

# Analysis and Control of Bifurcation and Chaos in Averaged Queue Length in TCP/RED Model

J.H.C. Nga and H.H.C. Iu\*

*School of Electrical, Electronic and Computer Engineering,  
The University of Western Australia, Australia*

*\*herbert@ee.uwa.edu.au*

B.W.K. Ling and H.K. Lam

*Department of Electronic Engineering, Division of Engineering, King's College London,  
WC2R 2LS, United Kingdom*

Received May 10, 2006; Revised November 23, 2006

This paper studies the bifurcation and chaos phenomena in averaged queue length in a developed Transmission Control Protocol (TCP) model with Random Early Detection (RED) mechanism. Bifurcation and chaos phenomena are nonlinear behaviour in network systems that lead to degradation of the network performance. The TCP/RED model used is a model validated previously. In our study, only the average queue size  $\bar{q}_k$  is considered, and the results are based on analytical model rather than actual measurements. The instabilities in the model are studied numerically using the conventional nonlinear bifurcation analysis. Extending from this bifurcation analysis, a modified RED algorithm is derived to prevent the observed bifurcation and chaos regardless of the selected parameters. Our modification is for the simple scenario of a single RED router carrying only TCP traffic. The algorithm neither compromises the throughput nor the average queuing delay of the system.

*Keywords:* Congestion Control; Transmission Control Protocol; Random Early Detection; Nonlinear phenomenon; Bifurcation; Chaos.

## 1. Introduction

The unprecedented growth of Internet, in terms of size and traffic, has given rise to the congestion control issue. Poor management of congestion may lead to a network being partly or fully inaccessible and degrade the performance of networking applications. Therefore, various approaches have been proposed over the years to address this issue. One of the approaches is to control the congestion level at each router through active queue management (AQM) mechanisms [Barden et al., 1998]. Random Early Detection (RED) mechanism is a widely deployed AQM algorithm. The goal of the RED mechanism, proposed by Floyd and Jacobson [Floyd et al., 1993], is to detect early signs of congestion and provide feedback by either dropping or marking their packets to the adaptive sources so that congestion can be avoided. The RED mechanism is conceptually simple. However, the interaction between Transmission Control Protocol (TCP) and RED at the router's gateway has at times induced strange behaviour. Such behaviour has been generally recognized as nonlinear phenomena. They lead to instability of the system and

degradation of the network performance. An accurate model of the nonlinear phenomena in TCP/RED systems is essential to investigate the problem [La, 2004], seek out the stability margins of the system and provide guidelines for selecting network system parameters and RED controller's parameters [Wang et al., 2005], or establish possible solutions to the problem [Ranjan et al., 2004a].

This paper studies the problem of bifurcation and chaos phenomena in averaged queue length in a TCP/RED model. Bifurcation and chaos are nonlinear behaviour in network systems. The TCP/RED model used is a model validated previously by Liu et al. [Liu et al., 2005] and the instabilities in the model are studied numerically using the conventional nonlinear bifurcation analysis [Ranjan et al., 2004b]. We also derive a modified RED algorithm to control the observed bifurcation and chaos.

The dynamics of the averaged queue size is important to the network engineers. The information may help understand and predict the dynamical behaviour of the network. In addition, it may help analyze the stability margins of the system and provide design guidelines for selecting network parameters and improving network robustness [Zhang et al., 2005].

This paper is organized as follows: In Section 2, we describe the TCP congestion control and the RED algorithm. In Section 3, we briefly describe the TCP/RED model under investigation. Section 4 explains the origin of the nonlinear phenomena in TCP/RED models. The simulation results of bifurcation and chaos phenomena are presented in Section 5. Section 6 discusses the modified RED algorithm and section 7 presents the simulation results. Section 8 presents the limitations of our model and Section 9 concludes the investigation.

## **2. TCP And RED Algorithms**

### **2.1. TCP Congestion Control Algorithm**

The transmission rate of a TCP connection is controlled by its congestion window (*cwnd*) size at the sender end. The *cwnd* determines the number of segments to be sent to the receiver end. The *cwnd* size is constantly adjusted to maximize the link utilization and to avoid congesting the network with large bursts of data at the same time. To adjust the window size, TCP congestion control algorithms employ four algorithms: slow start, congestion avoidance, fast retransmit, and fast recovery [Zhang et al., 2005; Allman et al., 1999; Jacobson, 1988, 1990]. The algorithms are shown in Figure 1.

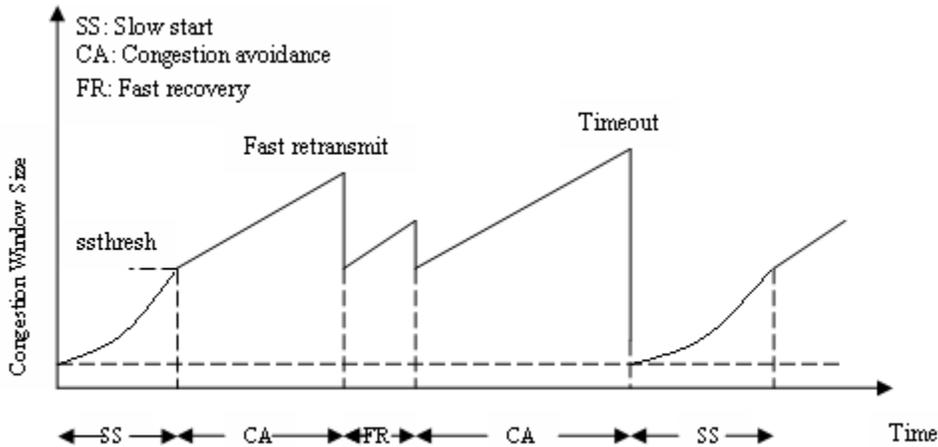


Fig 1: Evolution of window size in TCP congestion control algorithms.

Figure 2 is a simple topology of a TCP network. When a new connection is first established, the *cwnd* at the sender end is initialized to one segment. Upon receipt of every segment, an acknowledgement (ACK) packet is sent by the receiver to the sender. Upon receipt of every ACK, the TCP sender increases the *cwnd* by one segment. Two segments of data can now be sent. When each of those two segments is acknowledged (2 ACKs received by the sender), the *cwnd* is increased to four. This is the slow start (SS) phase. TCP sender opens up the window exponentially,  $1 \rightarrow 2 \rightarrow 4 \rightarrow 8 \rightarrow \dots$ , etc.

When *cwnd* exceeds a threshold *ssthresh*, the sender enters the congestion avoidance (CA) phase. During CA phase, *cwnd* is incremented by one segment size per round trip time (RTT), regardless of the number of ACKs received. TCP sender opens up the window linearly, i.e.  $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \dots$  until it reaches the receiver's advertised window size (*rwnd*). 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 phase.

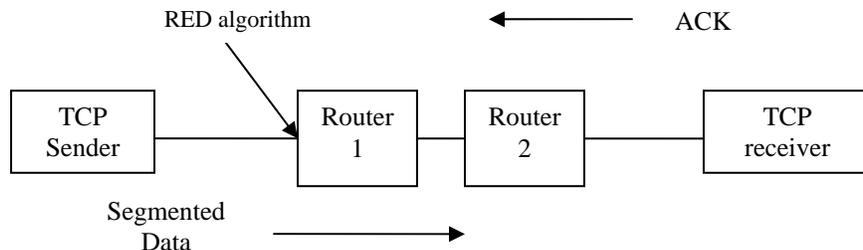


Fig 2: A simple topology of a TCP network.

In CA phase, upon receiving an out-of-order segment, TCP receiver generates an ACK, and immediately followed by a duplicate ACK. For each subsequent out-of-order segment, the receiver generates a duplicate ACK. When three duplicate ACKS have been received by the sender, it is assumed that a segment has been lost. TCP sender halves the

*cwnd* and re-transmitting the lost segment without waiting for a retransmission timer to expire. The algorithm is called the fast retransmit algorithm.

Until the retransmitted segment is received, receiver will continue to receive out-of-order segments and generate duplicate ACKs to the sender. After fast retransmit sends the missing segment, TCP sender increases its *cwnd* for each duplicate ACK received. Each duplicate ACK is an indication that one packet has reached the destination and the number of outstanding packets has decreased by one, and therefore, the TCP sender is allowed to increment its *cwnd*. This is the fast recovery algorithm. TCP switches back to CA phase when the retransmitted segment is received and a non-duplicate ACK is sent to the sender. Further details of TCP congestion control algorithm are illustrated by Allman et al. [Allman et al., 1999].

## 2.2. RED Algorithm

RED algorithm is a gateway based active queue management algorithm [Floyd & Jacobson, 1993; Firoiu & Borden, 2000]. It calculates an exponentially weighted moving average of the queue size at the router, as depicted in Figure 2. Upon arrival of each segment, the RED gateway updates the average queue size as:

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

where

$w_q$  = exponential averaging weight or weight factor

$q_k$  = actual queue size in round  $k$

$\bar{q}_k$  = average queue size in round  $k$ .

Drop probability in round  $k+1$ ,  $p_{k+1}$  is determined by the average queue size in round  $k+1$ ,  $\bar{q}_{k+1}$ , by comparing it to the minimum queue threshold,  $q_{min}$  and the maximum queue threshold,  $q_{max}$ . The drop probability will in turns determine the next actual queue size. The function of the drop probability is:

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

where  $p_{max}$  is maximum packet drop probability [Floyd & Jacobson, 1993]. The function is depicted in Figure 3.

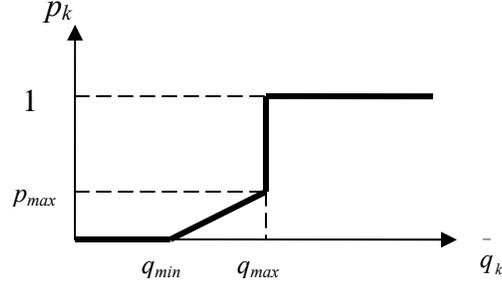


Fig 3: Drop probability as a function of average queue size.

### 3. TCP/RED Model

Liu et al. [Liu et al., 2005] have considered a simple network depicted in Figure 4 and developed a nonlinear first-order discrete-time TCP/RED model.

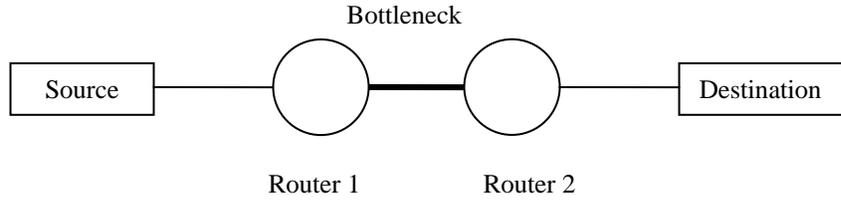


Fig 4: Topology of the modeled and simulated network.

The nonlinear first-order discrete-time dynamic model of TCP/RED is [Liu et al., 2005]:

$$\bar{q}_{k+1} = \begin{cases} (1 - w_q) \cdot \bar{q}_k + w_q \cdot \max\left(\frac{N \cdot K}{\sqrt{p_k}} - \frac{C \cdot d}{M}, 0\right) & \text{if } p_k \neq 0 \\ (1 - w_q) \cdot \bar{q}_k + w_q \cdot \max\left(rwnd \cdot N - \frac{C \cdot d}{M}, 0\right) & \text{if } p_k = 0. \end{cases} \quad (3)$$

The variables are:

$\bar{q}_{k+1}$  : average queue size in round  $k + 1$

$w_q$  : weight factor or exponential averaging weight

$N$  : 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.

The proposed model takes into account the slow start and timeout events, and the average queue size captures the queue dynamics in the RED gateway and reflects the dynamics of

the TCP congestion control mechanism. With several assumptions made, the model has been validated by varying RED parameters and comparing with ns-2 simulations [Zhang et al., 2005]. It has been shown by Floyd and Jacobson [Floyd & Jacobson, 1993] that the model follows the ns-2 simulations results closer than other models.

#### 4. Bifurcation Analysis

We try to investigate bifurcations and chaos in the TCP/RED model using the nonlinear analysis in [Ranjan et al., 2004b]. By referring to the drop probability function in (2) and substituting the drop probability with average queue size at previous RTT period,  $\bar{q}_k$ , (3) can be rewritten as

$$\bar{q}_{k+1} = g(\bar{q}_k, \rho) \quad (4)$$

where  $\rho$  summarizes all parameters except for  $\bar{q}_k$ .

The average queue size at period  $k$  is mapped to the average queue size at period  $k + 1$ . The fixed point of the map is an average queue size  $q^*$  such that  $q^* = g(q^*, \rho)$ . If the RED parameters are properly configured, the fixed point should fall between  $q_{min}$  and  $q_{max}$ . Therefore, the only area of interest is the equation where the average queue size is greater than  $q_{min}$ . Deriving from (4), the fixed point of the map is

$$q^* = \frac{N.K}{\sqrt{p^*}} - \frac{C.d}{M}. \quad (5)$$

The corresponding  $p^*$  is given as the square of the positive real solution of the polynomial

$$\frac{C.M}{v} y^3 + (CMq_{min} + dC^2)y - NMKC = 0 \quad (6)$$

where  $v = p_{max}/(q_{max} - q_{min})$ .

Local stability in the neighborhood of the fixed point is determined by the associated eigenvalue of the map

$$\begin{aligned} & \left. \frac{\partial}{\partial \bar{q}_k} g(\bar{q}_k, \rho) \right|_q \\ &= 1 - w_q - \frac{w_q \cdot N.K}{2 \cdot \sqrt{v} \cdot (q^* - q_{min})^{\frac{3}{2}}} \\ &= \lambda(q^*(\rho), \rho). \end{aligned} \quad (7)$$

The linear stability condition is

$$|\lambda(q^*(\rho), \rho)| = \left| 1 - w_q - \frac{w_q \cdot N \cdot K}{2 \cdot \sqrt[3]{v} \cdot (q^* - q_{\min})^2} \right| < 1. \quad (8)$$

When the eigenvalue becomes -1, it gives a period-doubling bifurcation (PDB) leading to oscillatory behaviour in the system [Alligood et al., 1996]. The critical value of  $w_q$  for which the eigenvalue becomes -1 is the bifurcation point of the system and is expressed as

$$w_{crit} = \frac{2}{1 + \frac{NK}{2(q^* - q_{\min})^2} \sqrt[3]{\frac{q_{\max} - q_{\min}}{P_{\max}}}}. \quad (9)$$

## 5. Bifurcation Analysis Simulation Results

### 5.1. Queue weight factor, $w_q$ as a bifurcation parameter

We investigate the bifurcation in the TCP/RED model using the following system parameters and RED parameters:

Table 1: System parameters.

Parameter	Value
Number of TCP connection, $N$	1
Link capacity, $C$ (bit/s)	1.54e+6
Packet size, $M$ (bits)	4,000
Round trip propagation, $d$ (s)	0.0228
Constant, $K$	$\sqrt{3/2}$
Receiver's advertised window size, $rwnd$	1,000

Table 2: RED parameters.

Parameter	Value
Maximum drop probability, $P_{max}$	0.1
Minimum queue threshold, $q_{min}$	5
Maximum queue threshold, $q_{max}$	15

Numerically, using (4) and parameters from Table 1 and Table 2, the fixed point of the system is approximately 5.712 packets. The bifurcation point is determined by using (9) and is found to be approximately 0.1794. Eigenvalues have been calculated using (8) to verify the PDB, they are shown in Table 3 and Figure 5.

Table 3: Eigenvalues computed for various values of  $w_q$ .

$w_q$	$\lambda$	Legend
0.1792	-0.9973	Close to PDB
0.1793	-0.9984	Close to PDB
0.1794	-0.9996	Close to PDB
0.1795	-1.0007	After PDB

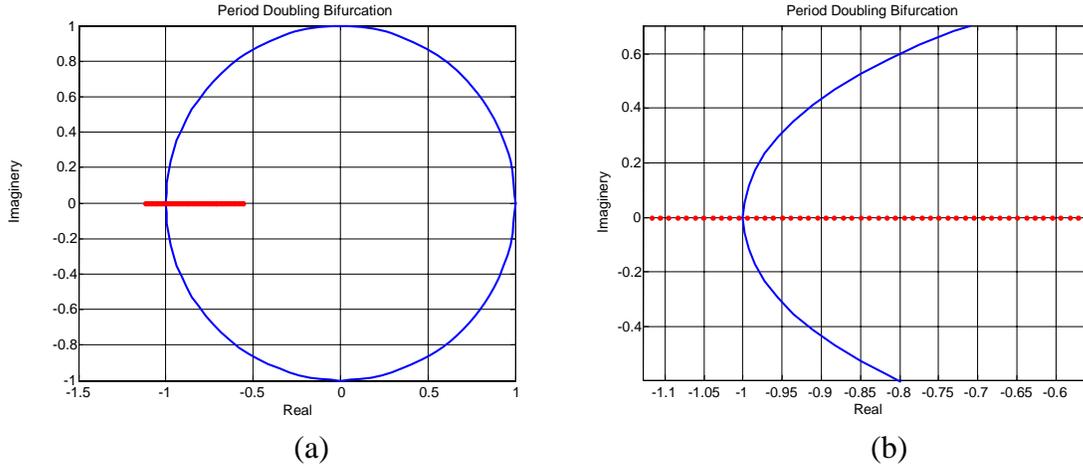


Fig 5: (a) Graphical presentation of real eigenvalues for  $w_q = 0.14$  to  $w_q = 0.19$ , with step 0.001, crossing the boundary of stability at -1 of the unit circle from right to left. (b) Zoom-in of (a).

The calculations of the non-linear analysis are further verified by the simulation results. We use  $w_q$  as the bifurcation parameter and vary it from 0.01 to 0.27, with step 0.001. The simulated results in Figure 6 show that the system transits from stable fixed point to chaos via a period doubling route. The inception point of the period doubling route is the bifurcation point at approximately 0.18. Oscillation between 2 points (period-2 orbit) is observed up to  $w_q = 0.221$  approximately. It then exhibits period doubling bifurcations and chaos. A small period-3 periodic window is embedded in the chaos region, when  $w_q \in [0.244, 0.246]$ .

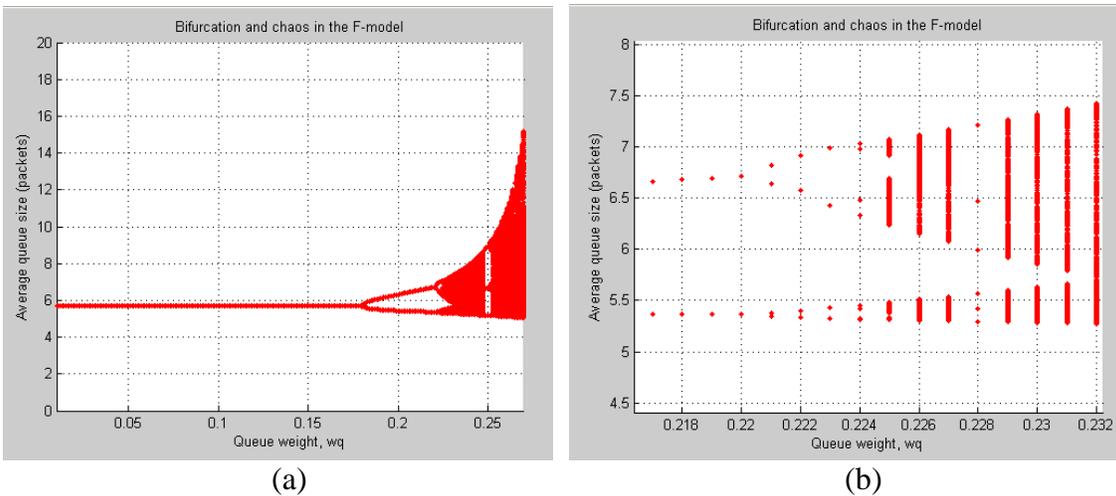


Fig 6: (a) Bifurcation diagram with queue weight,  $w_q$  as a parameter. (b) Zoom-in of the bifurcation diagram.

### 5.2. Minimum average queue threshold, $q_{min}$ as a bifurcation parameter

Minimum average queue threshold  $q_{min}$  is varied from 0 to 20, with step 0.001, for various values of  $w_q$ . As recommended by Floyd [Floyd, 1997],  $q_{max}$  is set to three times of  $q_{min}$ .

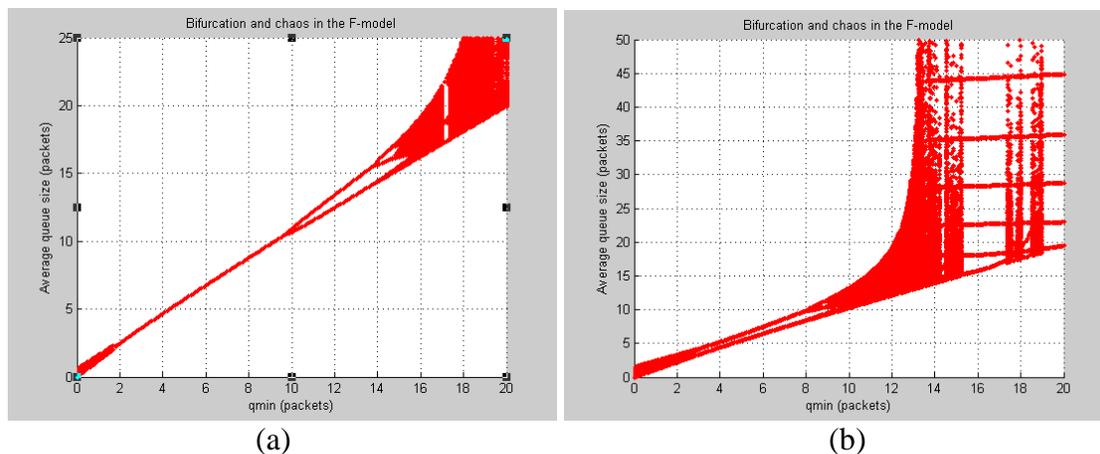


Fig 7: Bifurcation diagrams with  $q_{min}$  as a parameter, with various values of  $w_q$ . (a)  $w_q = 0.15$ . (b)  $w_q = 0.20$ .

When  $w_q$  is not fine tuned, Figure 7 (a) and (b) show that the system bifurcates and enters chaos via a period doubling route, when  $q_{min}$  is set to be too small or too large. For  $w_q = 0.15$ , Figure 7 (a) shows that a small period-3 periodic window is observed in the chaos region when  $q_{min} \in [17, 17.3]$ . For Figure 7(b), a period-3 periodic window is observed at  $q_{min} = 11.3$ . A period-13 periodic window is observed when  $q_{min} \in [15.3, 17.3]$ . When  $q_{min} \in [19.0, 20.0]$ , a period-20 periodic window is found in the midst of chaos. There are other various periodic windows in the chaos region that are too small to be observed.

The bifurcation and chaos reveal the limitation in setting RED parameters at the router. The average queue threshold is not allowed to be set above a certain value and that may result in desired throughput of the system to be compromised.

### 5.3. Maximum drop probability, $p_{max}$ as a bifurcation parameter

Maximum drop probability  $p_{max}$  is varied from 0.1 to 0.9, with step 0.001, for various values of  $w_q$ . For both cases, the instability is observed mainly in the region where values of  $p_{max}$  are high. This is expected as it is indicated by (2) that  $p_{max}$  is proportional to the actual drop rate, and high drop rate will cause the system to become unstable.

However, according to Floyd [Floyd, 1997], it is not necessary to set or to optimize  $p_{max}$  to anything higher than 0.1. In actual network, if a router is operating with steady-state

packet drop rates of 20-30%, it is assumed that something is wrong in the engineering of the network and it is not a region that we would like to optimize.

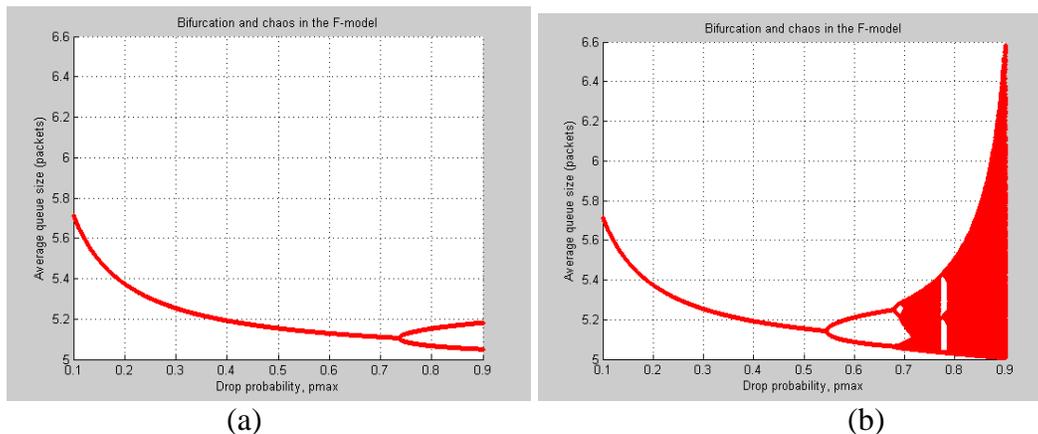


Fig 8: Bifurcation diagrams with  $p_{max}$  as a parameter, with various values of  $w_q$ . (a)  $w_q = 0.03$ . (b)  $w_q = 0.04$ .

## 6. Modified RED algorithm

As RED parameters vary, the average queue size drifts from the fixed point, bifurcation and subsequently chaos occur. Therefore, if an average queue size can be set at the fixed point, i.e.,  $\bar{q}_{k+1} = q^*$ , regardless of how the RED parameters change, it is possible to avoid bifurcation and chaos phenomena in the averaged queue length.

Here, we will illustrate how it is possible to achieve that. First and foremost, the prior knowledge fixed point,  $q^*$ 's corresponding  $p^*$  is required. It can be calculated using (6) at the routers of the system.

Initially, we assume the averaged queue size is in stable period-1 steady state. There are different routes to chaos in TCP dynamics [Liu et al., 2005; Chen et al., 2004; Wang, 2002; Gao et al., 2005], period-doubling route to chaos remains a very common one. In this particular case, the period-2 state is the first indication of oscillation. Once we detect this periodic-2 pattern for a sufficiently large time window, we will invoke the modified RED algorithm.

In simulation environment, the system parameters and queue weight,  $w_q$  of (3) are constant. Therefore  $\bar{q}_{k+1}$  is only dependent on the drop probability,  $p_k$ . It is apparent in (3) that if  $p_k = p^*$ ,  $\bar{q}_{k+1} = q^*$ . Therefore, our approach is to set  $p$  constant in the bifurcating and chaotic regions, and to keep the average queue size at the fixed point, to avoid bifurcation and chaos.

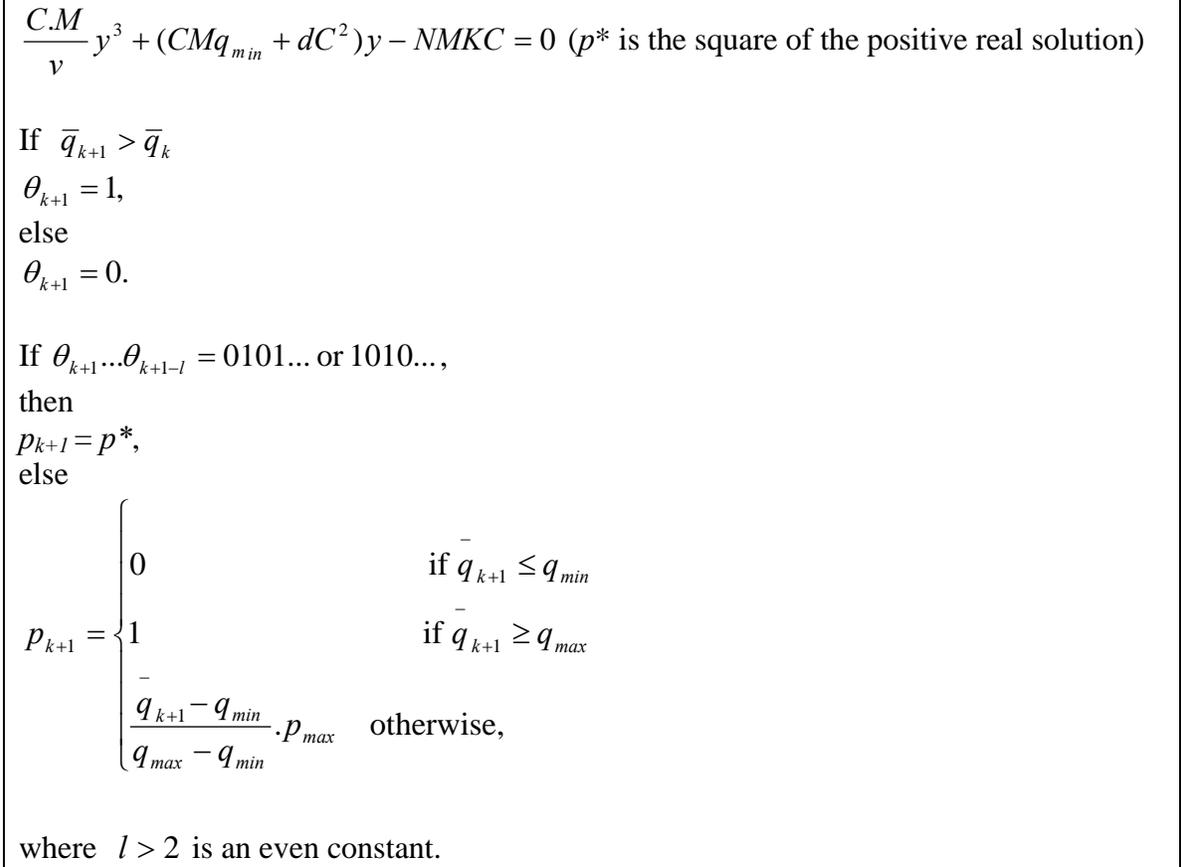


Fig 9: The modified RED algorithm.

The algorithm keeps track of the trend in average queue size. If oscillation occurs, the  $p$  is fixed to  $p^*$ , else  $p$  is calculated using (2) at the router. The idea of detecting oscillation is proposed by Wang [Wang, 2002; Chen et al., 2004].

## 7. Modified RED Algorithm Simulation results

### 7.1. Queue weight factor, $w_q$ as a bifurcation parameter

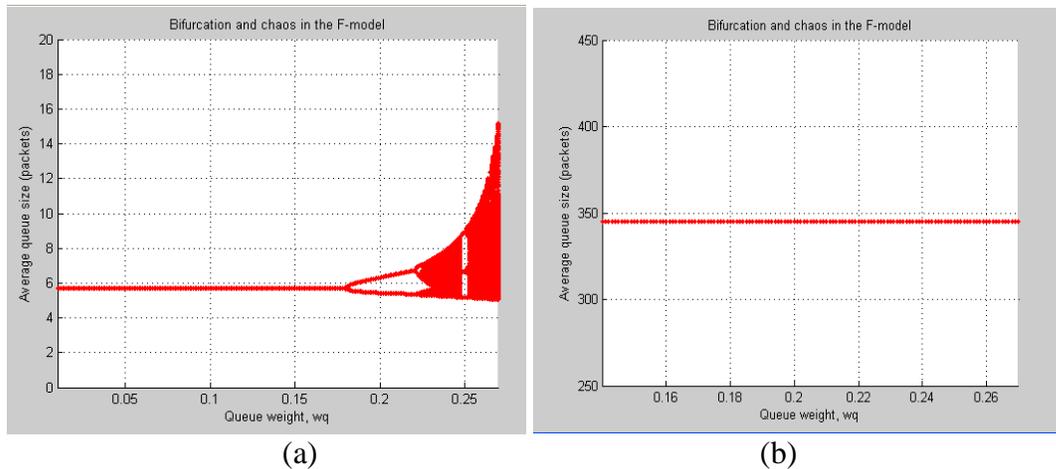


Fig 10: Bifurcation diagram of average queue size with respect to queue weight. (a) Original RED algorithm. (b) Modified RED algorithm.

**7.2. Minimum average queue threshold,  $q_{min}$  as a bifurcation parameter**

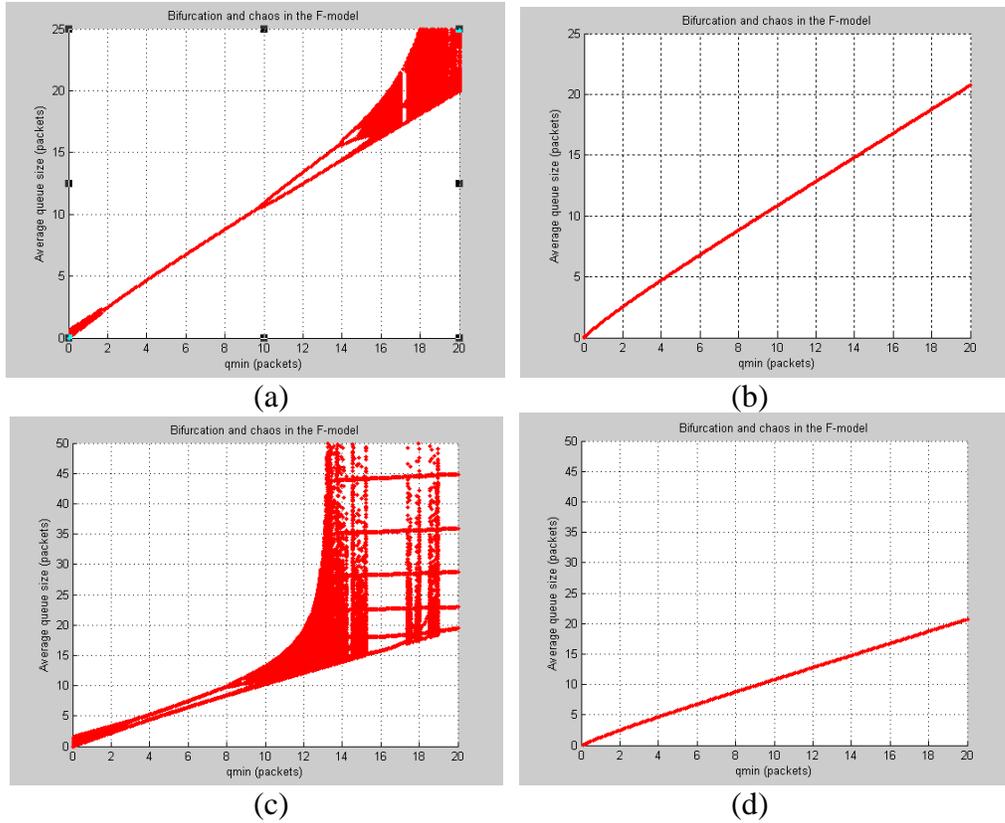
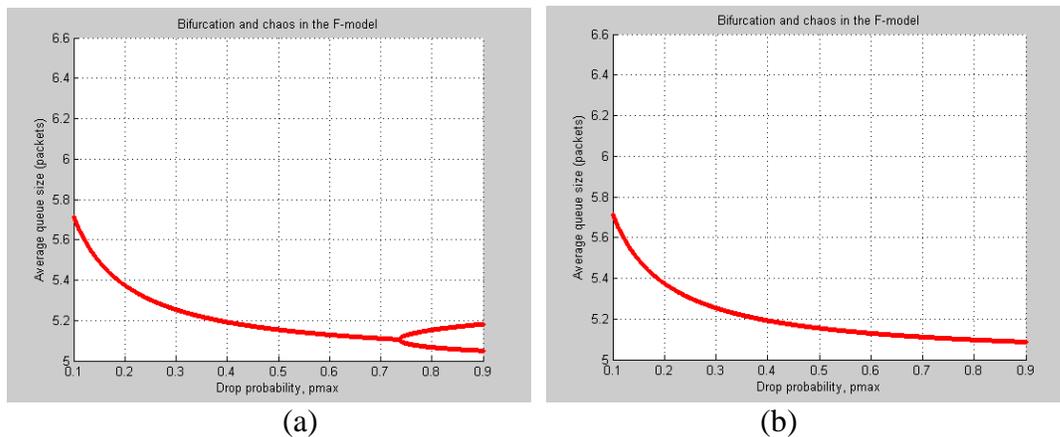


Fig 11: Bifurcation diagram of average queue size with respect to minimum average queue threshold. (a) Original RED algorithm ( $w_q = 0.03$ ). (b) Modified RED algorithm ( $w_q = 0.03$ ). (c) Original RED algorithm ( $w_q = 0.04$ ). (d) Modified RED algorithm ( $w_q = 0.04$ ).

**7.3. Maximum drop probability,  $p_{max}$  as a bifurcation parameter**



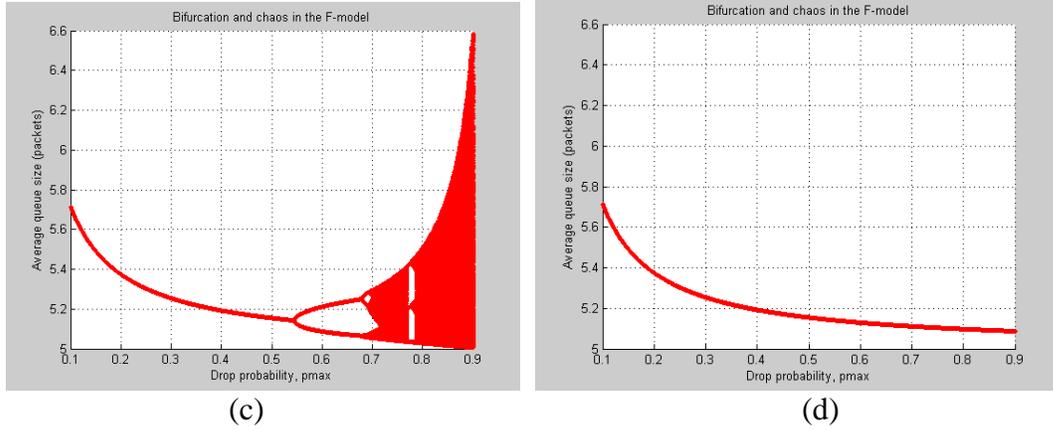


Fig 12: Bifurcation diagram of average queue size with respect to drop probability. (a) Original RED algorithm ( $w_q = 0.15$ ). (b) Modified RED algorithm ( $w_q = 0.15$ ). (c) Original RED algorithm ( $w_q = 0.2$ ). (d) Modified RED algorithm ( $w_q = 0.2$ ).

#### 7.4. Variation in system and RED parameters

System and RED parameters are selected according to parameters in [Ranjan et al., 2004b]:

Table 4: System parameters.

Parameter	Value
Number of TCP connection, $N$	250
Link capacity, $C$ (bit/s)	75e+6
Packet size, $M$ (bits)	4,000
Round trip propagation, $d$ (s)	0.1
Constant, $K$	$\sqrt{3/2}$
Receiver's advertised window size, $rwnd$	1,000

Table 5: RED parameters.

Parameter	Value
Maximum drop probability, $P_{max}$	0.1
Minimum queue threshold, $q_{min}$	250
Maximum queue threshold, $q_{max}$	750

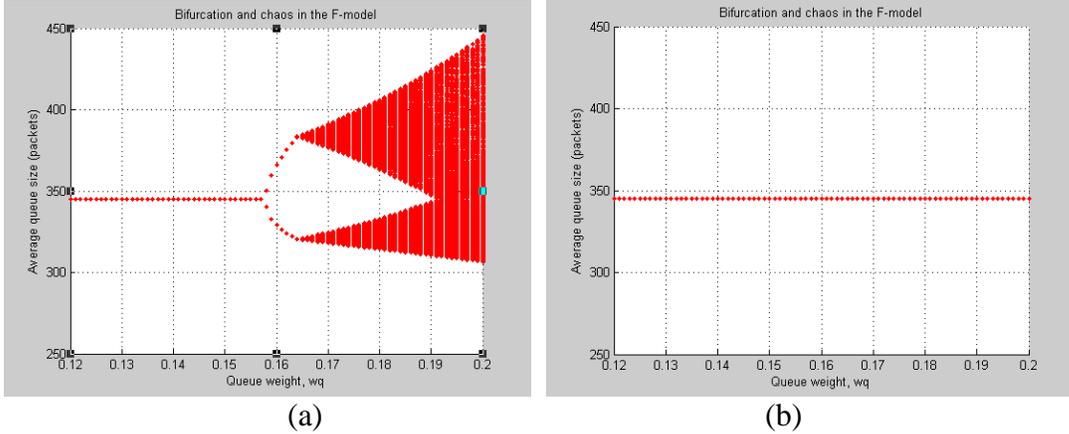


Fig 13: Bifurcation diagram of average queue size with respect to the averaging weight. (a) Original RED algorithm. (b) Modified RED algorithm.

The results from Figures 10-13 indicate that the modified algorithm successfully prevents bifurcation and chaos regardless of the system and RED parameters chosen. It thus removes the limitation in setting the parameters and restriction on desired throughput of the system.

The algorithm does not change the dynamic in the transient region (when a connection is first detected). The drop probability is only fixed to  $p^*$  when the router detects oscillation in the average queue in the “supposed” steady state. It does not therefore suppress the throughput of the system. The algorithm has no negative impact on the average queuing delay either.

## 8. Limitations

There are some limitations of our model:

- Even if the average queue length  $\bar{q}_k$  is non-chaotic, there are no guarantees that the actual queue length is not chaotic.
- TCP has many parameters such as congestion window size and average packet delay. Even if the average queue length  $\bar{q}_k$  is non-chaotic. The other parameters could be. For example, if RED employs a fixed  $p^*$ , the congestion window  $cwnd$  could be chaotic for certain values. As shown in [Rao et al., 2005], the fixed random drop rate of packets could result in chaotic  $cwnd$  trajectories. Even under the condition that the proposed RED modification stabilizes the average queue length  $\bar{q}_k$ , TCP dynamics can be chaotic in terms of other parameters.
- TCP measurements collected over Internet connections could be much more complicated than traces considered in this paper because typically there are multiple routers on TCP paths and there is significant amount of non-TCP traffic.

In fact, the measurements typically have both chaotic and stochastic components [Gao et al., 2005] even without random drops of RED routers.

## 9. Conclusion

Performance of current version of RED is restricted by the traffic and its parameter configuration. Bifurcation and chaos phenomena in the averaged queue length are observed when parameters are not set appropriately. It thus poses limitations in selecting parameters, as well as compromises throughput for stability.

Our modified RED algorithm succeeds in preventing bifurcation and chaos in the averaged queue length, regardless of parameters chosen. It removes the limitation of configuring parameters. The algorithm does not suppress the throughput and has no impact on average queue delay either.

In the original map of the system, to update the average queue size, a router requires knowledge of the system parameters such as number of connections, capacitor link, round trip propagation delay and packet size. Therefore, it should not be a concern that the router is not capable of computing the corresponding drop probability of the fixed point,  $p^*$  and the fixed point,  $q^*$ , which rely on both the system and RED parameters.

For the future work, the system parameters are not constant and this may affect the effectiveness of the algorithm in a real network. The algorithm should be further tested and clarified in various environments to investigate its feasibility to be deployed in the real network.

## References

- Alligood, K.T., Sauer, T.D. & Yorke, J.A. [1996] *Chaos: An Introduction to Dynamical Systems*. New York: Springer-Verlag.
- Allman, M., Paxson, V. & Steven, W. [1999] "TCP congestion control," *Request for Comment (RFC) 2581*.
- Barden, B. et al. [1998] "Recommendations on queue management and congestion avoidance in the internet," *Request for comments (RFC) 2309*.
- Chen, L., Wang, X.F. & Han, Z.Z. [2004] "Controlling bifurcation and chaos in internet congestion control model," *International Journal of Bifurcation and Chaos*, vol. 14, no. 5, pp.1863-1876.
- Firoiu, V. & Borden, M. [2000] "A study of active queue management for congestion control," *Proc. of IEEE INFOCOM'00*, Tel Aviv, Israel, pp. 1435-1444.
- Floyd, S. [1997] "RED: Discussion of setting parameters," <http://www.icir.org/floyd/REDparameters.txt>.
- Floyd, S. & Jacobson, V. [1993] "Random early detection gateways for congestion avoidance," *IEEE/ACM Trans. Networking*, vol. 1, no. 4, pp.397-413.
- Gao, J. & Rao N.S.V. [2005] "TCP AIMD dynamics over Internet connections," *IEEE Communications Letters*, vol. 9, pp. 4-6.

- Jacobson, V. [1988] "Congestion avoidance and control," *ACM Computer Communication Review*, vol. 18, no. 4, pp.314-329.
- Jacobson, V. [1990] "Modified TCP congestion avoidance algorithm," <ftp://ftp.isi.edu/end2end/end2end-interest-1990.mail>.
- La, R.J. [2004] "Instability of a tandem network and its propagation under RED," *IEEE Trans. on Automatic Control*, vol. 49, no. 6, pp1006-1011.
- Liu, M., Zhang, H. & Trajkovic, L. [2005] "Stroboscopic model and bifurcations in TCP/RED," *Proc. IEEE Int. Symp. Circuits and Systems*, Kobe, Japan, pp. 2060-2063.
- Ranjan, P., La, R.J. & Abed, E.H. [2004a] "Washout filter-aided RED control," *Proc. American Control Conference*, Boston, USA, pp. 2966-2971.
- Ranjan, P., Abed, E.H. & La, R.J. [2004b] "Nonlinear instabilities in TCP-RED," *IEEE/ACM Trans. on Networking*, vol. 12, no. 6, pp. 1079-1092.
- Rao, N.S.V., Gao, J. & Chua, L.O. [2005] "On dynamics of transport protocols in wide-area Internet connections," in *Complex Dynamics in Communication Networks*, Springer Verlag Series: Understanding Complex Systems, pp. 69-102.
- Wang, L., Li, Z.Z. & Chen, Y.P. [2005] "A control theoretic analysis of mixed TCP and UDP traffic under RED based on nonlinear dynamic model," *Proc. Third International Conference on Information Technology and Applications*, Sydney, Australia, pp. 747-750.
- Wang, X.F. [2002] "Controlling bifurcation and chaos in Internet congestion control system," *Proc. Forth World Congress on Intelligent Control and Automation*, Shanghai, P.R. China, pp. 573-576.
- Wolfram Research [1999] "Bifurcation," <http://www.mathworld.wolfram.com/Bifurcation.html>.
- Zhang, H., Liu, M., Vukadinovic, V. & Trajkovic, L. [2005] "Modeling TCP/RED: a dynamical approach," in *Complex Dynamics in Communication Networks*, Springer Verlag Series: Understanding Complex Systems, pp. 251-278.