Stroboscopic Model and Bifurcations in TCP/RED

Mingjian Liu, Hui Zhang, and Ljiljana Trajković
Simon Fraser University
Vancouver, British Columbia, Canada
{jliu1, hzhange, ljilja}@cs.sfu.ca

Abstract— **In this paper, we derive a simple first-order discrete-time model for the Transmission Control Protocol (TCP) with Random Early Detection (RED). We view the network as a discrete feedback control system where TCP adjusts its sending rate depending on whether or not it has detected a packet loss. We then use a stroboscopic map to investigate bifurcations and chaos in a TCP/RED system with a single connection.**

I. Introduction

One important goal of modeling the Transmission Control Protocol (TCP) [1], [2] is to investigate its nonlinear behavior. We use an iterative map to derive a simple first-order discrete-time model that captures the interactions between the TCP congestion control algorithm and the Random Early Detection (RED) mechanism [3]. We use the concepts proposed in [4] and construct a nonlinear dynamical model of TCP/RED that employs the average queue size as a state variable. The average queue size captures the queue dynamics in the RED gateway and reflects the dynamics of the TCP congestion control mechanism. The novelty of the proposed model is in its simplicity and ability to capture the detailed dynamical behavior of TCP/RED systems. The model was verified via ns-2 simulations.

We also investigate the bifurcation and chaos phenomena [5] in a TCP/RED system with a single connection. These phenomena are easily observed in the derived TCP/RED iterative map. To visualize chaos and bifurcation, we use bifurcation diagrams by varying a range of RED parameters such as the weight factor, maximum packet drop probability, and minimum and maximum queue thresholds. Of particular interest is the observed "double-bifurcation" phenomenon: the existence of chaos regions for both small and large values of the RED queue thresholds.

The paper is organized as follows: in Section II, we briefly describe TCP congestion control and the RED algorithm. The nonlinear first-order discrete-time model is introduced in Section III. In Section IV, we investigate the bifurcation and chaos phenomena in a TCP/RED system with a single connection. Conclusions are given in Section V.

II. TCP and RED algorithms

A. TCP Congestion Control Algorithms

To adjust the window size, TCP congestion control mechanism [1], [2] employs four algorithms: slow start, congestion avoidance, fast retransmit, and fast recovery, as shown in Fig. 1. Older TCP implementations used a simple window-based congestion control (early 1980s). TCP Tahoe employed the slow start, congestion avoidance, and fast retransmit algorithms (late 1980s). The fast recovery algorithm was introduced in TCP Reno (early 1990s).

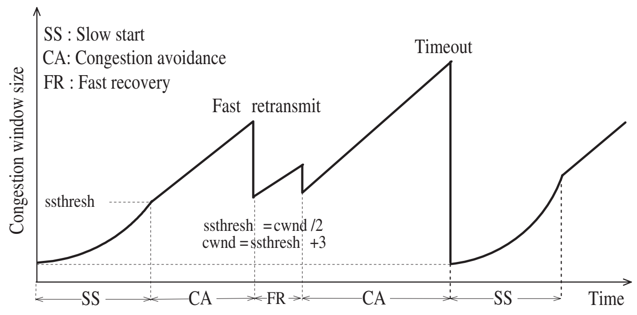

Fig. 1. Evolution of the window size in TCP Reno. It consists of slow start, congestion avoidance, fast retransmit, and fast recovery phases.

In order to avoid congesting the network with large bursts of data, an established TCP connection first employs the slow start algorithm to detect the available network bandwidth. Typically, the TCP sender initializes its congestion window *cwnd* to one or two segments, depending on the TCP implementation. Upon receiving a new acknowledgment (ACK) from the receiver, TCP increments the *cwnd* by one segment size.

When *cwnd* exceeds a threshold (*ssthresh*), the sender's congestion control mechanism leaves the slow start and enters the congestion avoidance phase. During the congestion avoidance phase, *cwnd* is incremented by one segment size per round trip time (RTT). A retransmission timer is set every time the sender sends a packet. A packet loss is detected by the timeout mechanism if the timer expires before the receipt of the packet has been acknowledged. In this case, the TCP sender adjusts its *ssthresh* and switches back to slow start.

The fast retransmit algorithm is used for recovery from losses detected by triple duplicate ACKs. Whenever a TCP receiver receives an out-of-order segment, it immediately sends a duplicate ACK that 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 the *cwnd* and re-transmitting the lost packet without waiting for the retransmission timer to expire.

This research was supported by the NSERC Grant 216844-03 and Canada Foundation for Innovation New Opportunities program.

The fast recovery algorithm is used to control data transmission after the 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 reached the destination. Since the number of outstanding packets has decreased by one, the 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.

B. RED Algorithm

The traditional DropTail queue management mechanism drops the most recently arrived packet if the queue is full. However, this method has two drawbacks. First, it may allow few connections to monopolize the queue space so that other flows are starved. Second, DropTail allows queues to be full for a long period of time. During that period, incoming packets are dropped in bursts. This causes severe reduction in the throughput of the TCP flows. One solution is to deploy active queue management (AQM) algorithms [3]. The purpose of AQM is to react to incipient congestion before the queue overflows. Active queue management allows responsive flows, such as TCP flows, to react timely and reduce their sending rates in order to prevent congestion and severe packet losses.

The most widely implemented active queue management algorithm is RED [3]. The RED mechanism calculates an exponentially weighted moving average of the queue size. At every packet arrival, the RED gateway updates the average queue size as:

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

where w_q is the weight factor and q_{k+1} is the current queue size. The average queue size is compared to two parameters: minimum queue threshold q_{min} and maximum queue threshold q_{max}. If the average queue size is smaller than q_{min}, the packet is admitted to the queue. If it exceeds 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_{k+1} that is a linear function of the average queue size:

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

III. Modeling TCP/RED

Analytical TCP models may be classified in three categories based on the duration of the TCP flows. Flow duration determines the dominant TCP congestion control algorithms to be modeled and the aspects of TCP performance that may be captured by the model [6], [7]. The first category models short-lived flows, where TCP performance is strongly affected by the connection establishment and slow start phases. These models typically approximate the average latency, defined as the time necessary to complete a transfer of a certain amount of data. The second category models long-lived flows that characterize the steady-state performance of bulk TCP transfers during the congestion avoidance phase. These models approximate aspects such as the average throughput and window size evolution. The final category includes models for flows of arbitrary duration and may accommodate both short-lived and long-lived flows.

From the control theory point of view, developed models of TCP and TCP/RED may be classified into two types: averaged models and iterative map models. An averaged model is described by a set of continuous differential equations. It neglects the detailed dynamics and only captures the "low-frequency characteristics" of the system. It may be used to analyze the steady-state performance and to predict low-frequency slow-scale bifurcation behavior, such as Hopf bifurcations. In contrast, an iterative map model has a discrete-time form and employs a set of difference equations. It provides relatively complete dynamical information. Iterative maps are adequate to explore nonlinear phenomena, such as period-doubling and saddle-node bifurcations, which may appear across a wide spectrum of frequencies and cause existence of solutions in the high-frequency range.

A. Discrete-Time Dynamical Model of TCP/RED

The basic idea behind RED is to sense impending congestion before it occurs and to try to provide feedback to senders by either dropping or marking packets. Hence, from the control theory point of view, the network may be considered as a complex feedback control system. TCP adjusts its sending rate depending on whether or not it has detected a packet drop in the previous RTT interval. The drop probability of RED may be considered as a control law of the network system. The discontinuity in the drop probability function is the main reason for oscillations and chaos in the system. Hence, it is natural to model the network system as a discrete-time model. In this paper, we model TCP/RED systems using a "stroboscopic map", which is the most widely used type of discrete-time maps for modeling power converters [5]. This map is obtained by periodically sampling the system state. In our model, the sampling period is one RTT. Window and queue sizes behave as step functions of RTT. Hence, one RTT is the sampling period that captures their changes [4]. Higher sampling rate would not significantly improve the accuracy of the model, whereas lower sampling rate would ignore the changes and would affect the model accuracy.

B. Stroboscopic Model of TCP/RED

We employ the average, rather than the instantaneous, queue size as a state variable in the proposed model. The average queue size captures the behavior of TCP and the queue dynamics of RED. We consider a simple network consisting of N TCP sources, N destinations, and two routers, with a link between the routers. RED mechanism is employed in the router connected to the sources. TCP Reno connections are established between the sources and the destinations. Data packets are sent from a source to a destination, while traffic in the opposite direction consists of ACK packets only.

We made several assumptions in order to construct a simple TCP/RED model. We assume that ACK packets are never lost. The connections are long-lived and the sources always have sufficient data to send. The link that connects the two routers is the only bottleneck in the network. All connections are TCP Reno with identical and constant round trip propagation delay and data packet size. The receivers' advertised window size (*rwnd*), the largest amount of data that a receiver could accept in one round, is sufficiently large and it does not influence TCP sending rate. We also assume that timeouts are caused only by packet loss and that the duration of the timeout period is 5 RTTs [4].

The state variable of the system is sampled at the end of every RTT period. We assume that the queue size is constant during each sampling period. The average queue size (1) depends on the instantaneous queue size q_{k+1} at the sampling period $k+1$. q_{k+1}, in turn, depends on the remaining queue size q_k from the previous sampling period and the number of incoming and outgoing packets in the current round.

In our derivation, we employ the following variables:

$\bar{q}_{k+1} \doteq$ average queue size in round $k+1$
$\bar{q}_k \doteq$average queue size in round k
$w_q \doteq$ queue weight in RED
$N \doteq$ number of TCP connections
$K \doteq$ constant $(\sqrt{3/2})$ [6]
$p_k \doteq$ drop probability in round k
$C \doteq$ capacity of the link between the two routers
$d \doteq$ round-trip propagation delay
$M \doteq$ packet size
$rwnd \doteq$ receiver's advertised window size.

The model includes two cases, depending on the loss probability in the previous RTT period.

Case 1: Drop probability $p_k \neq 0$:

$$\begin{aligned} q_{k+1} &= q_k + B(p_k) \cdot RTT_{k+1} \cdot N - \tfrac{C \cdot RTT_{k+1}}{M} \\ &= q_k + \tfrac{K}{\sqrt{p_k} \cdot RTT_{k+1}} \cdot RTT_{k+1} \cdot N - \tfrac{C}{M}(d + \tfrac{q_k M}{C}) \\ &= \tfrac{K \cdot N}{\sqrt{p_k}} - \tfrac{C \cdot d}{M}, \end{aligned} \tag{3}$$

where $B(p_k)$ is the TCP sending rate [6]. $B(p_k) \cdot RTT_{k+1} \cdot N$ and $C \cdot RTT_{k+1}/M$ are the number of incoming and outgoing packets in round $k+1$, respectively. Substituting q_{k+1} in (1) gives:

$$\bar{q}_{k+1} = (1 - w_q) \cdot \bar{q}_k + w_q \cdot \max(\frac{N \cdot K}{\sqrt{p_k}} - \frac{C \cdot d}{M}, \ 0). \tag{4}$$

Case 2: Drop probability $p_k = 0$:

$$\begin{aligned} q_{k+1} &= q_k + B(p_k) \cdot RTT_{k+1} \cdot N - \tfrac{C \cdot RTT_{k+1}}{M} \\ &= q_k + \tfrac{rwnd}{RTT_{k+1}} \cdot RTT_{k+1} \cdot N - \tfrac{C}{M}(d + \tfrac{q_k M}{C}) \\ &= rwnd \cdot N - \tfrac{C \cdot d}{M}. \end{aligned} \tag{5}$$

Substituting q_{k+1} (5) into (1) gives the average queue size:

$$\bar{q}_{k+1} = (1 - w_q) \cdot \bar{q}_k + w_q \cdot (rwnd \cdot N - \frac{C \cdot d}{M}). \tag{6}$$

Finally, the dynamic model of TCP/RED is:

$$\bar{q}_{k+1} = \begin{cases} (1 - w_q) \cdot \bar{q}_k + w_q \cdot \max(\frac{N \cdot K}{\sqrt{p_k}} - \frac{C \cdot d}{M}, \ 0) \text{ if } p_k \neq 0 \\ (1 - w_q) \cdot \bar{q}_k + w_q \cdot (rwnd \cdot N - \frac{C \cdot d}{M}) \text{ if } p_k = 0. \end{cases} \tag{7}$$

C. Model Validation

The accuracy of the proposed model is validated by comparing its performance with results obtained from ns-2 simulations. The simulated network consists of one source, one sink, and two routers, with a bottleneck link between the two routers. System parameters are: $C = 1.54$ Mbps, $M = 4,000$ bits, $N = 1$, $w_q = 0.002$, $p_{max} = 0.1$, $q_{min} = 5$ packets, and $q_{max} = 15$ packets. The average values of the queue size during steady state are 5.71 (model) and 5.77 (ns-2 simulations), indicating a difference of $\sim 1.05\%$. The proposed model is also validated by varying system parameters: queue weight w_q, drop probabilities p_{max}, and thresholds q_{min} and q_{max}. The average queue size during the steady state for $w_q \in [0.001, \ 0.01]$ is shown in Fig. 2.

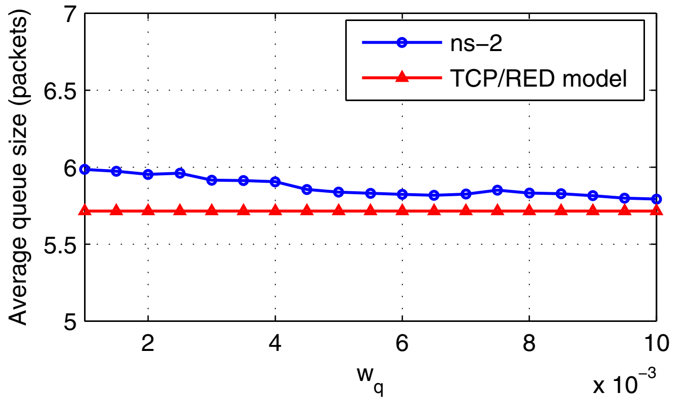

Fig. 2. Comparison of the average queue size for various w_q.

IV. Bifurcation and Chaos Phenomena in TCP/RED

We investigate bifurcation and chaos in the TCP/RED iterative map that we derived. Similar observations have been reported in a TCP/RED system with multiple connections [8], [9]. Chaos and bifurcation phenomena may be visualized using time-waveforms of the state variables, phase portraits, Poincare or first-return maps, and bifurcation diagrams [5]. We employ here bifurcation diagrams to investigate these nonlinear phenomena. They capture a periodic steady state of the system by recording the periodicity of the system for a fixed parameter. In the case of chaotic systems, numerous points are plotted indicating infinity period because the points never fall at the same position. We investigate the change of behavior 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 minimum queue threshold q_{min}, and the maximum queue threshold q_{max}, are varied. Other system parameters are: $C = 1.54$ Mbps, $K = \sqrt{3/2}$, $d = 0.0228$ sec, $M = 4,000$ bits, $rwnd = 1,000$ packets, and $N = 1$.

A. Bifurcation Parameter: RED Weight Factor w_q

We vary w_q from 0.14 to 0.27, with step 0.001. The bifurcation diagram is shown in Fig. 3. The TCP/RED system transits to chaos via a period-doubling route. A small periodic window is embedded in the chaos region. It exhibits period-doubling cascade (starting with period-3) when $w_q \in [0.244,\ 0.246]$.

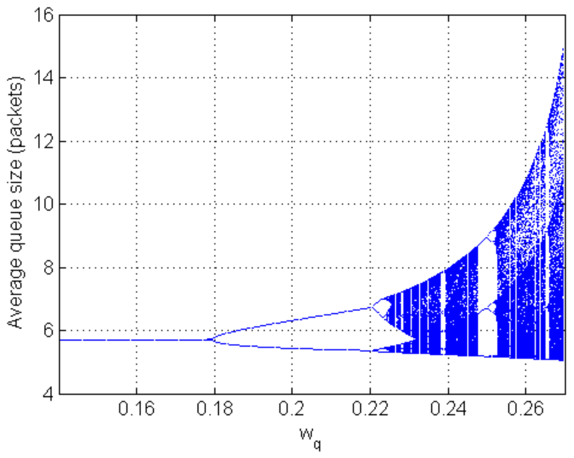

Fig. 3. Bifurcation diagram of the average queue size with w_q as a parameter ($p_{max} = 0.1$, $q_{min} = 5$, and $q_{max} = 15$).

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

The RED maximum drop probability p_{max} is varied from 0.1 to 0.9, with step 0.001, producing a bifurcation diagram shown in Fig. 4. The TCP/RED system exhibits a period-doubling route to chaos. A small periodic window embedded in the chaos region is observed when $p_{max} \in [0.76,\ 0.79]$.

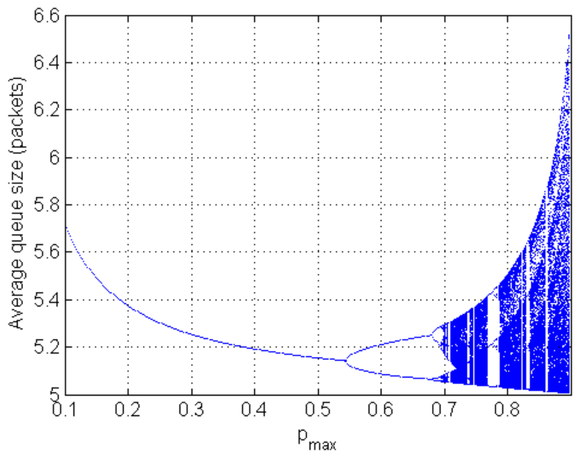

Fig. 4. Bifurcation diagram of the average queue size with p_{max} as a parameter ($w_q = 0.04$, $q_{min} = 5$, and $q_{max} = 15$).

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

The RED queue thresholds q_{min} and q_{max} may also serve as bifurcation parameters, producing similar bifurcation diagrams. Of particular interest is the diagram shown in Fig. 5. The TCP/RED system bifurcates and enters chaos via "double-bifurcation". It exhibits chaos for both small and large values of q_{min}.

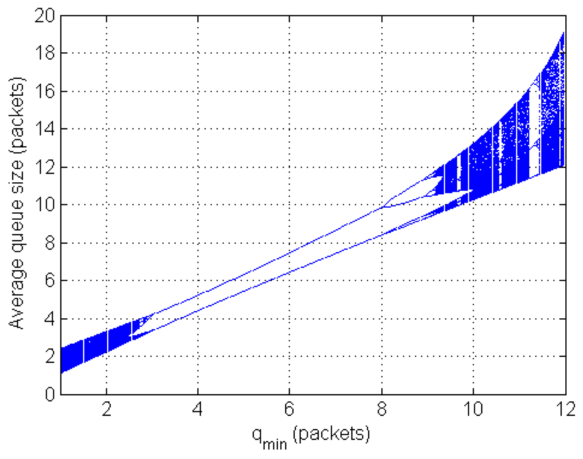

Fig. 5. Bifurcation diagram of the average queue size with q_{min} and q_{max} as parameters ($w_q = 0.2$, $p_{max} = 0.1$, $q_{max} = 3 \cdot q_{min}$).

V. Conclusions

In this paper, we developed a nonlinear first-order discrete model for TCP Reno with the RED algorithm. TCP/RED is a feedback control system where TCP adjusts its sending rate depending on the packet loss probability determined by the RED mechanism. The proposed 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 validated the model for various RED parameters by comparing its performance with ns-2 simulation results. The model was used to observe the bifurcation and chaos phenomena in a TCP/RED system with a single connection.

References

[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. Stevens, "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*, Tel Aviv, Israel, Mar. 2000, vol. 3, pp. 1435–1444.

[5] M. di Bernardo and C. K. Tse, "Chaos in power electronics: an overview," *Chaos in Circuits and Systems*, New York: World Scientific, pp. 317–340, 2002.

[6] 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.

[7] 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.

[8] 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.

[9] R. J. La, P. Ranjan, and E. H. Abed, "Nonlinearity of TCP and instability with RED," in *Proc. SPIE ITCom, Internet Performance and Control of Network Systems III*, Boston, MA, USA, July 2002, vol. 4865, pp. 283–294.