

Discontinuity-Induced Bifurcations in TCP/RED Communication Algorithms

Mingjian Liu
Simon Fraser University
Vancouver, Canada
jliu1@cs.sfu.ca

Alfredo Marciello and Mario di Bernardo
University of Naples Federico II
Naples, Italy
alfredomarciello@libero.it, mario.dibernardo@unina.it

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

Abstract—In this paper, we describe a simple second-order discrete-time model for the Transmission Control Protocol (TCP) with Random Early Detection (RED) algorithm. The TCP/RED mechanism is viewed as a feedback control system where TCP adjusts its sending rate depending on packet loss. We investigate bifurcations and chaos in a TCP/RED system with a single TCP connection. We conjecture that the complex behavior observed in the system is attributed to a class of discontinuity-induced bifurcations observed in piecewise smooth systems.

I. INTRODUCTION

Numerous Internet applications are delivered via the Transmission Control Protocol (TCP). Random Early Detection (RED) is an Active Queue Management (AQM) scheme employed along with TCP congestion control mechanisms in order to meet the increasing demand for better Internet performance.

An accurate model of TCP/RED algorithm helps predict the dynamical behavior of the network. We derive a second-order discrete-time model to capture interactions of the TCP congestion control algorithm with the RED mechanism. Based on the concepts reported in [1]–[3], we construct a nonlinear dynamical model of TCP/RED that employs two state variables: the window size and the average queue size [4]. The change of window size reflects the dynamics of TCP congestion control, while the average queue size captures the queue dynamics in RED gateway. The novelty of the proposed model is in capturing the detailed dynamical behavior of TCP/RED. The proposed model considers a slow start phase and takes into account timeout events common in TCP.

The resulting model is a piecewise smooth map with a gap. Such maps have been recently studied in [5]–[7]. It was shown that they can exhibit a novel type of so-called discontinuity-induced bifurcations [5] that may organize intricate bifurcation diagrams characterized by period-adding cascades and sudden transitions to higher-periodic orbits and chaotic attractors.

In this paper, we present the derivation of an improved TCP-RED model and analyze its bifurcation behavior. Bifurcation diagrams are shown that capture the dynamics of the average queue length under variations of typical parameters such as the maximum drop probability and the round-trip propagation

delay. It is suggested that the observed behavior is organized by a particular type of discontinuity-induced bifurcations: border-collisions of fixed points in two-dimensional piecewise-smooth maps with a gap.

The paper is organized as follows: in Sections II and III, after providing a brief overview of the TCP/RED algorithms, we describe the nonlinear second-order discrete-time TCP/RED model named S-model. In Section IV, we investigate the bifurcation and chaos phenomena and discuss the nature of the bifurcations observed in a TCP/RED system with a single connection. Conclusion is given in Section V.

II. TCP CONGESTION CONTROL ALGORITHMS: AN OVERVIEW

The TCP congestion control mechanism consists of four algorithms: slow start, congestion avoidance, fast retransmit, and fast recovery [8], [9].

An established TCP connection first employs the slow start algorithm to detect the available bandwidth in the network. Typically, a TCP sender initializes its congestion window ($cwnd$) to one or two segments, depending on the TCP implementation. Upon receipt of each acknowledgment (ACK) that confirms receipt of new data by the receiver, the TCP increases $cwnd$ by one segment size.

When $cwnd$ exceeds a threshold ($ssthresh$), the sender mechanism enters the congestion avoidance phase. During the congestion avoidance, $cwnd$ is incremented by one segment size per round-trip time (RTT). A timer is set every time a packet is sent. A packet loss is detected by the timeout mechanism if the timer expires before the receipt of the packet has been acknowledged. If a packet loss is detected by the timeout mechanism, the TCP sender adjusts its $ssthresh$ and switches back to the slow start.

The fast retransmission algorithm is used for recovery from losses detected by triple duplicate ACKs. Whenever a TCP destination receives an out-of-order segment, it immediately sends a duplicate ACK, which informs the sender of the sequence number of the packet that the receiver expects. The receipt of triple duplicate ACKs (four consecutive ACKs acknowledging the same packet) is used as an indication of packet loss. The TCP sender reacts to the packet loss by halving $cwnd$ and re-transmitting the lost packet, without waiting for the retransmission timer to expire.

This research was supported in part by the NSERC Grant 216844-03, Canada Foundation for Innovation New Opportunities program, and the EU-FP5 project SICONOS (IST-2001-37172).

The fast recovery algorithm is used to control data transmission after fast retransmission of the lost packet. During this phase, the TCP sender increases its $cwnd$ for each duplicate ACK received. The fast recovery algorithm recognizes each duplicate ACK as an indication that one packet has left the channel and has reached the destination. Since the number of outstanding packets has decreased by one, TCP sender is allowed to increment its $cwnd$. When a non-duplicate ACK is received, TCP switches from the fast recovery to the congestion avoidance phase.

A. RED Algorithm

The RED mechanism [10] calculates exponentially weighted moving average of the queue size. Let w_q be the weight factor and q_{k+1} be the current queue size. At every packet arrival, the RED algorithm updates the average queue size \bar{q}_{k+1} as:

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

If the average queue size is smaller than the minimum queue threshold q_{min} , the packet is admitted to the queue. If it exceeds the maximum queue threshold q_{max} , the packet is marked or dropped. If the average queue size is between q_{min} and q_{max} , the packet is dropped with a drop probability p :

$$p_{k+1} = \begin{cases} 0 & \text{if } \bar{q}_{k+1} \leq q_{min} \\ 1 & \text{if } \bar{q}_{k+1} \geq q_{max} \\ \frac{\bar{q}_{k+1} - q_{min}}{q_{max} - q_{min}} p_{max} & \text{otherwise} \end{cases}, \quad (2)$$

where p_{max} is the maximum packet drop probability.

III. DISCRETE-TIME DYNAMICAL MODEL OF TCP/RED

We model the TCP/RED system as a discrete-time map obtained by periodically sampling the system state. Since window size and queue size behave as step functions of RTT, one RTT is the sampling period that captures their changes [11].

Based on the concepts reported in [1]–[3], we proposed the S-model [4] that employs window size and average queue size as state variables. Variations of the window size reflect the dynamics of TCP congestion control. The window size increases exponentially and linearly in slow start phase and congestion avoidance phase, respectively. It multiplicatively decreases when packet loss occurs. The average queue size captures the queue dynamics in RED because it is updated upon every packet arrival. We do not consider instantaneous queue size as an independent state variable.

A. Modeling Assumptions

In order to construct an approximate model, we assume that ACK packets are never lost. The connection is long-lived and the source always has sufficient data to send. Round-trip propagation delay d between the source and destination and the data packet size M are constant. We also assume that the timeout is caused only by packet loss and that the duration of the timeout period is $5 \times RTT$ [11]. The state variables of

the system are sampled at the end of every RTT period. We assume that the queue size is constant during each sampling period. Depending on the probability of the number of lost packets in the previous RTT period, the model includes three cases: no loss, single loss, and multiple packet losses.

B. Case 1: No Loss

Let W_k , q_k , and \bar{q} be the window size, queue size, and average queue size at the end of the sampling period k . If no packet is dropped during the last RTT period, TCP Reno increases its window size. The window size is increased exponentially in the slow start phase and linearly in the congestion avoidance phase:

$$W_{k+1} = \begin{cases} \min(2W_k, ssthresh) & \text{if } W_k < ssthresh \\ \min(W_k + 1, rwnd) & \text{if } W_k \geq ssthresh \end{cases}, \quad (3)$$

where $rwnd$ is the receiver advertised window size, i.e., the largest window size that the receiver could accept in one round. Usually, $rwnd$ is greater than window size, and hence $rwnd$ does not affect its variations. If the window size increases linearly and reaches the value $rwnd$, then the window size is kept at $rwnd$ until loss occurs.

The average queue size (1), may be calculated from the queue size at the sampling period $k + 1$. This queue size depends on the queue size in the previous period, the current window size, and the number of packets that have left the queue during the previous sampling period. Therefore, the current queue size is:

$$\begin{aligned} q_{k+1} &= q_k + W_{k+1} - \frac{C \cdot RTT_{k+1}}{M} \\ &= q_k + W_{k+1} - \frac{C}{M} \left(d + \frac{q_k \cdot M}{C} \right), \\ &= W_{k+1} - \frac{C \cdot d}{M} \end{aligned}, \quad (4)$$

where

- $q_{k+1} \doteq$ instantaneous queue size in round $k + 1$
- $q_k \doteq$ instantaneous queue size in round k
- $W_{k+1} \doteq$ current TCP window size in round $k + 1$
- $RTT_{k+1} = d + \frac{q_k \cdot M}{C} \doteq$ round-trip time in round $k + 1$
- $C \doteq$ link capacity
- $M \doteq$ packet size
- $d \doteq$ round-trip propagation delay.

Substituting q_{k+1} in (1) gives the average queue size:

$$\bar{q}_{k+1} = (1 - w_q)\bar{q}_k + w_q \cdot \max(W_{k+1} - \frac{C \cdot d}{M}, 0). \quad (5)$$

The RED algorithm updates the average queue size at every packet arrival. Hence, \bar{q} is updated W_{k+1} times during the current sampling period. We assume that the queue size is constant during each period and that \bar{q} is given as:

$$\bar{q}_{k+1} = (1 - w_q)^{W_{k+1}} \bar{q}_k + (1 - (1 - w_q)^{W_{k+1}}) \cdot \max(W_{k+1} - \frac{C \cdot d}{M}, 0). \quad (6)$$

Finally, if $p_k W_k < 0.5$, which implies that no packet loss occurred in the previous sampling period, the state variables of the model are:

$$\begin{aligned}
W_{k+1} &= \begin{cases} \min(2W_k, ssthresh) & \text{if } W_k < ssthresh \\ \min(W_k + 1, rwnd) & \text{if } W_k \geq ssthresh \end{cases} \\
\bar{q}_{k+1} &= (1 - w_q)^{W_{k+1}} \bar{q}_k + (1 - (1 - w_q)^{W_{k+1}}) \\
&\quad \cdot \max(W_{k+1} - \frac{C \cdot d}{M}, 0).
\end{aligned} \tag{7}$$

C. Case 2: One Packet Loss

If $0.5 \leq p_k W_k < 1.5$, which implies that one packet loss occurred in the previous RTT period, the congestion control mechanism of TCP Reno halves the window size in the current sampling period. The average queue size is updated in a manner similar to Case 1:

$$\begin{aligned}
W_{k+1} &= \frac{1}{2} W_k \\
\bar{q}_{k+1} &= (1 - w_q)^{W_{k+1}} \bar{q}_k + (1 - (1 - w_q)^{W_{k+1}}) \\
&\quad \cdot \max(W_{k+1} - \frac{C \cdot d}{M}, 0)
\end{aligned} \tag{8}$$

D. Case 3: At Least Two Packet Losses

In this case $p_k W_k \geq 1.5$, which implies that at least two packets are lost in the previous RTT period. When multiple packets are lost from the same window, TCP Reno may not be able to send a sufficient number of new packets in order to receive three duplicate ACKs for each packet lost. TCP source will often have to wait for the timeout before retransmitting the lost packet. During the timeout period, the source does not send packets into the network. In the S-model, the window size is equivalent to the number of packets that are sent by the source during one RTT period. Hence, we assume that the window size is zero during the timeout period.

The RED mechanism updates the average queue size for each packet arrival. However, during timeout period there are no packet arrivals. The average queue size is not updated and has the same value as in the previous RTT period. RED takes this “idle time” period into account when it updates the average queue size upon the next packet arrival. However, the S-model does not take into account the “idle time”. Thus, the TCP/RED system during the timeout period is modeled as:

$$\begin{aligned}
W_{k+1} &= 0 \\
\bar{q}_{k+1} &= \bar{q}_k
\end{aligned} \tag{9}$$

IV. BIFURCATION AND CHAOS PHENOMENA IN TCP/RED

TCP/RED models were shown to exhibit bifurcations and chaos for appropriate choices of system parameters [12]–[15]. The bifurcation behavior of a one-dimensional piecewise-smooth continuous model of TCP/RED was investigated in [15]. Here, we employ bifurcation diagrams to investigate the nonlinear behavior and dynamics of the S-model, described by a two-dimensional piecewise-smooth *discontinuous* map.

We investigate the dynamical behavior of the average queue size in a TCP/RED system with a single connection when RED parameters, such as the weight factor w_q , the maximum packet drop probability p_{max} , the round-trip propagation delay d , the minimum queue threshold q_{min} , and the maximum queue threshold q_{max} , are varied. Other system parameters

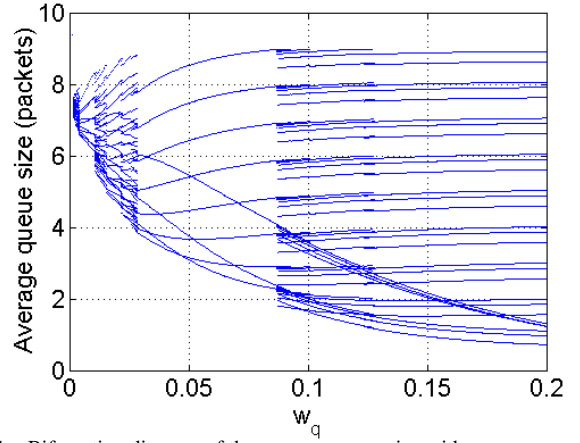


Fig. 1. Bifurcation diagram of the average queue size with w_q as a parameter ($p_{max} = 0.1$, $q_{min} = 5$, $q_{max} = 15$, and $ssthresh = 80$).

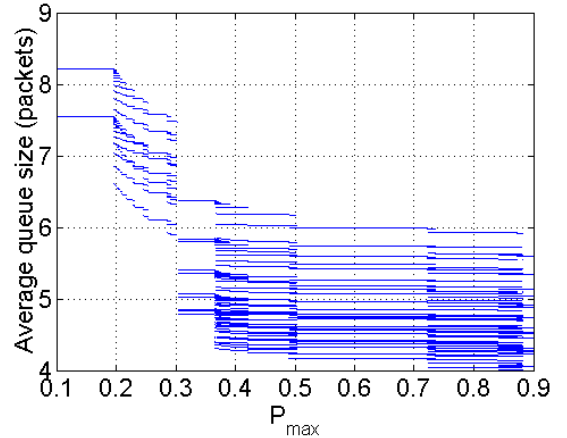


Fig. 2. Bifurcation diagram of the average queue size with p_{max} as a parameter ($w_q = 0.01$, $q_{min} = 5$, $q_{max} = 15$, and $ssthresh = 20$).

are: $C = 1.54$ Mbps, $d = 0.0228$ sec, $M = 4,000$ bits, and $rwnd = 1,000$ packets.

A. Bifurcation Parameter: RED Weight Factor w_q

The bifurcation diagram of the TCP/RED system transiting to chaos when the RED parameter w_q varies from 0.001 to 0.2, with step 0.001, is shown in Fig. 1. For small w_q , the system exhibits chaos with the average queue size oscillating around 7.3 packets. The system then exhibits bifurcations to different high-periodic regimes.

B. Bifurcation Parameter: Maximum Drop Probability p_{max}

The bifurcation diagram shown in Fig. 2 is generated when the RED maximum drop probability p_{max} varies from 0.1 to 0.9, with step 0.001. A stable period-2 region is observed for $p_{max} \in [0.1, 0.2]$, when the average queue size is either 7.5 or 8.2 packets. As p_{max} increases, the system exhibits a *devil staircase* (predicted in [5] for maps with a gap), before locking to higher-periodic solutions for larger p_{max} implying that more packets are dropped (marked) as indicated by large variations of the average queue size.

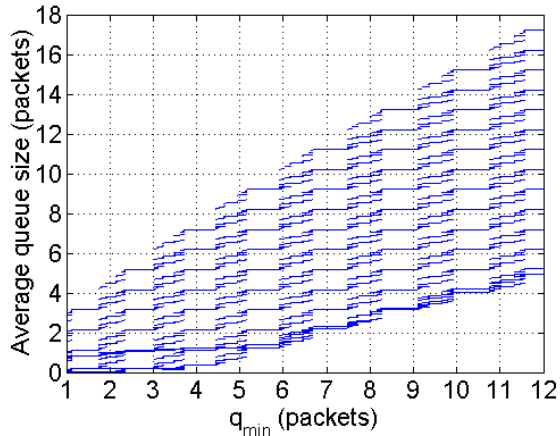


Fig. 3. Bifurcation diagram of the average queue size with q_{min} and q_{max} as a parameter ($w_q = 0.01$, $p_{max} = 0.1$, $q_{max} = 3 \cdot q_{min}$, and $ssthresh = 20$).

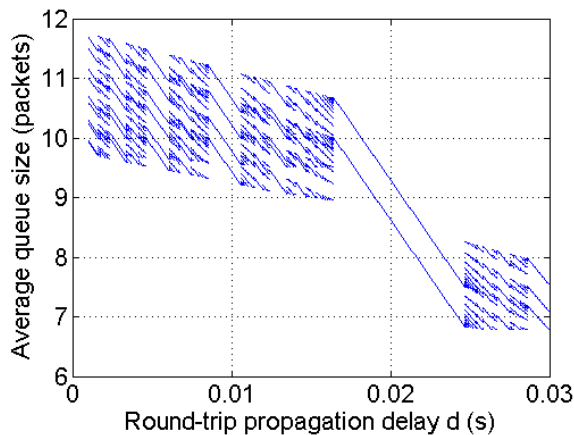


Fig. 4. Bifurcation diagram of the average queue size with round-trip propagation delay d as a parameter ($w_q = 0.01$, $p_{max} = 0.1$, $q_{min} = 5$, $q_{max} = 15$, and $ssthresh = 20$).

C. Bifurcation Parameters: RED Minimum and Maximum Queue Thresholds q_{min}

We also considered the RED queue thresholds q_{min} and q_{max} as bifurcation parameters, producing bifurcation diagrams shown in Fig. 3. Period-adding cascades with orbits of increasing period alternate with bands of more complex behavior. This behavior was observed in a two-dimensional discontinuous map described in [5].

D. Bifurcation Parameters: RED Round-Trip Propagation Delay d

Bifurcation diagram when the RED parameter d , which frequently fluctuates in deployed TCP/RED systems, varies from 0.001 s to 0.03 s, is shown in Fig. 4. The queue dynamics locks on several periodic attractors interleaved with ranges where cascades of bifurcations leading to higher periodic solutions may be detected.

E. Discontinuity-induced bifurcations

The bifurcation phenomena observed in the analyzed TCP/RED model show a striking resemblance to those recently

observed in a class of piecewise smooth systems. In particular, it was shown [5] that *corner bifurcations* in impact oscillators may be associated with bifurcation diagrams characterized by period-adding cascades, devil staircases, and phenomena similar to bifurcations described in this paper. The observed phenomena were explained in terms of border-collisions of fixed points of two-dimensional piecewise linear maps with a gap. (The case of such one-dimensional maps was investigated in [6].) Thus, we conjecture that the bifurcation behavior of the TCP/RED model may also be explained in terms of such discontinuity-induced bifurcations occurring when the map fixed point crosses one of the phase space discontinuity sets.

V. CONCLUSIONS

We described a nonlinear second-order discrete model for TCP Reno with the RED algorithm. The model takes into account the slow start and timeout events and captures the dynamical behavior of TCP/RED in terms of the average queue size. We reported the numerical bifurcation analysis of the model in a TCP/RED system with a single connection showing the presence of complex behavior. It was conjectured that the behavior observed can be explained in terms of a novel class of bifurcations of piecewise-smooth maps with a gap.

REFERENCES

- [1] 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.
- [2] I. Khalifa and Lj. Trajković, "An overview and comparison of analytical TCP models," in *Proc. IEEE ISCAS*, Vancouver, BC, Canada, May 2004, vol. V, pp. 469–472.
- [3] M. Liu, H. Zhang, and Lj. Trajković, "Stroboscopic model and bifurcations in TCP/RED," *Proc. IEEE ISCAS*, Kobe, Japan, May 2005, pp. 2060–2063.
- [4] H. Zhang, M. Liu, V. Vukadinović, and Lj. Trajković, "Modeling TCP/RED: a dynamical approach," in *Complex Dynamics in Communication Networks*, G. Vattay and L. Kocarev, Eds., Springer Verlag, Series: Understanding Complex Systems, 2005, pp. 251–278.
- [5] C. J. Budd and P. Piiroinen, "Corner bifurcations in nonsmoothly forced impact oscillators," submitted to *Physica D*, 2005.
- [6] P. Jain and S. Banerjee, "Border-collision bifurcations in one-dimensional discontinuous maps," *Int. Journal Bifurcation and Chaos*, vol. 13, pp. 3341–3351, 2003.
- [7] S. J. Hogan, L. Higham and T. C. L. Griffin, "Dynamics of a piecewise linear map with a gap," to appear on *Proc. Royal Society*, London, 2005.
- [8] V. Jacobson, "Congestion avoidance and control," *ACM Computer Communication Review*, vol. 18, no. 4, pp. 314–329, Aug. 1988.
- [9] M. Allman, V. Paxson, and W. Stevens, "TCP congestion control," *IETF Request for Comments, RFC 2581*, Apr. 1999.
- [10] 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.
- [11] V. Firoiu and M. Borden, "A study of active queue management for congestion control," in *Proc. IEEE INFOCOM*, Tel Aviv, Israel, Mar. 2000, vol. 3, pp. 1435–1444.
- [12] P. Ranjan and E. H. Abed, "Bifurcation analysis of TCP-RED dynamics," in *Proc. ACC*, Anchorage, AK, USA, May 2002, pp. 2443–2448.
- [13] P. Ranjan, E. H. Abed, and R. J. La, "Nonlinear instabilities in TCP-RED," in *Proc. IEEE INFOCOM*, New York, NY, USA, June 2002, vol. 1, pp. 249–258.
- [14] 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.
- [15] P. Ranjan, E. H. Abed, and R. J. La, "Nonlinear instabilities in TCP-RED," *IEEE/ACM Trans. Networking*, vol. 12, no. 6, pp. 1079–1092, Dec. 2004.