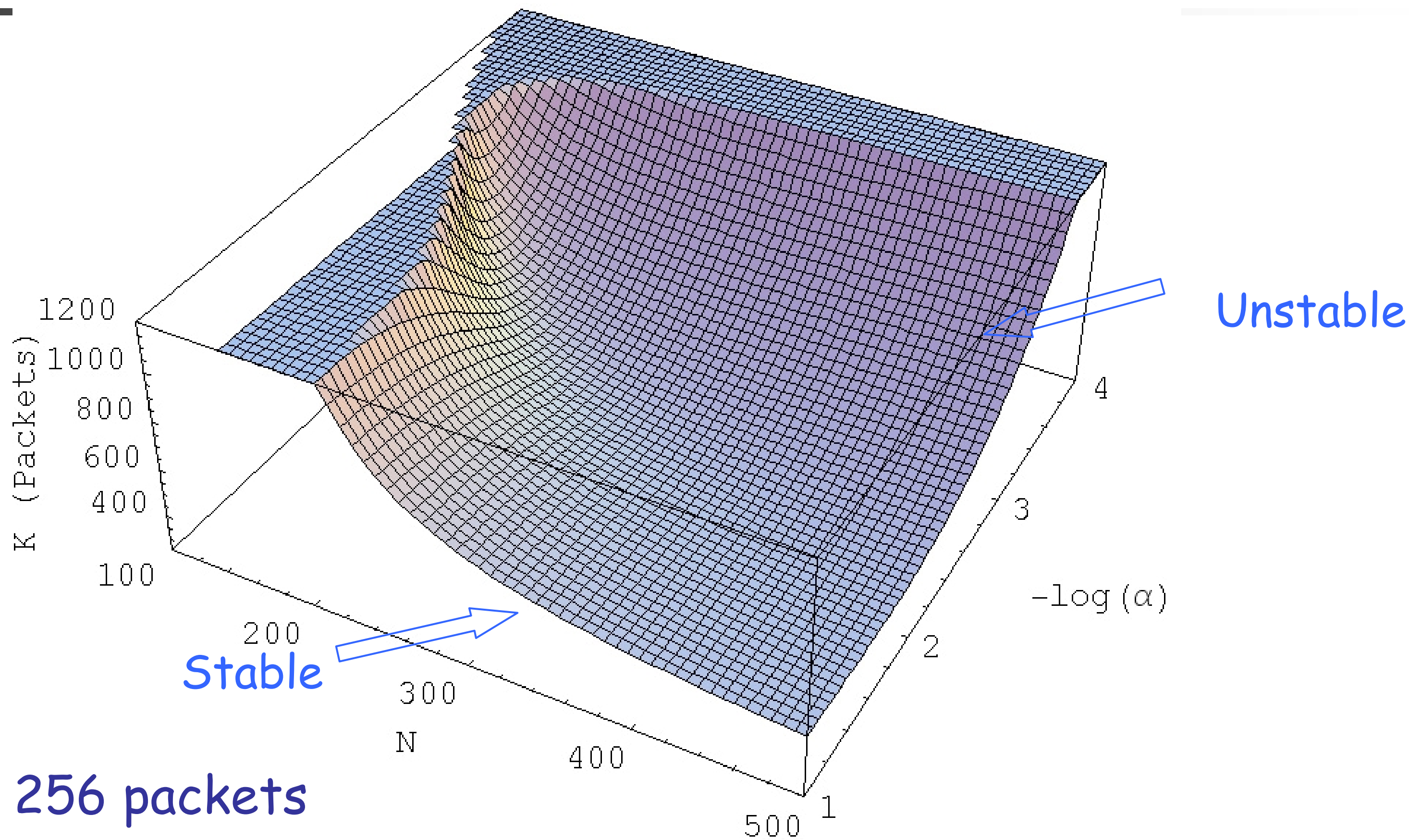
Stability conditions for TCP/RED

The Routh-Hurwitz stability criterion provides the necessary and sufficient stability conditions for the approximated system in terms of network parameters (N, α, K, Φ):

$$\left(\frac{4N}{K^2}-\frac{6N\alpha_1}{K}-2\alpha_1+\frac{2\alpha_1 N}{\Phi}\right)\left[\left(\frac{3}{K}-\alpha_1\right)\left(\frac{6N}{K^2}+\frac{2}{K}-3\alpha_1\right)\right.$$

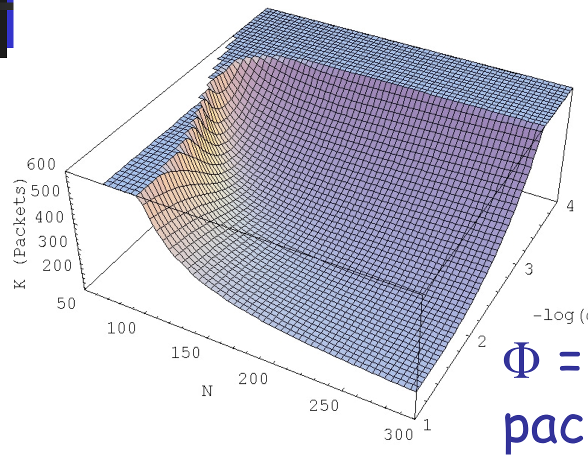
$$\left.-\left(\frac{4N}{K^3}-\frac{6N\alpha_1}{K^2}-\frac{2\alpha_1}{K}+\frac{2\alpha_1}{\Phi K}\right)\right]+4N\alpha_1\left(\frac{3}{K}-\alpha_1\right)^2\left(\frac{1}{K}-\frac{1}{\Phi}\right)>0$$

Stable region (K, N, α)

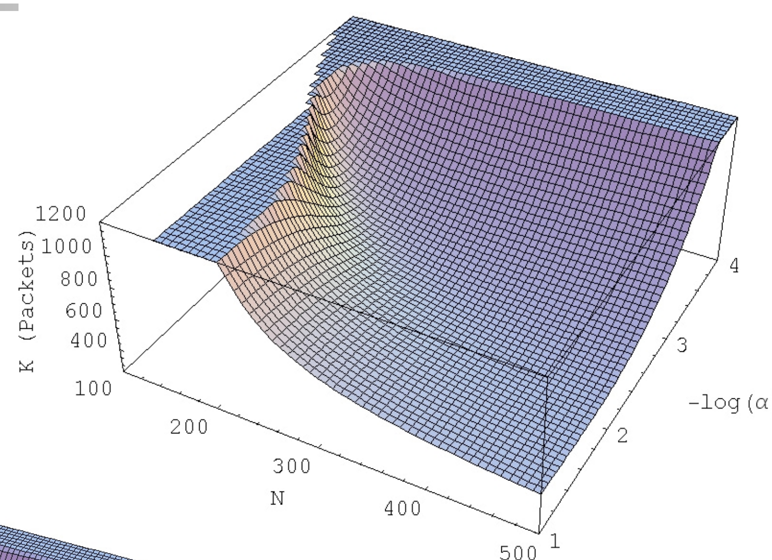


Φ = 256 packets

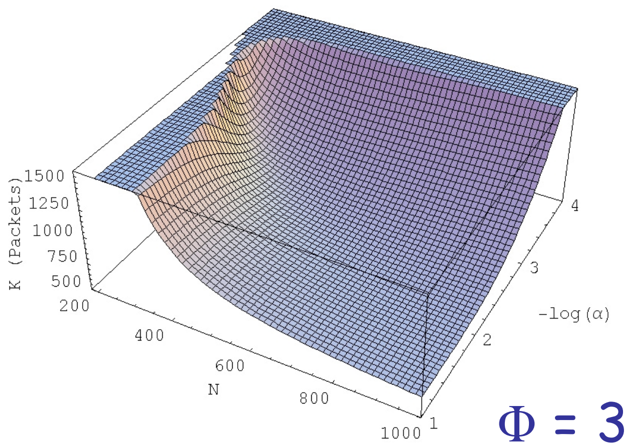
Stable region (K, N, α)



Φ = 128 packets



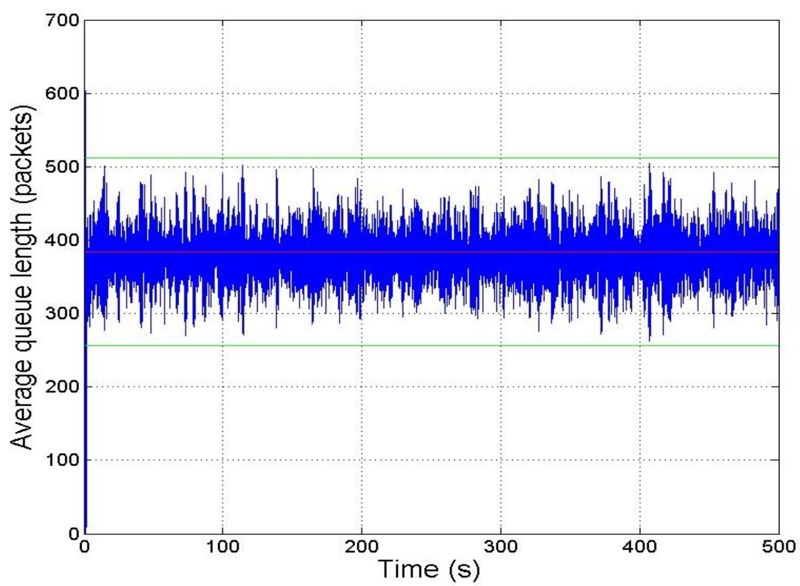
Φ = 256 packets



Φ = 384 packets

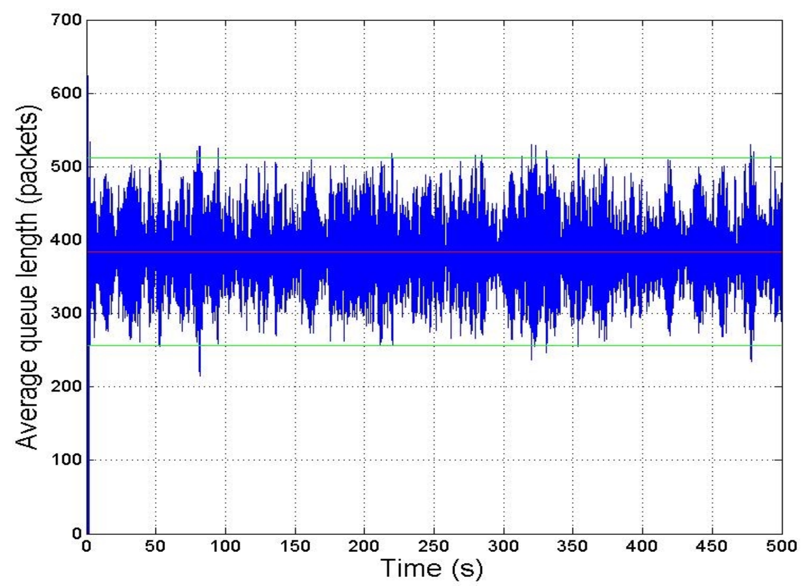
Stable and unstable waveforms of queue length: ns-2 simulations (1)

Stable



K = 758.5 packets (r_0 = 64 ms),
Φ = 256 packets, α = 0.01

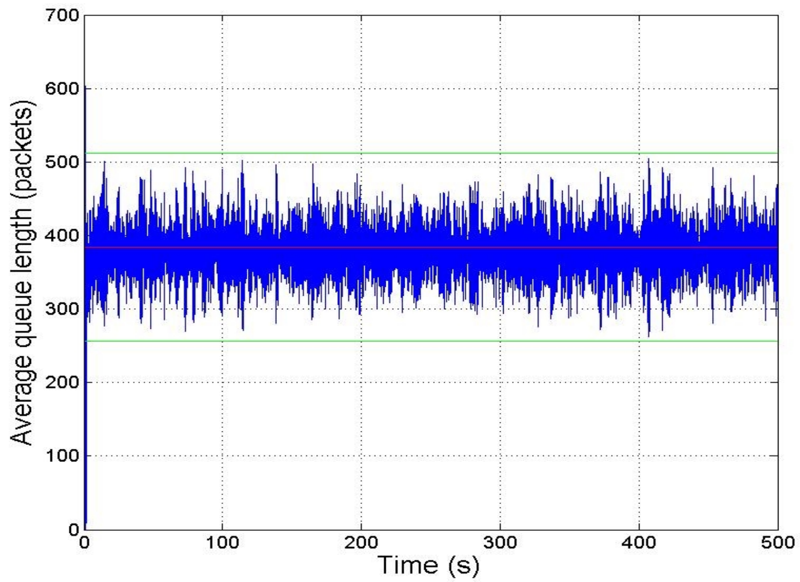
Unstable



K = 865.2 packets (r_0 = 73 ms),
Φ = 256 packets, α = 0.01

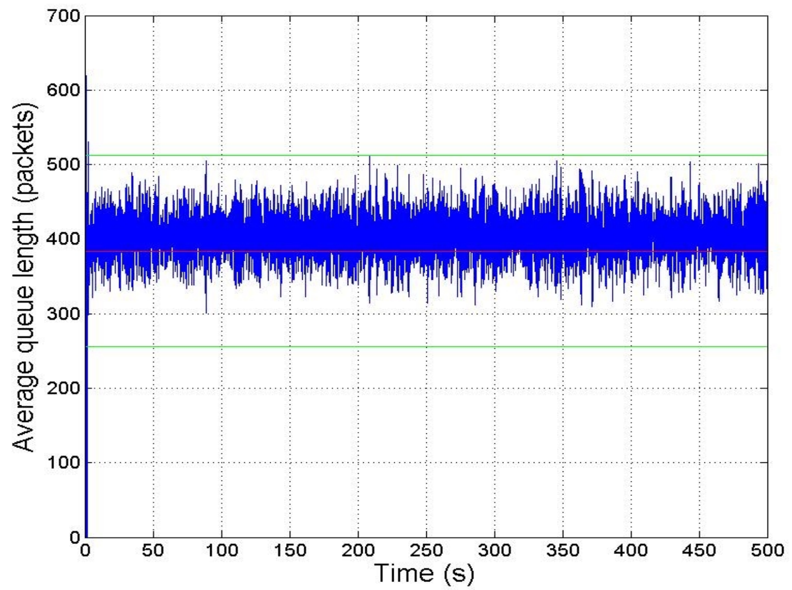
Stable and unstable waveforms of queue length: ns-2 simulations (2)

Stable



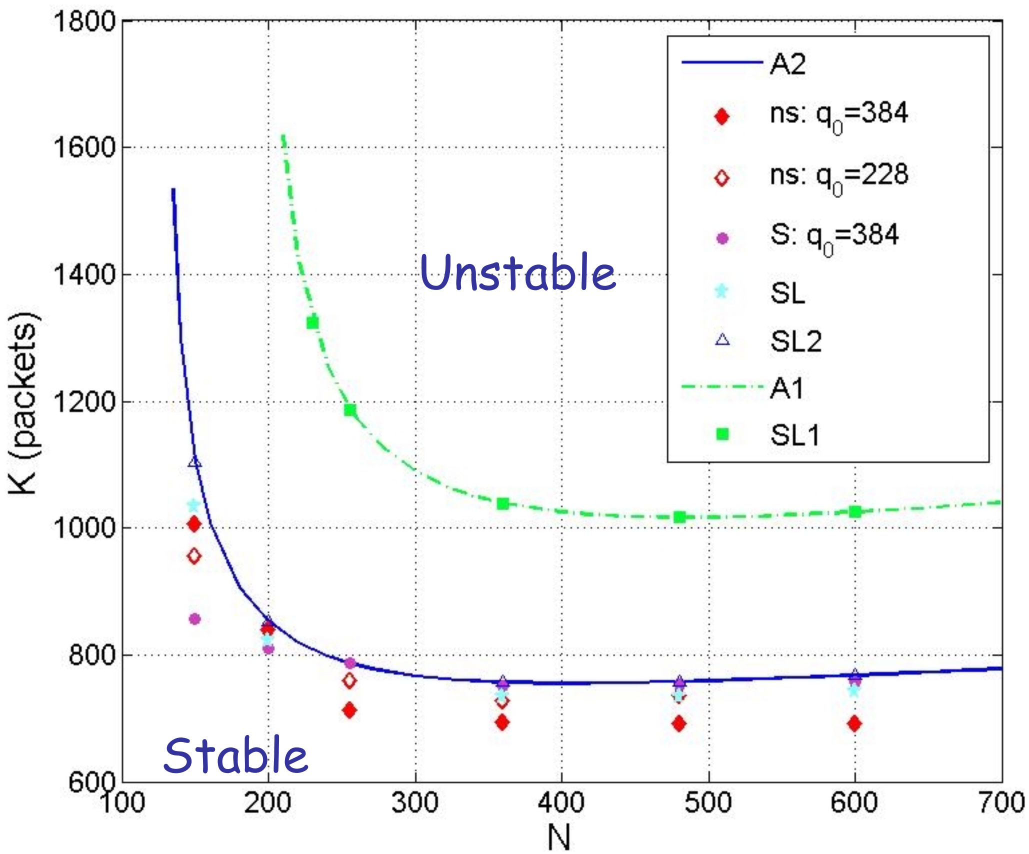
K = 758.5 packets (r_0= 64 ms), Φ = 256 packets, α = 0.01

Unstable



K = 865.2 packets (r_0= 75 ms), Φ = 256 packets, α = 0.01

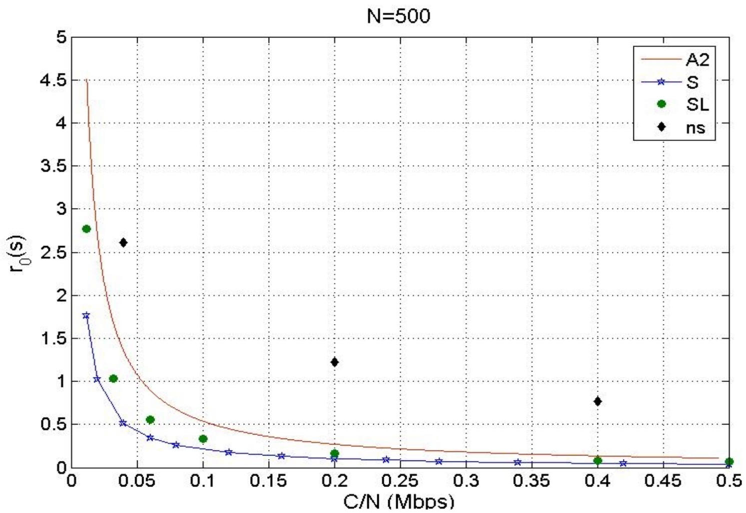
Comparison of simulation methods and approximations

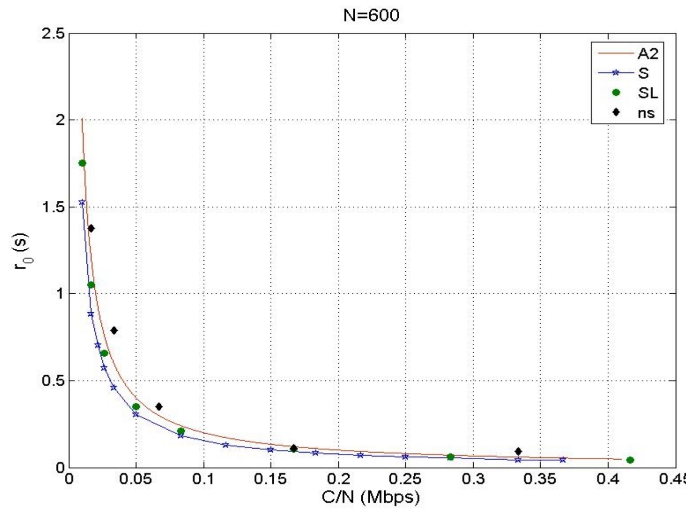


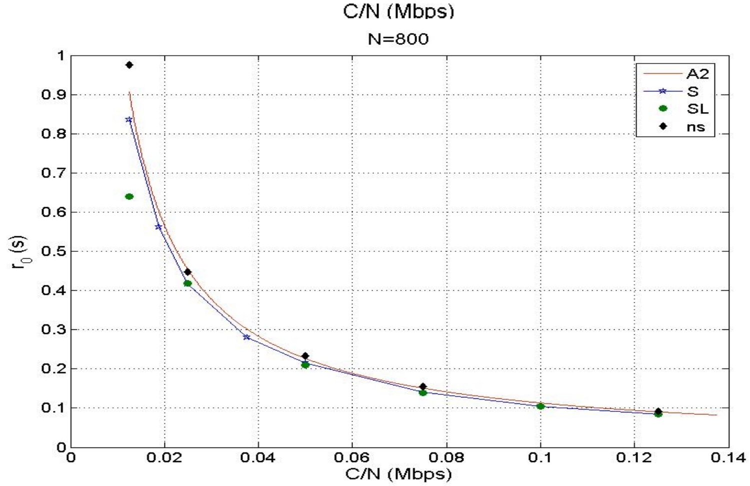
ns	ns-2 simulation
S	Full fluid flow simulation
SL	Linearized fluid flow simulation
SL1	Linearized fluid flow simulation with Padé(0,1) approximation
SL2	Linearized fluid flow simulation with Padé(1,1) approximation
A1	Analytical closed form solution with Padé(0,1) approximation
A2	Analytical closed form solution with Padé(1,1) approximation

Φ = 256 packets and α = 0.001

Comparisons: $K = Cr_o$

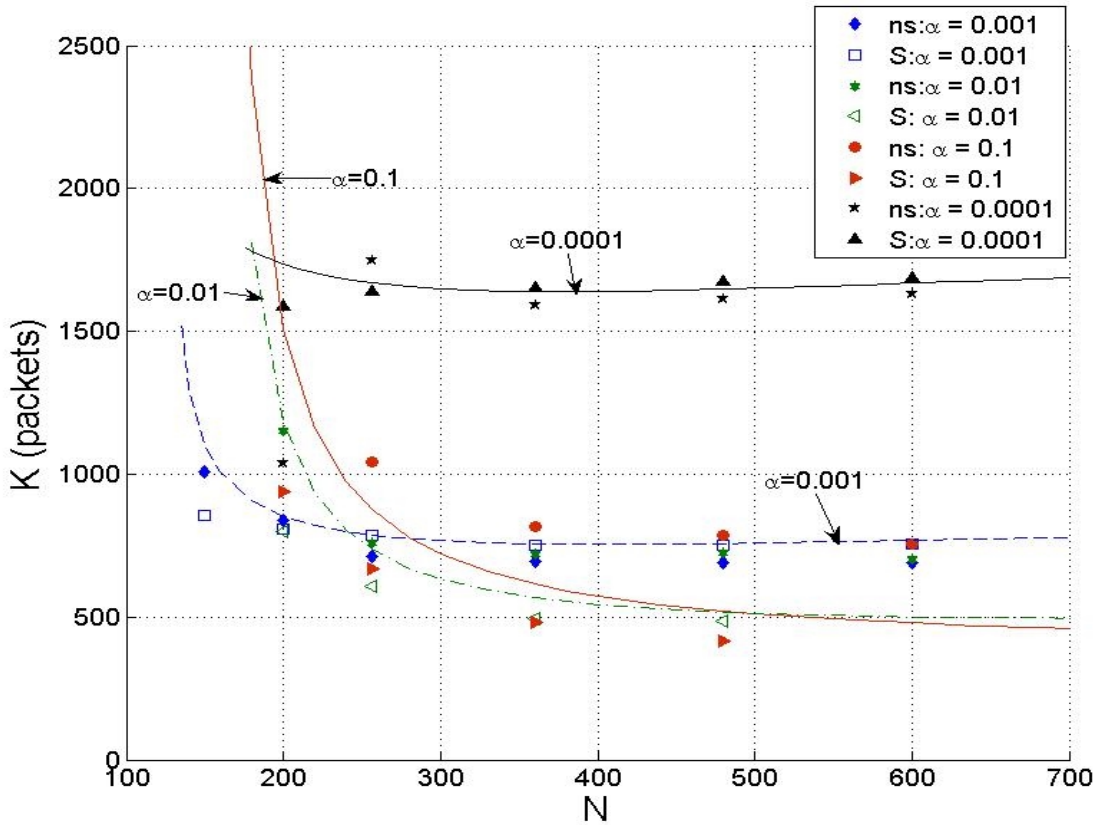






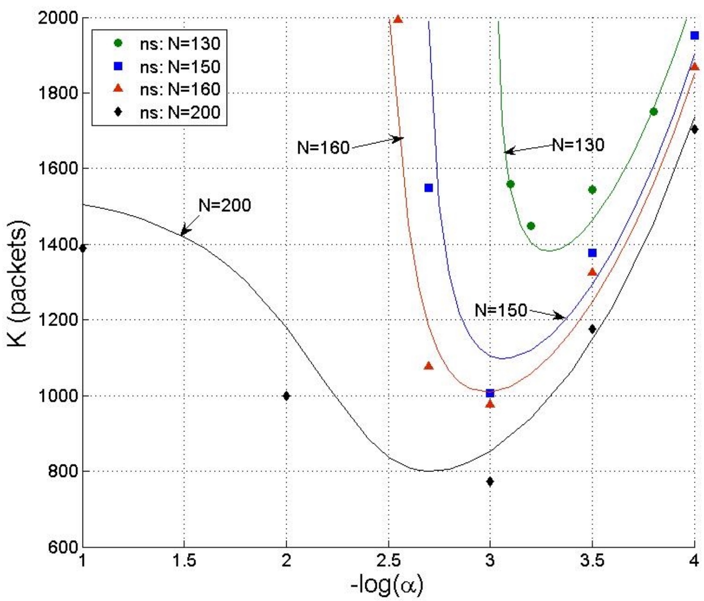
Capacity vs. Round Trip Time:
N = 500, 600, and 800
Φ = 800 packets
q_0 = 500 packets
α = 0.001

Comparisons: varying α

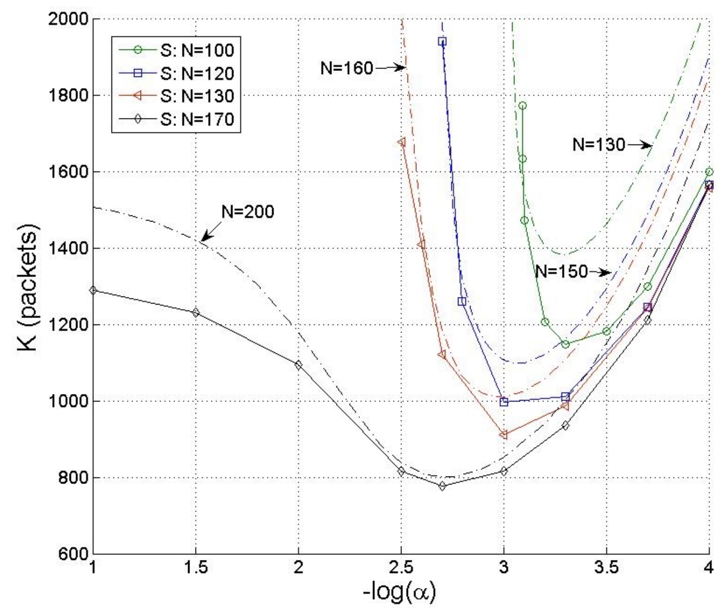


Φ = 256 packets and q_0 = 384 packets

Comparisons: varying N

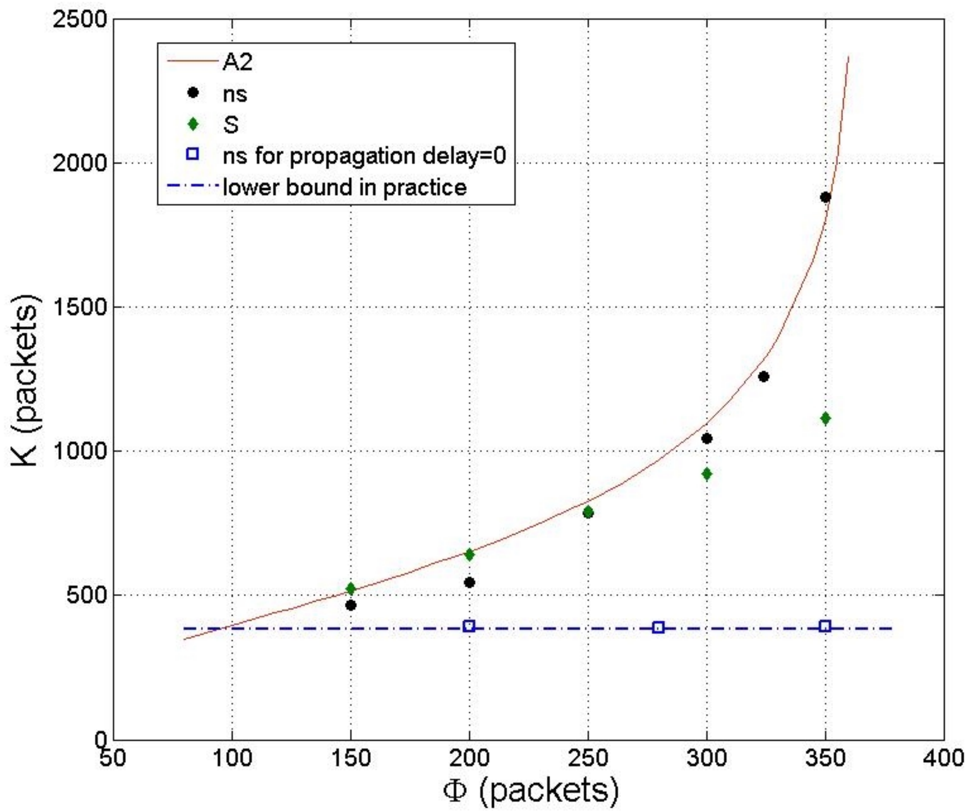


ns-2 results vs. analytical solution for Φ = 256 packets and q_0 = 384 packets



SIMULINK results vs. analytical solution for Φ = 256 packets and q_0 = 384 packets

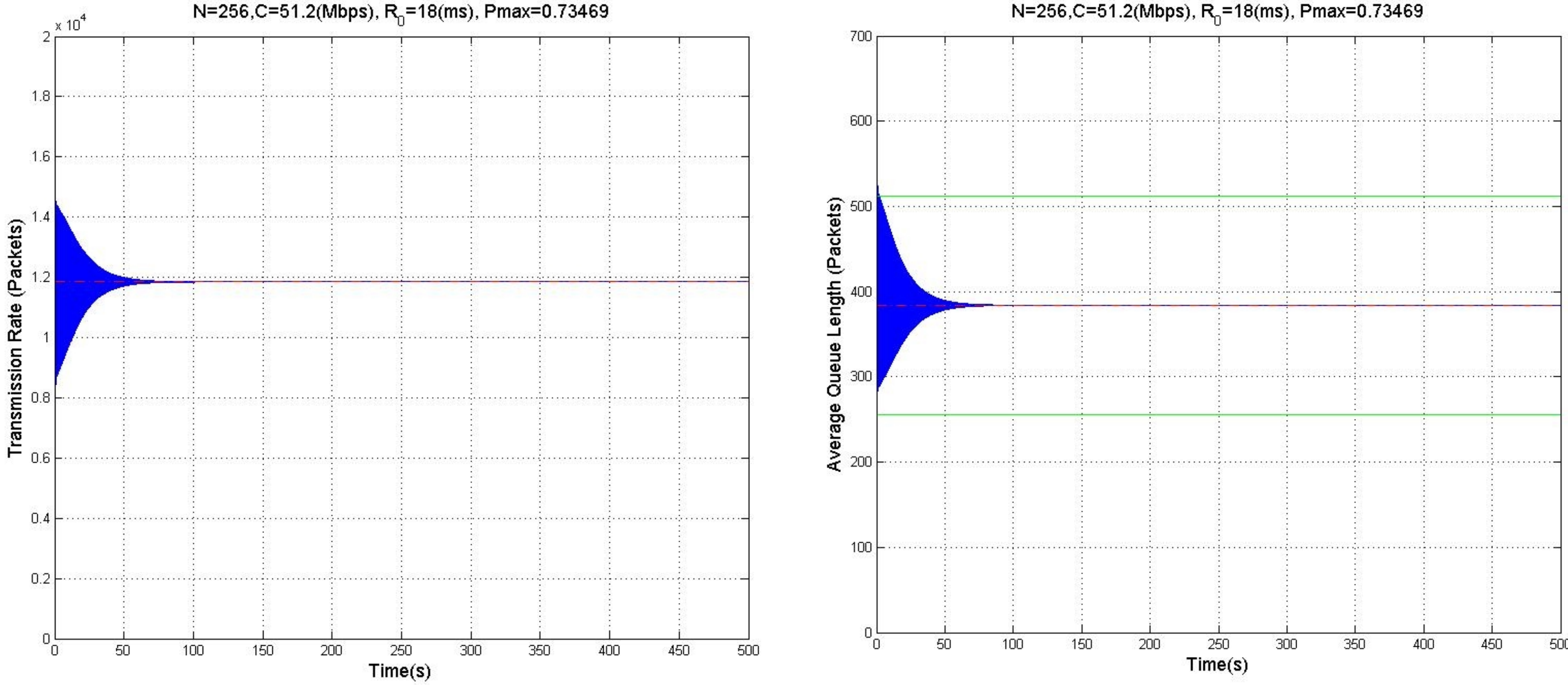
Comparisons: varying Φ



N = 256, α = 0.001 and q_0 = 384 packets

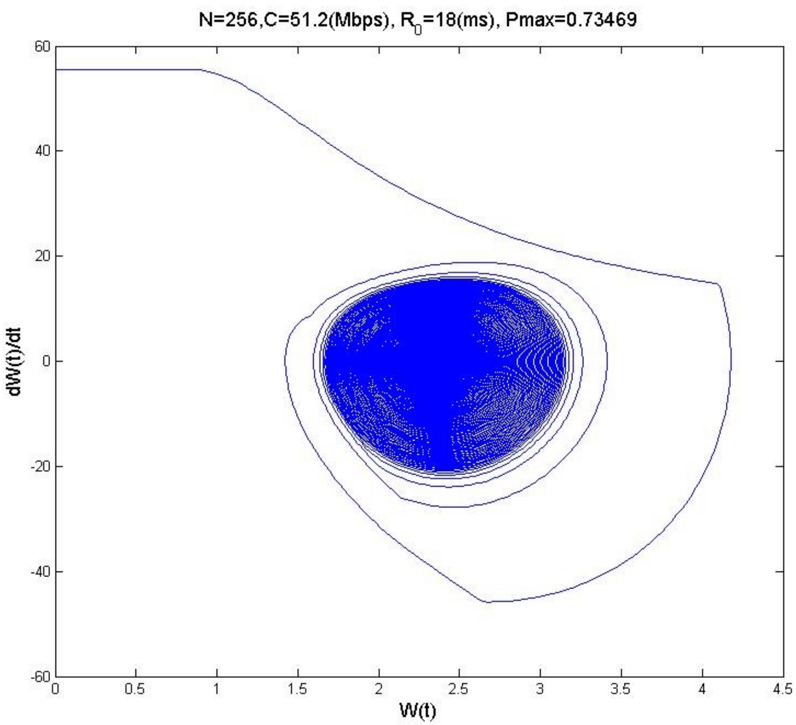
Appendix 1

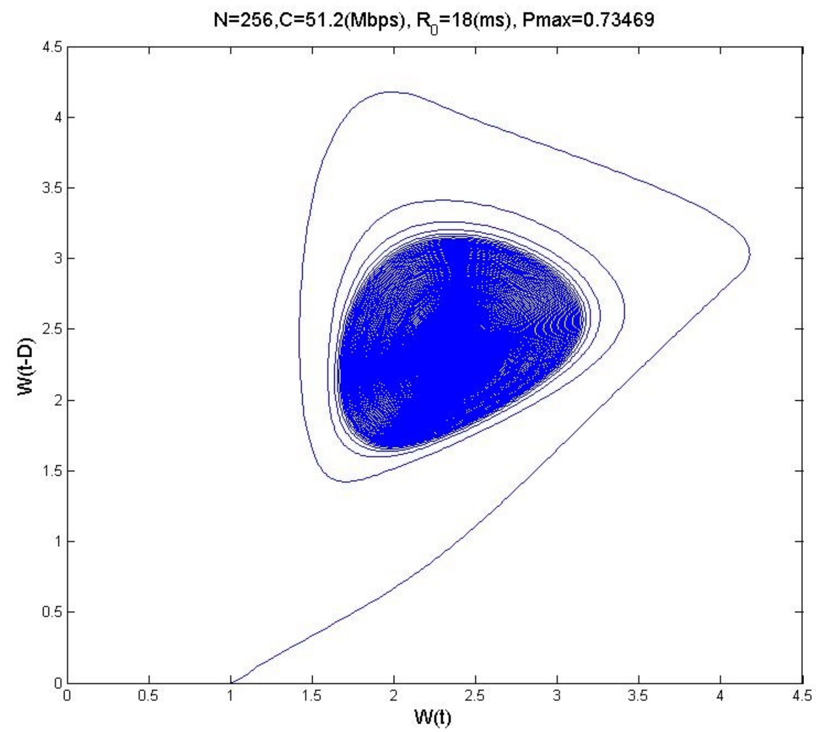
Stable queue length waveforms: Simulink



$K = 597.4$ packets ($r_0 = 51$ ms), $\Phi = 256$ packets, $\alpha = 0.01$

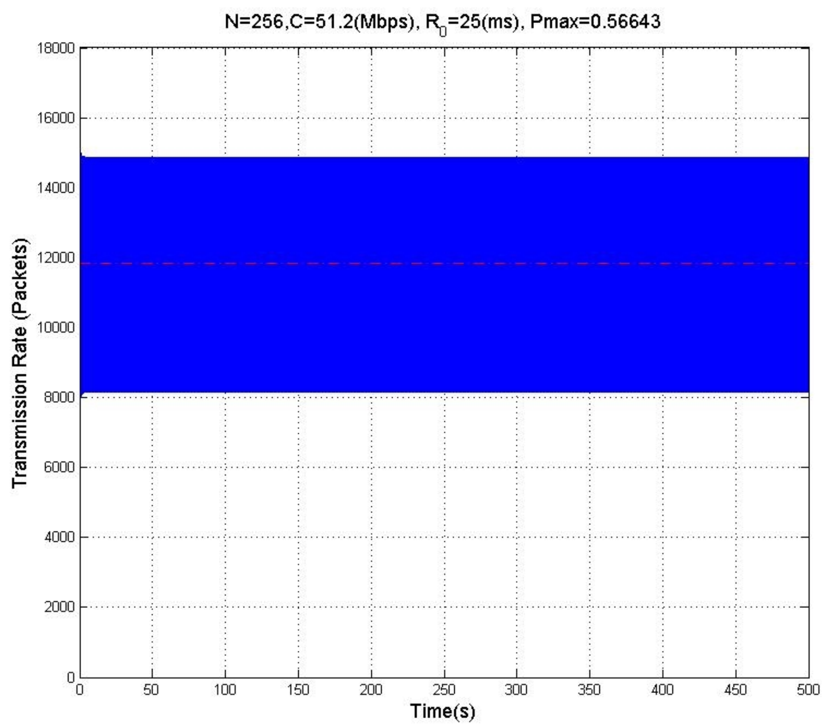
Stable queue length: Simulink

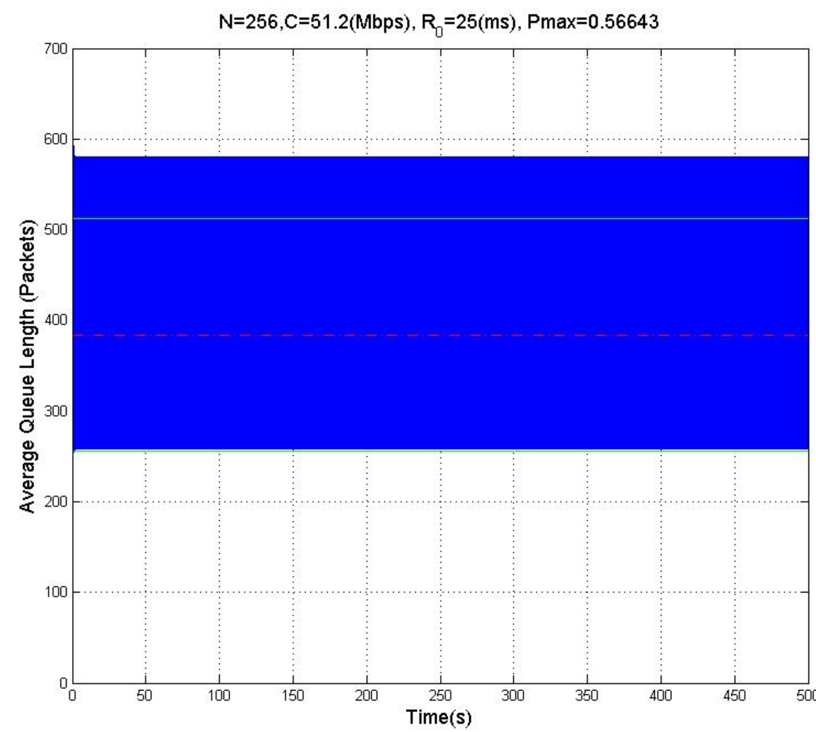




K = 597.4 packets (r_0 = 51 ms), Φ = 256 packets, α = 0.01

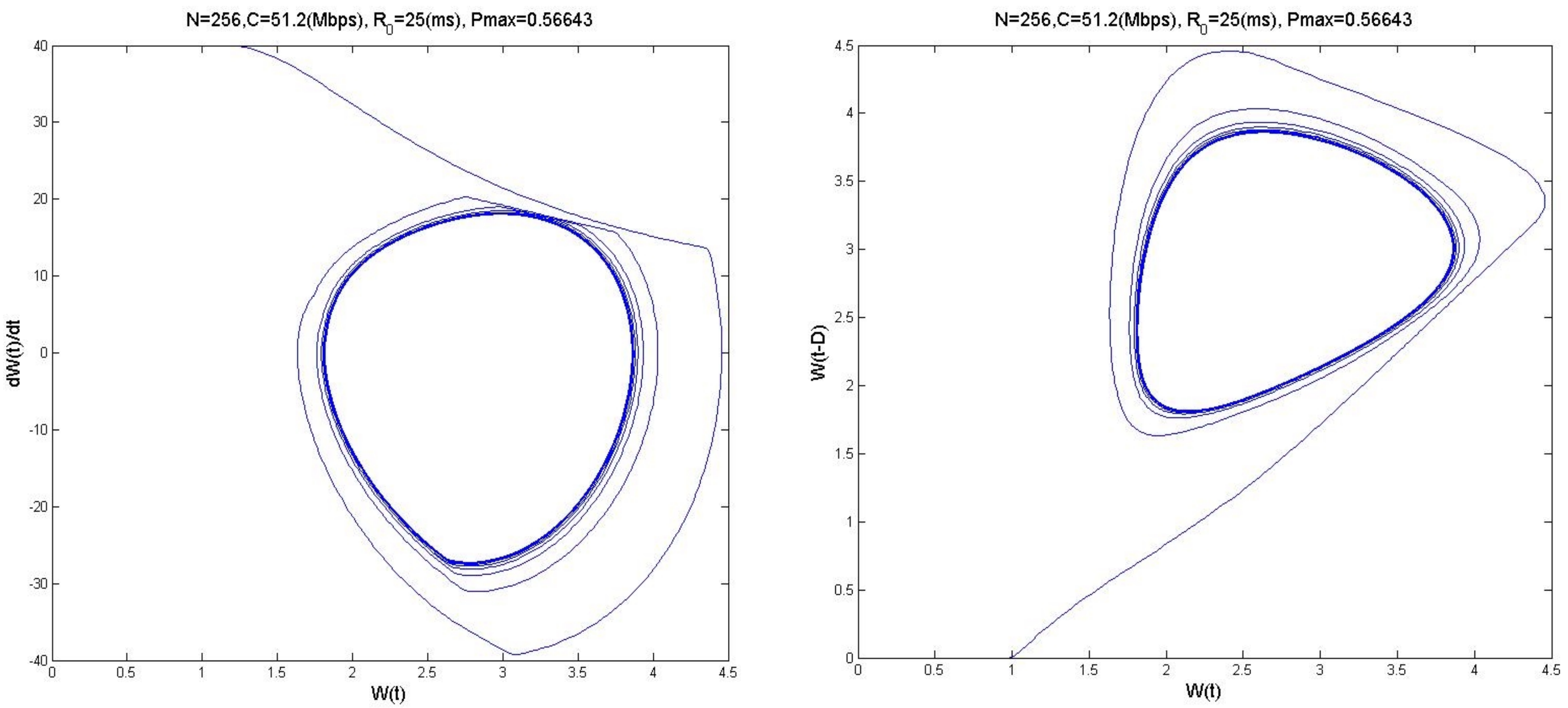
Unstable queue length waveforms: Simulink





K = 680.3 packets (r_0 = 58 ms), Φ = 256 packets, α = 0.01

Unstable queue length: Simulink

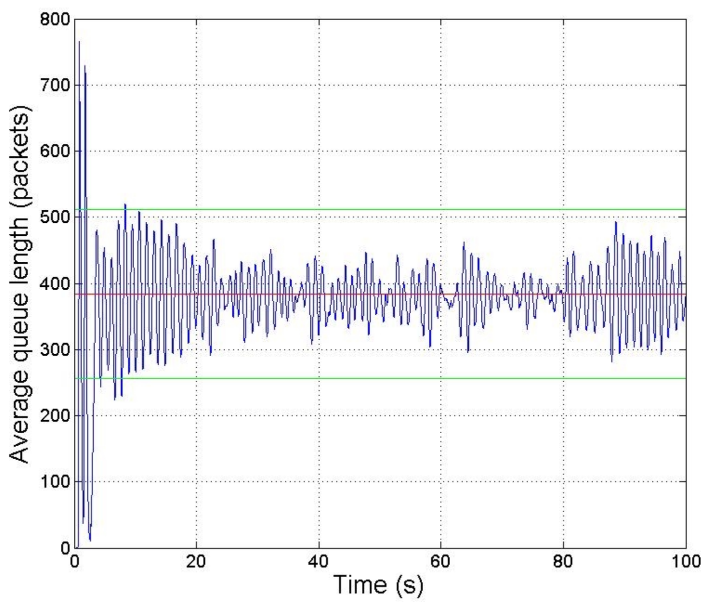


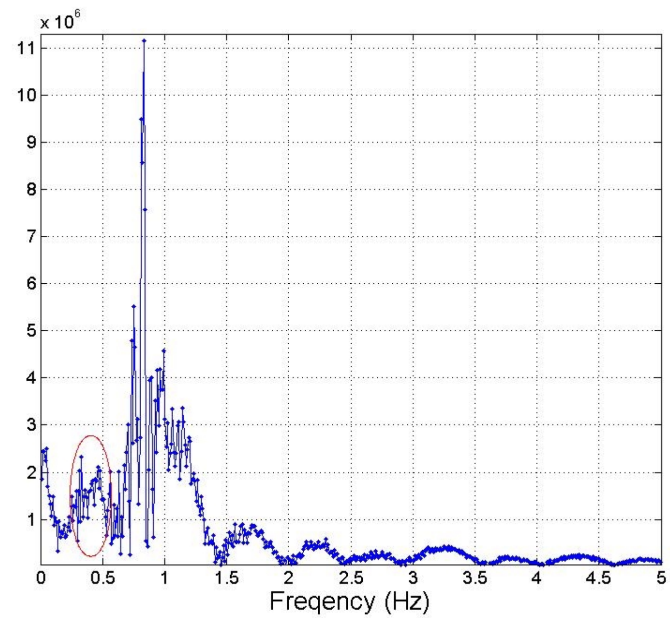
K = 680.3 packets (r_0 = 58 ms), Φ = 256 packets, α = 0.01

Appendix 2

Stable and unstable queue length waveform: ns-2 (2)

Stable or not ?



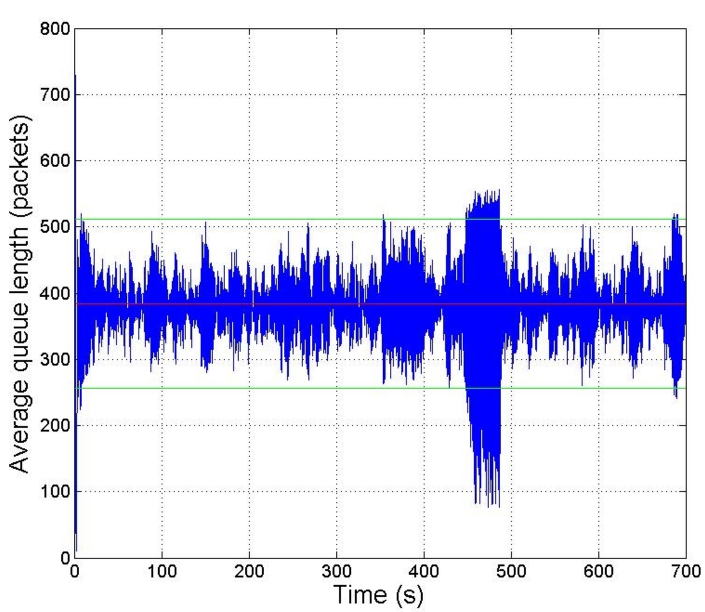


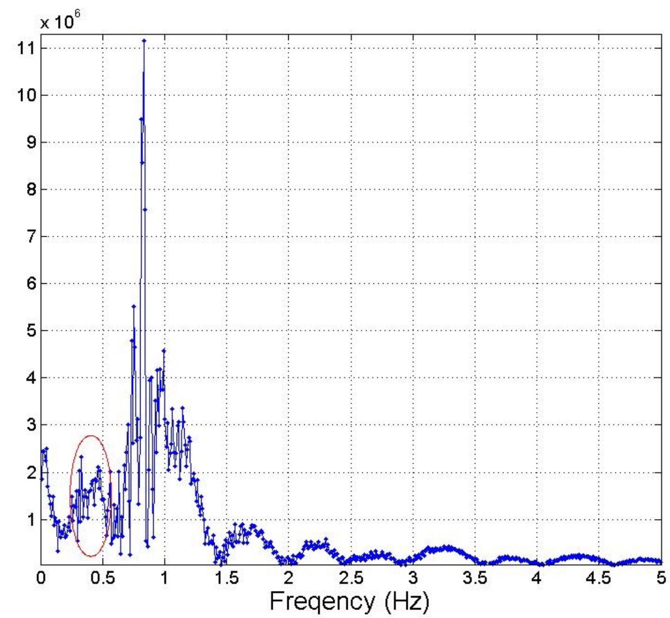
K = 1078.8 packets (r_0 = 155 ms), Φ = 256 packets, α = 0.001

FFT of average queue for first 100 seconds K = 1078.8 packets (r_0 = 155 ms), Φ = 256 packets, α = 0.001

Stable and unstable queue length waveform: ns-2 (2)

Unstable



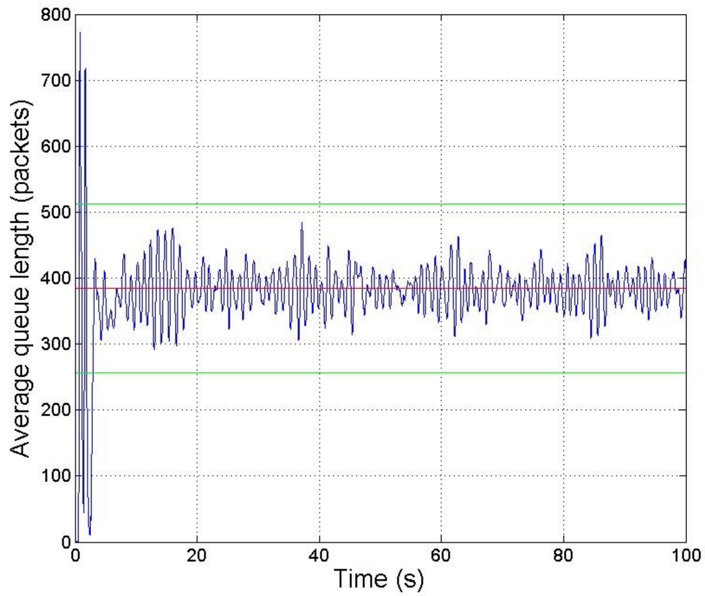


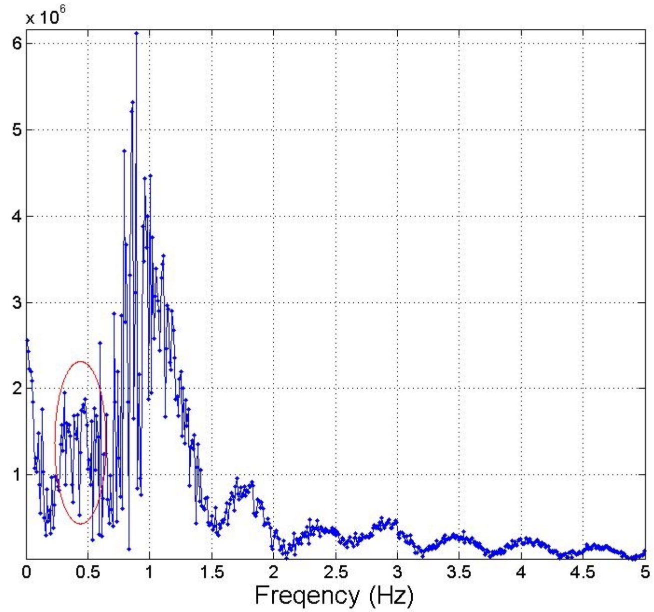
K = 1078.8 packets (r_0 = 155 ms), Φ = 256 packets, α = 0.001

FFT of average queue for first 100 seconds K = 1078.8 packets (r_0 = 155 ms), Φ = 256 packets, α = 0.001

Stable and unstable queue length waveform: ns-2 (2)

Stable or not?



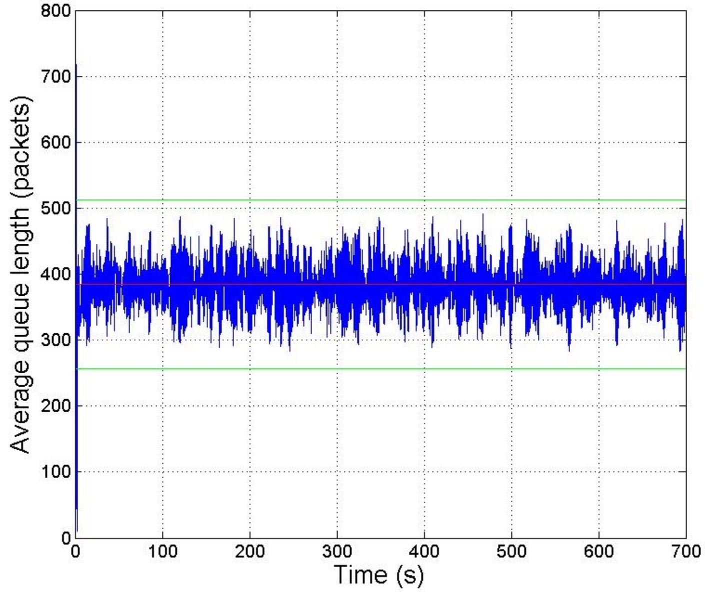


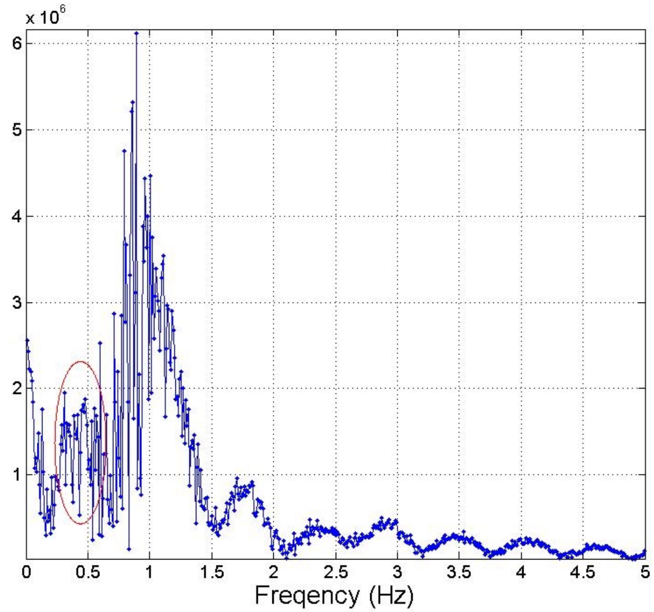
K = 1006.9 packets (r_0 = 145 ms), Φ = 256 packets, α = 0.001

FFT of average queue for first 100 seconds K = 1006.9 packets (r_0 = 145 ms), Φ = 256 packets, α = 0.001

Stable and unstable queue length waveform: ns-2 (2)

Stable





K = 1006.9 packets (r_0 = 145 ms), Φ = 256 packets, α = 0.001

FFT of average queue for first 100 seconds K = 1006.9 packets (r_0 = 145 ms), Φ = 256 packets, α = 0.001

Stability analysis: conclusions

- We have derived stability boundaries of the RED scheme based on an analytical closed-form solution using the Padé(1,1) linearized fluid flow models
- Very good match between the stability boundaries found from the Padé(1,1) linearized fluid flow model and ns-2 simulations has been achieved
- The model (verified by ns-2) can be used to predict dynamical behavior of the system

Roadmap

- Introduction
- TCP/RED congestion control algorithms: an overview
- Discrete-time dynamical models of TCP Reno with RED
- Discontinuity-induced bifurcations in TCP/RED
- Stability analysis of RED gateway with multiple TCP Reno connections
- Stability study of TCP/RED system using detrended fluctuation analysis
- Conclusions and References

Detrended fluctuation analysis (DFA)

DFA method was first proposed for determining the statistical self-affinity of a signal:

*C-K Peng, SV Buldyrev, S Havlin, M Simons, HE Stanley, and AL Goldberger, "Mosaic organization of DNA nucleotides," *Phys. Rev. E* 49: 1685-1689, 1994.

DFA method has been applied in the analysis of DNA nucleotides, fractals, electrocardiogram (ECG/ EKG), electroencephalography (EEG), climate, and stock market.

Detrended fluctuation analysis (DFA)

DFA method:

- permits the detection of intrinsic self-similarity embedded in a seemingly nonstationary time series
- it avoids the spurious detection of apparent self-similarity, which may be an artifact of extrinsic trends

Note:

- A *stationary* time series is characterized by its mean, standard deviation, higher moments, and correlation functions being invariant under time translation.
- Signals that are not stationary are *nonstationary*.

DFA computation

Consider signal s(i) of length of L

$$\bar{s} \equiv \frac{1}{L}\sum_{j=1}^{L}[s(j)]$$

Step 1: integrate s(i) and obtain $y(i) = \sum_{j=1}^{i}[s(j) - \bar{s}]$

Step 2: divide y(i) into boxes of equal length of l

Step 3: subtracting the local trend $y_l(i)$ for box of length l to detrended y(i):

$$Y_l(l) \equiv y(i) - y_l(i)$$

Step 4: for each box l, the characteristic size of fluctuation for the integrated and detrended time series is calculated by:

$$F(l) \equiv \sqrt{\frac{1}{L}\sum_{j=1}^{L}[Y_l(j)]^2}$$

Repeat Steps 2 - 4 for several length of l (different scales)

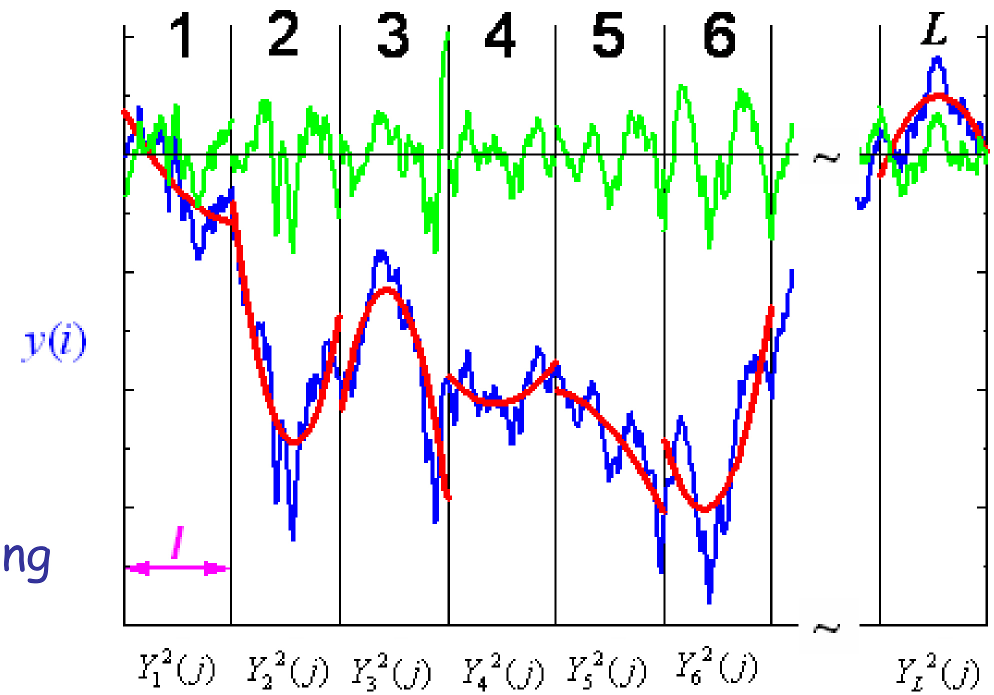
DFA method

1. Reconstruction of the signal

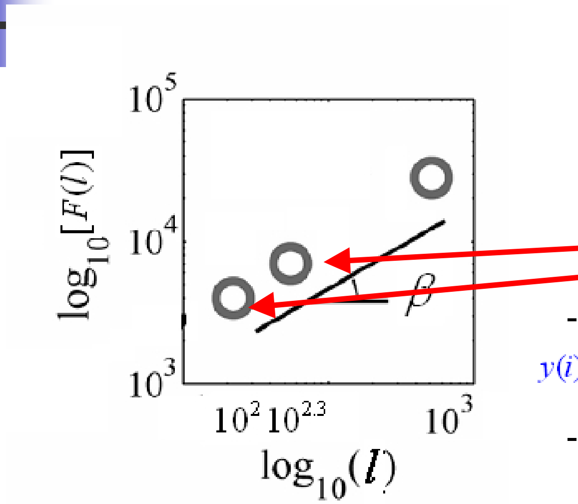
2. Splitting of the record into segments of scale length *l*

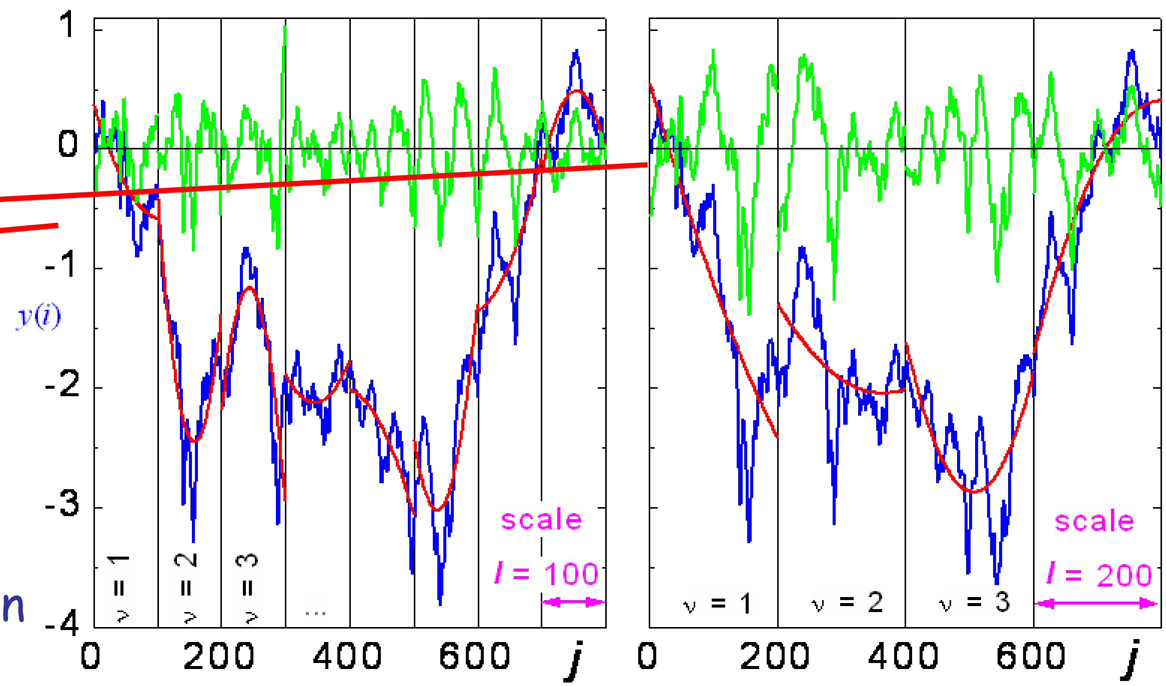
3. Regression in each segment

4 Calculation of variance *Y(j)* in each segment and then taking averaging to obtain *F(l)*



DFA method (cont.)





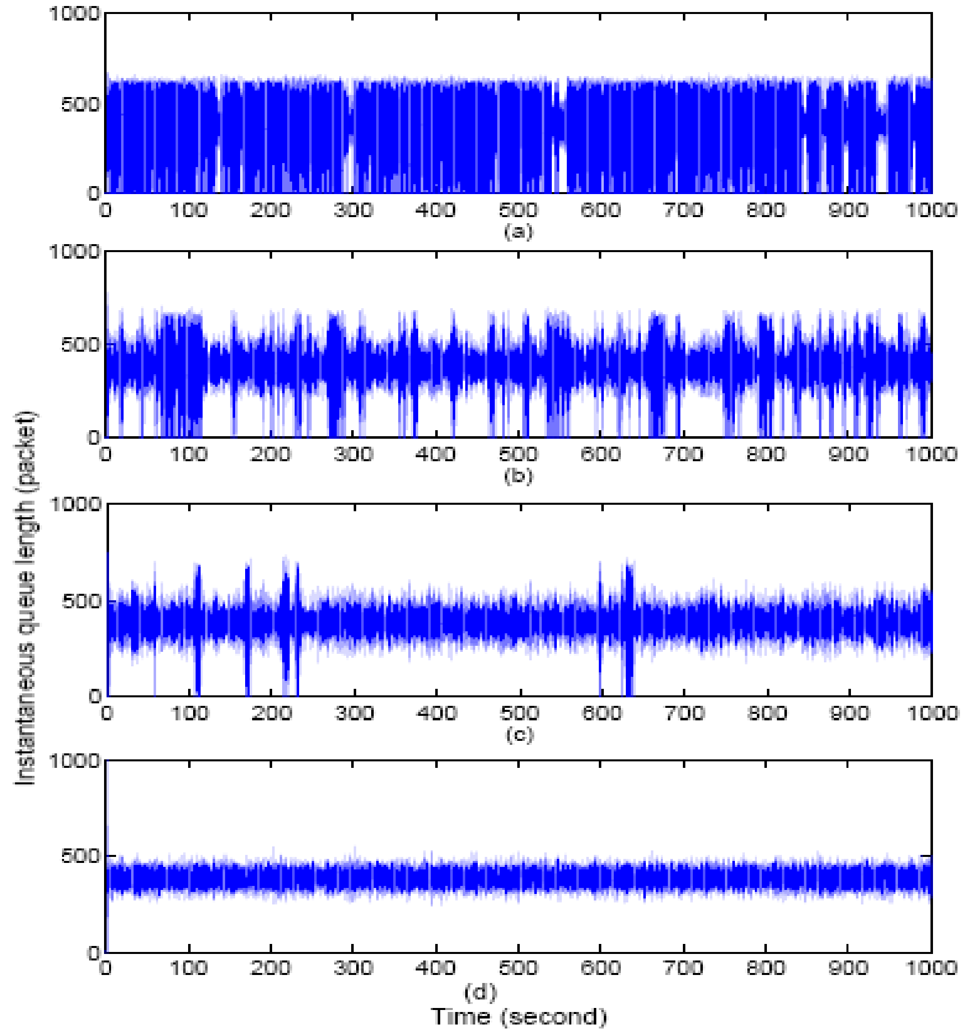
Scale invariant signals with power-law correlations between the scaling function F(l) and the scale l :

$$F(l) \propto l^{\beta}, \quad l \to \infty$$

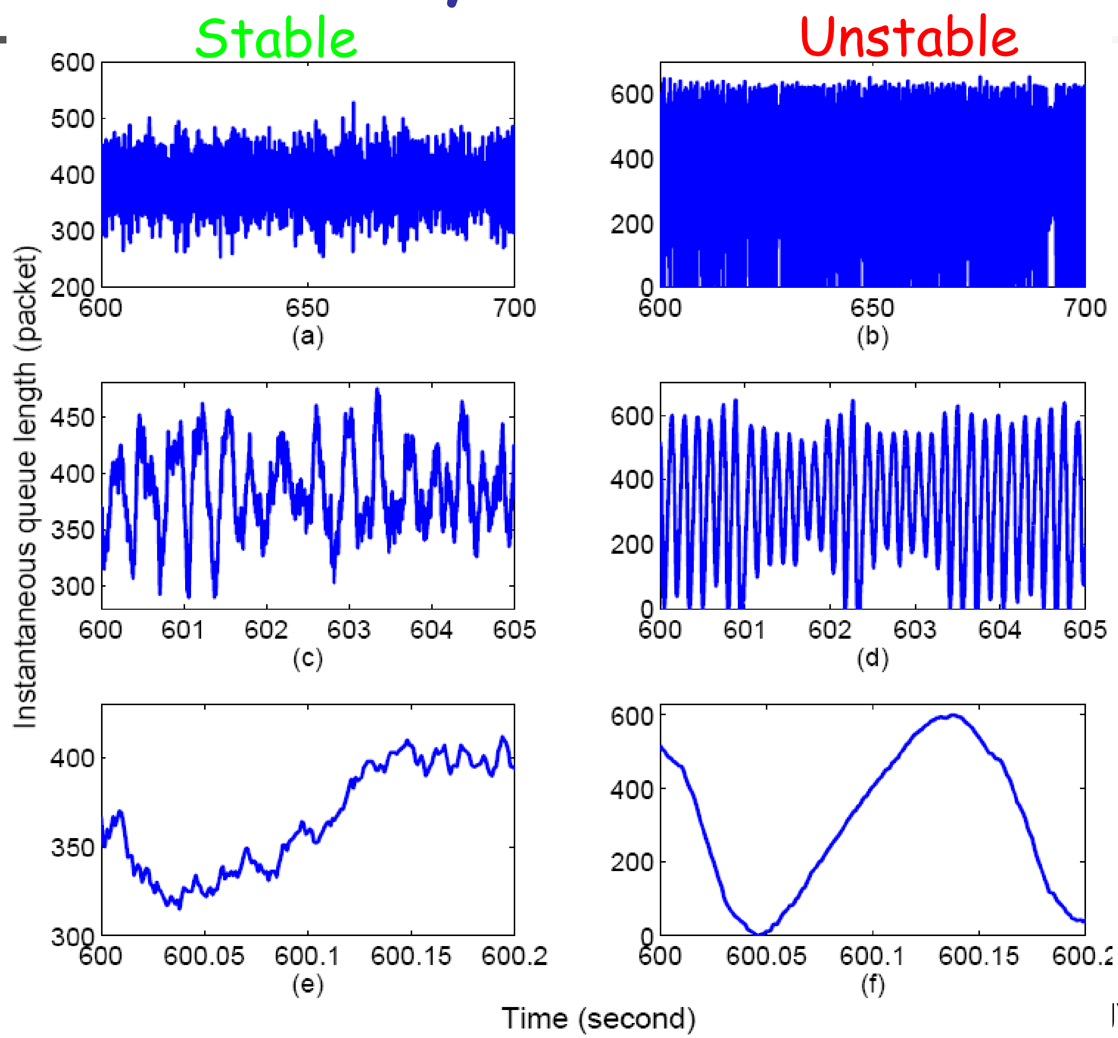
β represents the degree of the correlation

Non stationarity and instability in TCP/RED system

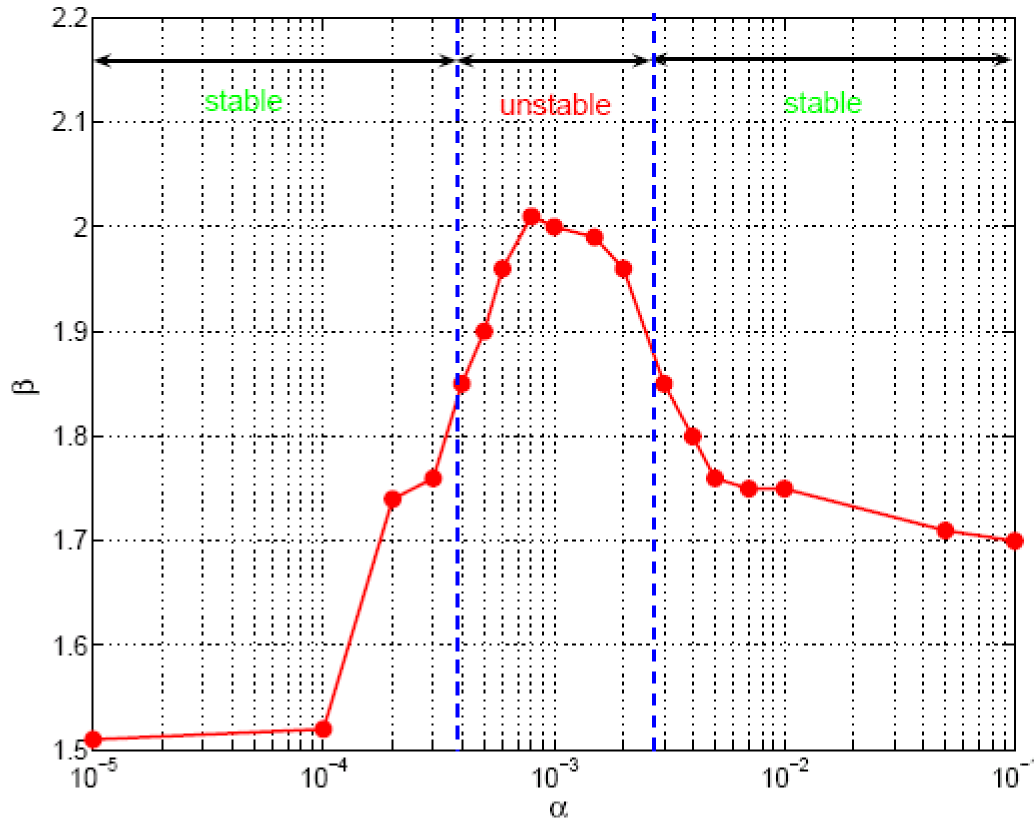
DFA method helps quantitatively describe the level of the instability of TCP/RED systems



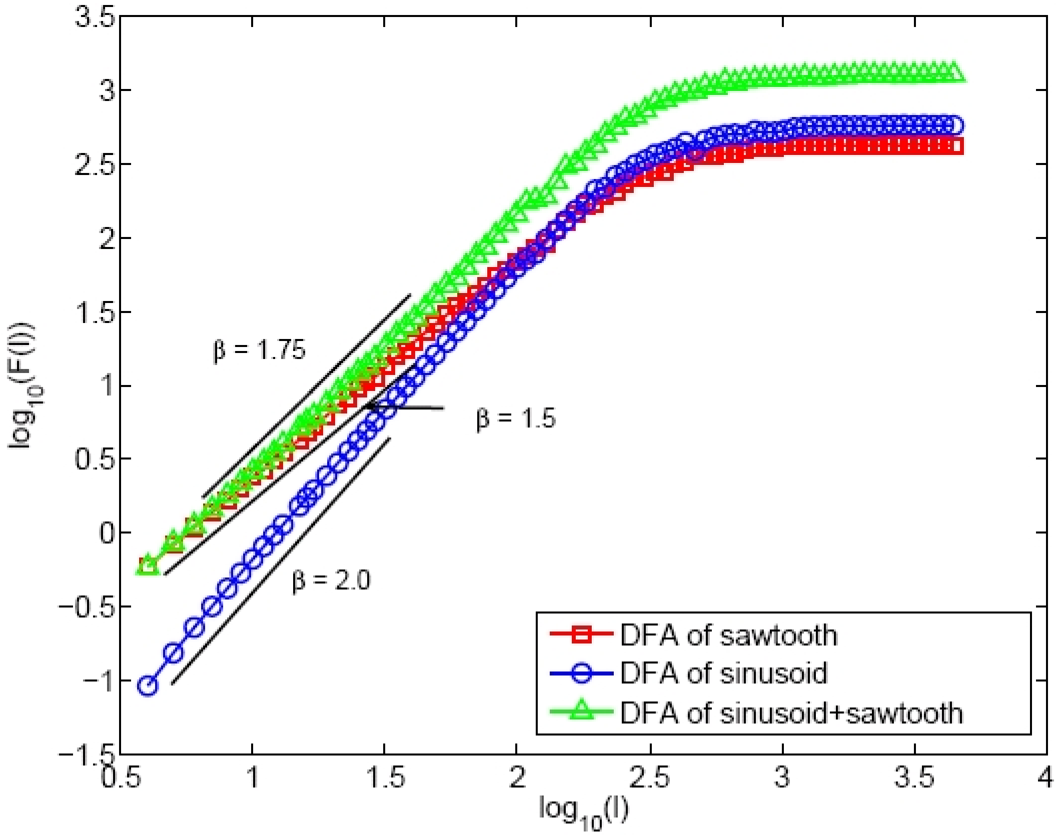
Self-similarity in TCP/RED system



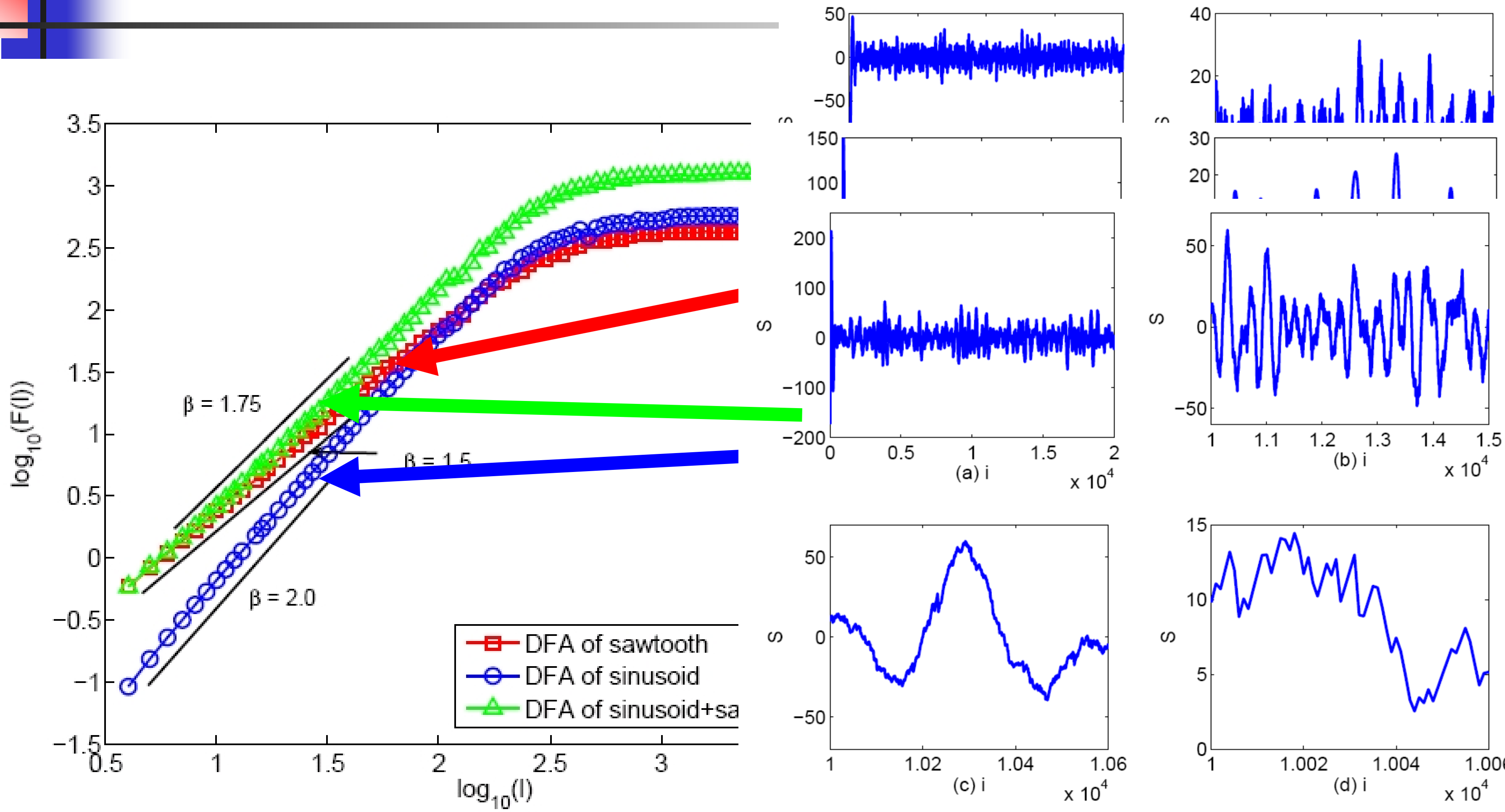
DFA results: instability



Interpretation of DFA: the waveform viewpoint



DFA exponents: sine and saw tooth



DFA method: conclusions

- Stability of the queue length has been explored using the DFA method
- The degree of instability can be described by DFA exponent, which varies with the relative stability of RED gateway
- We provided an interpretation of the relationship between the DFA exponent and the stability of RED system
- The DFA exponent may used as indicator for TCP/RED system and thus control the stability of the system

Roadmap

- Introduction
- TCP/RED congestion control algorithms: an overview
- Discrete-time dynamical models of TCP Reno with RED
- Discontinuity-induced bifurcations in TCP/RED
- Stability analysis of RED gateway with multiple TCP Reno connections
- Stability study of TCP/RED system using detrended fluctuation analysis
- References

References: TCP/RED and the model

[1] V. Jacobson, "Congestion avoidance and control," *ACM Computer Communication Review*, vol. 18, no. 4, pp. 314–329, Aug. 1988.

[2] M. Allman, V. Paxson, and W. Steven, "TCP congestion control," *IETF Request for Comments , RFC 2581*, Apr. 1999.

[3] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Trans. Networking*, vol. 1, no. 4, pp. 397–413, Aug. 1993.

[4] *V. Firoiu and M. Borden, "A study of active queue management for congestion control," in Proc. IEEE INFOCOM 2000, Tel-Aviv, Israel, Mar. 2000, vol. 3, pp. 1435–1444.*

[5] J. Padhye, V. Firoiu, and D. F. Towsley, "Modeling TCP Reno performance: a sample model and its empirical validation," *IEEE/ACM Trans. Networking*, vol. 8, no. 2, pp. 133-145, Apr. 2000.

[6] Khalifa and Lj. Trajkovic, "An overview and comparison of analytical TCP models," in *Proc. IEEE Int. Symp. Circuits and Systems*, Vancouver, BC, Canada, May 2004, vol. V, pp. 469-472.

References: bifurcations

[7] C. J. Budd and P. Piiroinen, "Corner bifurcations in nonsmoothly forced impact oscillators," to appear in *Physica D*, 2005.

[8] P. Jain and S. Banerjee, "Border-collision bifurcations in one-dimensional discontinuous maps," *Int. Journal Bifurcation and Chaos*, vol. 13. pp. 3341--3351, 2003.

[9] S. J. Hogan, L. Higham, and T. C. L. Griffin, "Dynamics of a piecewise linear map with a gap,'' to appear in *Proc. Royal Society*, London, 2005.

[10] P. Ranjan and E. H. Abed, "Bifurcation analysis of TCP-RED dynamics," in *Proc. ACC*, Anchorage, AK, USA, May 2002, pp. 2443--2448.

[11] P. Ranjan, E. H. Abed, and R. J. La, "Nonlinear instabilities in TCP- RED," in *Proc. IEEE INFOCOM 2002*, New York, NY, USA, June 2002, vol. 1, pp. 249–258.

[12] R. J. La, "Instability of a tandem network and its propagation under RED," *IEEE Trans. Automatic Control*, vol. 49, no. 6, pp. 1006-1011, June 2004.

[13] P. Ranjan, E. H. Abed, and R. J. La, "Nonlinear instabilities in TCP-RED," *IEEE/ACM Trans. on Networking*, vol. 12, no. 6, pp. 1079–1092, Dec. 2004.

References:

http://www.ensc.sfu.ca/~ljilja/publications_date.html

- M. Liu, H. Zhang, and Lj. Trajkovic, "Stroboscopic model and bifurcations in TCP/RED," in *Proc. IEEE Int. Symp. Circuits and Systems*, Kobe, Japan, May 2005, pp. 2060-2063.
- H. Zhang, M. Liu, V. Vukadinovic, and Lj. Trajkovic, "Modeling TCP/RED: a dynamical approach," *Complex Dynamics in Communication Networks*, Springer Verlag, Series: Understanding Complex Systems, 2005, pp. 251-278.
- M. Liu, A. Marciello, M. di Bernardo, and Lj. Trajkovic, "Discontinuity-induced bifurcations in TCP/RED communication algorithms," *Proc. IEEE Int. Symp. Circuits and Systems*, Kos, Greece, May 2006, pp. 2629-2632.
- X. Chen, S.-C. Wong, C. K. Tse, and Lj. Trajkovic, "Stability analysis of RED gateway with multiple TCP Reno connections," *Proc. IEEE Int. Symp. Circuits and Systems*, New Orleans, LA, May 2007, pp. 1429-1432.
- X. Chen, S.-C. Wong, C. K. Tse, and Lj. Trajkovic, "Stability study of the TCP-RED system using detrended fluctuation analysis," *Proc. IEEE Int. Symp. Circuits and Systems*, Seattle, WA, May 2008, pp. 324-327.