TECHNICAL RESEARCH REPORT

Nonlinear Instabilities in TCP-RED

by Priya Ranjan and Eyad H. Abed

TR 2001-37

Nonlinear Instabilities in TCP-RED

Priya Ranjan and Eyad H. Abed

Abstract—**This work introduces a discrete time model for a simplified TCP network with RED control. It is argued that by sampling the state space at certain instants, the dynamics of the system can be described explicitly as a discrete time feedback control system. This system is used to analyze the operating point of TCP-RED and its stability with respect to various controller and system parameters. With the help of bifurcation diagrams, it is numerically shown that non-trivial (not due to the discontinuity in the system or the control law) instabilities in the system are possible due to the presence of a strong nonlinearity in the characteristics of TCP throughput of a sender as a function of drop probability at the gateway. Some of the bifurcations observed in the system are the period-doubling sequence and border collisions leading to a change in the system periodicity and chaos. Analytical techniques are provided to help in the understanding of this kind of anomalous behavior. An explicit stability condition in terms of different parameters is given.**

Keywords—**Congestion, computer networks, chaos, bifurcation, control**

I. Introduction

Computer networks are highly complicated systems, both in their temporal and spatial behavior [1]. Although they have traditionally been modeled and analyzed using stochastic methods, there have recently been several papers that use deterministic nonlinear modeling and analysis (e.g., [6], [7], [8], [16], [14], [5]).

In this paper, we study a modified deterministic dynamical model of a simple computer network running TCP at the sender end and implementing RED at the router end. The basic model that we consider was proposed recently by Firoiu and Borden [5]. We modify their model with a simpler TCP throughput function [3], [4] to facilitate analysis. The calculations we give here show that the model exhibits a rich variety of bifurcation behavior leading to chaotic behavior of the computer network. The bifurcations occur as control parameters are slowly varied, moving the dynamics from a stable fixed point to oscillatory behavior and finally to a chaotic state.

A glimpse at the history of network congestion control reveals significant attempts to control congestion in the general network and telephony literature. Congestion and synchronization in tandem telephone queues have been studied in [15] using a piecewise affine model. A similar model has been applied to the dynamics of choke packets in a LAN to explain synchronization and sustained oscillations [16]. These models indeed explain qualitative changes in the operation of a network or that of a network component as parameters cross critical values. In contrast to the deterministic setting of [15] and [16], multi-stability or emergence of pseudo-stable states has been reported in a stochastic setting in [14]. The paper discusses the qualitative changes in the stochastic behavior of the network due to parameter change, which may lead to degradation in network performance.

There have been several attempts to deal with congestion in TCP which is the most popular network mechanism for data transfer. The most important scheme to avoid impending congestion was published in [2] and is known as random early drop, or RED. The basic idea of RED is to sense impending congestion before it happens and try to give feedback to the sender by dropping its packets. The dropping probability is the control administered by the gateways once they detect queue build-up beyond a certain threshold. This scheme involves three parameters: 1) p_{max}, 2) q_{max}, and 3) q_{min} that need to be selected. (The meanings of these parameters will be identified in the next section.) Most of the rules for setting these parameters are empirical, and come from networking experience. These rules have been evolving as the effects of controller parameters is not very clearly understood. There are papers discouraging implementation of RED (e.g., [9]), arguing that there is insufficient consensus on how to select controller parameter values, and that RED does not provide a drastic improvement in performance.

Initially, there was very little in the way of mathematical modeling of TCP-RED. However, with the recent efforts toward modeling TCP throughput for a transmission line with a packet drop probability [5], [6], [7], [8], [4], several papers have discussed TCP-RED in the framework of feedback control systems. Most of the models used are continuous-time and the analysis uses basic control theoretic results. The biggest problem with the continuous-time models is their inability to reflect delay, which is prominent in networks and can be very significant for large trunks [7], [8]. Continuous-time models with variable (state dependent) delay are hard to analyze [8]. The analysis reported on these models deals mostly with the stability

The authors are with the Department of Electrical and Computer Engineering and the Institute for Systems Research, University of Maryland, College Park, MD 20742 USA. Email: abed@isr.umd.edu.

of fixed points and limit cycles under different parameter settings. For the first time, chaotic behavior of TCP has been reported in [13]. The evidence for this irregularity is mostly explored by simulations. Some theoretical work on flow synchronization in TCP has been reported in [4], [17], but one of the very important issues which currently isn't well understood is how does a smoothly operating network transition into chaos. To borrow dynamical systems terminology, the route to chaos starting from a stable fixed point is not well-studied.

In this work, a discrete-time map will be used to model the TCP-RED interaction. A dynamical systems approach will be used to explain the loss of stability, bifurcation behavior, and routes to chaos in TCP-RED networks. We will use bifurcation-theoretic ideas to explain nontrivial periodic behavior of the system. The appearance of bifurcation and chaos should not be surprising, considering that the system response is nonlinear especially during heavy load conditions. We will show the performance of the system as a function of various control and system parameters in general and try to explain these irregular behaviors with the help of bifurcation diagrams.

Our work begins by realizing that the model proposed in [5] can be viewed as a first-order (rather than third-order) discrete nonlinear model. Our replacement of the TCP throughput function of [5] with a simpler version makes the analysis feasible. However, symbolic calculations could be used to allow treatment of the more complex throughput function of [5]. The advantage of the current work is that the calculations are simple enough that the results are easily understood.

We borrow the model proposed by Firiou et al. [5] and use the well known formula for TCP throughput proposed by many others including [3], [4]. The motivation behind not using Firiou's forumla for TCP throughput is its complexity. Complex operations like inverse of a function in different parameters, which are needed to connect the TCP to the control mechanism RED, demand simplicity in the TCP throughput formulation. This seems to be the reason why Firiou et al. postponed the study of their proposed map [5]. Although this TCP-RED formulation may not be the exact representation of the complicated mechanism, it does give a qualitative handle on its dynamics and enhances our understanding of chaos and other instabilities. We hope that this understanding will lead to monitor the network congestion better and help us in formulating robust but simplified control mechanisms.

This paper is organized as follows. In Section 2, we describe the TCP-RED mechanism in control system framework. Section 3 contains the discrete map of TCP-RED mechanism. Section 4 deals with the stability of this map which is the core of the paper. Section 5 tries to explain the different nonlinear phenomena we have observed in our models and try to make a connection with chaotic scenario. Finally, in section 6 we discuss the results in networking context.

II. TCP-RED: Feedback System Modeling

A computer network implementing TCP-RED is essentially a feedback loop where senders adjust their sending rate based on the feedback they receive from their nearest routers in the form of dropped packets. Routers on the other hand implement a control policy which can be either drop tail or RED [2]. There have been different approaches to model the dynamics of TCP-RED and various control schemes have been proposed [5], [6], [7], [8], [4] not only to control the system but to also enhance its dynamic performance. We closely follow the approach taken in [5] with a modified TCP throughput formula.

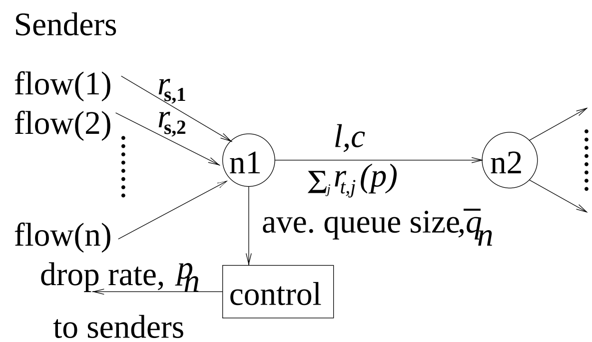

Fig. 1. Simplified Network Diagram

Each flow at a router sends packets with rate $r_{s,i}$. The sending rates of all n flows combine at the buffer of link l and generate a queue of size q which is limited by its buffer size B. The controller at the router drops packets with probability p which is a function of average queue size $\overline{q}$. For ith flow let the forwarding rate at the router be $r_{t,i}$ which is the same as $r_{s,i}$ sans dropped packets. When a sender notices that its packets are being dropped, it adjusts its sending rate based on the drop probability p it observes.

This makes a control system with sender's rate as control variable with the controller sitting at the router which issues the feedback signal in the form of a drop probability. The aim of this control system is to keep the cumulative throughput below or equal to the link's capacity c:

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

Let's assume for simplicity that TCP flows have a long duration and that their number n does not change, then the throughput of each TCP flow follows the steady state model derived in [3], [4].

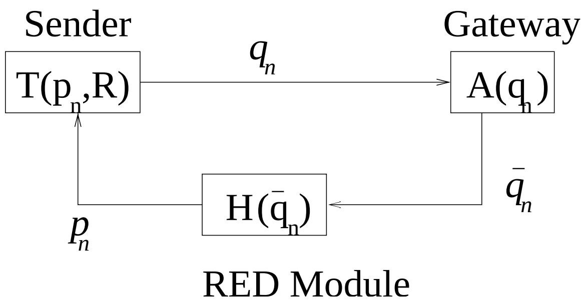

Fig. 2. Feedback control system corresponding to the network shown

$$r_{t,i} = T(p, , R_i)$$

$$T(p, R) = \frac{M}{R}\frac{k}{\sqrt{p}} \tag{1}$$

where

$$\begin{aligned} T &= \text{Throughput of a TCP flow (in bits/sec)} \\ M &= \text{Maximum Segment Size or Packet Size} \\ R &= \text{Round Trip Time} \\ k &= \text{constant which varies between 1 and } \sqrt{8/3}\ [3] \\ p &= \text{probability of packet loss} \end{aligned}$$

To simplify matters even further let's assume that all flows are uniform or they all have same round trip time R, same maximum segment or packet size M and maximum congestion window size advertised by TCP's receiver W_{max} is large enough to not affect $T(p, R)$. This implies

$$\begin{aligned} r_{t,i}(p, R) &= r_{t,j}(p, R), \quad 1 \leq i, j \leq n \text{ and hence} \\ &\leq \frac{c}{n} \end{aligned}$$

So this assumption enables us to reduce the n-flow system to a single flow system with feedback although it is important to keep in mind that feedback is based on the sending rate of all the flows since the router has no way to differentiate between them, at least in this set up.

To define this control system mathematically, we model the queue as a function of control variable $\overline{q} = G(p)$, which acts like a plant in control system literature. To analyze this control system we also need the control function $p = H(\overline{q})$ implemented at the gateways. This control function H is given by the policy implemented at the drop module, such as Drop-Tail or RED [2]. Now following the procedure suggested in [5] we can define the plant function $G(p)$ as follows:

$$G(p) = \begin{cases} \max(B, \frac{c}{M}(T_R^{-1}(p, \frac{c}{n}) - R_0)) & : \ p \leq p_0 \\ 0 & : \ otherwise \end{cases} \tag{2}$$

Where

$$\begin{aligned} R_o &= \text{propagation and transmission time and} \\ p_o &= T_p^{-1}(c/n, R_0) \end{aligned}$$

$T_R^{-1}(c/n, R_0)$ denotes the inverse of $T(p, R)$ in R, $T_p^{-1}(c/n, R_0)$ denotes the inverse of $T(p, R)$ in p and p_0 is maximum probability for which the system is fully utilized i.e. for $p \geq p_0$ senders will have their rates too small to keep the link fully utilized. For $T(p, R)$ defined by eq. 1.

$$p_0 = \left(\frac{Mk}{R_0\frac{c}{n}}\right)^2 \tag{3}$$

$$G(p) = \begin{cases} \max(B, \frac{c}{M}(\frac{Mk}{\frac{c}{n}\sqrt{p}} - R_0) & , \ \text{if } p \leq p_0 \\ 0 & , \ \text{otherwise} \end{cases} \tag{4}$$

RED control law can be expressed as follows:

$$\begin{aligned} p &= H(\overline{q_e}) \\ &= \begin{cases} 0 & , \ 0 \leq \overline{q_e} < q_{min} \\ \frac{\overline{q_e} - q_{min}}{q_{max} - q_{min}} p_{max} & , \ q_{min} \leq \overline{q_e} < q_{max} \\ 1 & , \ q_{max} \leq \overline{q_e} \leq B \end{cases} \end{aligned} \tag{5}$$

where $\overline{q_e}$ is the exponential weighted moving average of queue size, q_{min}, q_{max}, p_{max} are configurable RED parameters, and B is buffer size.

III. Discrete model for TCP-RED

It is argued in [5] that TCP adjusts its sending rate depending on whether it has sensed that packets are discarded. Hence, this process can be modeled as a stroboscopic map where the instant of observation is one RTT or return trip time. This technique has been utilized before for different clocked systems in power electronics for modeling the dynamics of power converters. Following similar arguments it seems reasonable to model TCP-RED dynamics as a discrete map. Although one would prefer that the sampling interval be regular, there are models where the dynamics is sampled at irregular intervals and the resulting maps are known as "impact maps" [11].

Let p_k be the drop probability at t_k. At time $t_{k+1} = t_k + RTT$ the sender observes drop rate p_k and in an average sense, adjust its sending rate. This in turn forces the buffer to its new state $q_{k+1} = G(p_k)$ following the queue law in eq. 4. The RED module now computes a new estimate of queue size $\overline{q}_{e,k+1} = A(\overline{q}_{e,k}, q_{k+1})$, following the exponential weighted moving average:

$$A(\overline{q}_{e,k}, q_{k+1}) = (1-w)\overline{q}_{e,k} + wq_{k+1} \tag{6}$$

where w is the weight used for averaging. After computing $\overline{q}_{e,k+1}$, the RED module adjusts it dropping rate to $p_{k+1} = H(\overline{q}_{e,k+1})$ given by its "feedback control law" in eq. 5. This completes the description of a discrete time dynamical system modeled as follows:

$$\begin{aligned} q_{k+1} &= G(p_k) \\ \overline{q}_{e,k+1} &= A(\overline{q}_{e,k}, q_{k+1}) \\ p_{k+1} &= H(\overline{q}_{e,k+1}) \end{aligned} \tag{7}$$

From eq. 7 one can derive a simple one dimensional recurrence equation since map $G(.)$ and map $H(.)$ has no dynamics. The only dynamics that comes into the picture is in map $A(.,.)$. After substitution one can easily derive the following equation for exponential weighted moving average for queue length at time t_{k+1}:

$$\overline{q}_{e,k+1} = (1-w)\overline{q}_{e,k} + wG(H(\overline{q}_{e,k})) \tag{8}$$

Below, we illustrate some interesting dynamical behavior of eq. 8. This equation is rather simple in most of its domain of definition.

We know that $G(.)$ is identically 0, $\forall\ p \geq p_0$. So we can find corresponding value b_1 of $\overline{q}_{e,k}$ such that for any $\overline{q}_{e,k} \geq b_1$, $G(.)$ is identically 0 if we assume a monotone feedback law.

$$b_1 = \begin{cases} \frac{p_0(q_{max}-q_{min})}{p_{max}} + q_{min} &, \text{ if } p_{max} \geq p_0 \\ q_{max} &, \text{ otherwise} \end{cases} \tag{9}$$

This gives an explicit formula for map in eq. 8 $\forall\, \overline{q}_{e,k} \geq b_1$:

$$\overline{q}_{e,k+1} = (1-w)\overline{q}_{e,k}$$

Now consider the other boundary value b_2 of $\overline{q}_{e,k}$ such that $\forall\ \overline{q}_{e,k} \leq b_2$ we have $G(.) = B$ or buffer is always full. This value can be computed from eq. 4 and eq. 5. b_2 is given by:

$$b_2 = \frac{\left(\frac{nk}{B+\frac{R_0 c}{M}}\right)^2}{p_{max}}(q_{max}-q_{min}) + q_{min} \tag{10}$$

This gives an explicit formula for map in eq. 8 $\forall\, \overline{q}_{e,k} \leq b_2$:

$$\overline{q}_{e,k+1} = (1-w)\overline{q}_{e,k} + wB$$

It is clear that most of the interesting dynamics happens for $b_2 \leq \overline{q}_{e,k} \leq b_1$. Map in eq. 8 can be written for this region as follows:

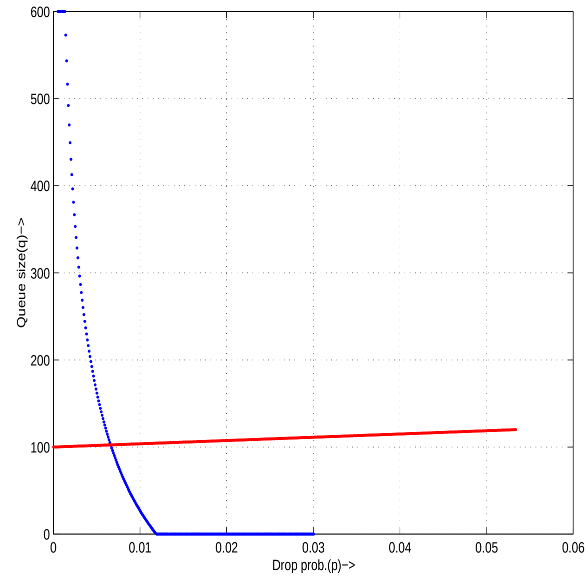

Fig. 3. Feedback law $H(\overline{q}_{e,k})$ in red and queue law G(p) in blue

$$\begin{aligned} \overline{q}_{e,k+1} &= (1-w)\overline{q}_{e,k} + w\left(\frac{nk}{\sqrt{\frac{p_{max}(\overline{q}_{e,k}-q_{min})}{(q_{max}-q_{min})}}} - \frac{R_0 c}{M}\right) \\ &:= f(\overline{q}_{e,k}, \rho) \end{aligned} \tag{11}$$

where ρ summarizes the parameters in the system.

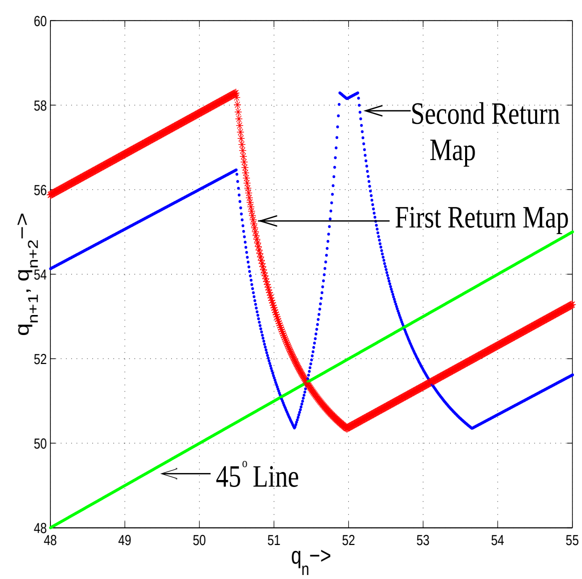

Fig. 4. 1. Red curve shows first return map, 2. blue curve shows second return map, and 3. green line is 45^o line whose intersection denotes the fixed points of the map

We remark that solving eq. 11 leads to a third degree polynomial in fixed point $\overline{q}_e^*$ which interestingly does not depend on w as should be expected since, both the "queue law" and the "feedback control law" are not functions of

w. The polynomial is given below.

$$(\overline{q}_e^* - q_{min})(\overline{q}_e^* + \frac{R_0 c}{M})^2 = \frac{(nk)^2}{p_{max}}(q_{max} - q_{min}) \quad (12)$$

IV. Stability

Stability of this fixed point $\overline{q}_e^*$ can be assessed by computing its eigenvalue:

$$\begin{aligned} \frac{df(\overline{q}_{e,k}, \rho)}{d\overline{q}_{e,k}} &= 1 - w - \frac{wnk}{2(\overline{q}_e^* - q_{min})^{\frac{3}{2}}}\sqrt{\frac{q_{max} - q_{min}}{p_{max}}} \\ &:= \lambda(\rho) \end{aligned} \quad (13)$$

Although $\lambda(\overline{q}_e^*, \rho)$ is a function of the fixed point $\overline{q}_e^*$ itself, we know that the fixed point will always be bounded from above by $f(b_2, \rho)$ and from below by $f(b_1, \rho)$. It is also clear that $f(b_2, \rho) > f(b_1, \rho)$ since control mechanism kicks in once the queue length at the router grows beyond b_2 decreasing the average queue length. In fact $f(\overline{q}_{e,k}, \rho)$ decreases monotonically in the interval b_2 to b_1, but the slope decreases in the magnitude. Hence an approximate stability condition for fixed point in terms of parameters can be derived by taking the upper bound of $f(\overline{q}_e^*, \rho)$ which is $f(b_2, \rho)$ and thats when $f(\overline{q}_{e,k}, \rho)$ has its eigenvalue negative and largest in magnitude. Hence, this stability condition can be formulated as:

$$|\lambda(\overline{q}_e^*, \rho)| < 1, \text{ or by substituting } b_2 \text{ by } \overline{q}_e^*$$
$$\left| 1 - w - \frac{wnk}{2(b_2 - q_{min})^{\frac{3}{2}}}\sqrt{\frac{q_{max} - q_{min}}{p_{max}}} \right| < 1 \quad (14)$$

where b_2 is given by eq. 10. Please note that stability condition given by eq. 14 involves the buffer size B despite of the fact that fixed point of the map does not depend on the the buffer size. The inclusion of buffer size makes the result conservative but it can be argued that a conservative design is good for the system's convergence since it has finite capacity and hence, even a marginally stable system may not be acceptable in practise.

V. Numerical Results

The behavior of this map can be explored numerically in parameter space to look for interesting dynamical phenomena. As eigenvalue moves towards unit circle, fixed point will become unstable and depending on the nature of bifurcation there can be new fixed points or chaos. There is also a possibility of fixed point colliding with either border b_1 or b_2 which has its own rich world of bifurcations.

A whole range of different dynamical scenario is presented here. Lets first consider the effect of varying q_{min} on fixed point of the map with different values of exponential averaging weight w. We analyze this effect with the help of numerical bifurcation diagrams.

A. How to Read a Bifurcation Diagram

A bifurcation diagram shows the qualitative changes in the nature or the number of solutions of a dynamical system as a parameter varies. On the horizontal axis we plot the different value of parameter (q_{min} or w in this case). The vertical axis shows the corresponding value of fixed points or periodic orbits, which is different queue buildups in this context. We have normalized the actual queue buildup by dividing it with q_{min} for the easy of visualization. Thats why the legend on the vertical axis reads *Norm. queueing at the router.* The way to read a bifurcation diagram is fix a point on the horizontal axis and draw a vertical line. The number of places the bifurcation curve intersect that vertical line is the number of equilibrium points of the system. If there is only one point then it is a stable fixed point for that parameter whereas the presence of more than one point indicates that system has a stable periodic orbit. Since we have only plotted stable solutions corresponding to the different value of parameters, the intersection of vertical line and the bifurcation curve only indicated the number of stable solutions. Disappearance of a branch implies that, solution corresponding to that branch became unstable and vice versa. All the bifurcation diagrams use three types of symbols. Red star, green triangle and blue dot denote the normalized borders b_2, b_1 and the system solution respectively.

B. Effect of Exponential Averaging Weight w

The following parameters are common to next three bifurcation plots [5].

$$\begin{aligned} q_{max} = 100, \quad q_{min} = 50, \quad c = 1500kbps, k = \sqrt{8/3} \\ B = 300 \text{ packets}, \quad R_0 = 0.1sec, \quad M = 0.5kb \\ n = 20, \quad w = \text{bifurcation parameter} \end{aligned}$$

The first three bifurcation plots(Figs. 5,6,7) show the effect of varying the exponential weight w with different values of p_{max}. For the small value of w these plots have a fixed point which looks like a straight line but after some critical value of w this straight line splits into two and this map exhibits period-doubling bifurcation. This is first indication of an oscillatory behavior appearing in the system due to its inherent nonlinearity as opposed to discontinuity in "queue or control law" which has been proposed earlier. This period two oscillation starts batching load at the router as shown in the plots. Increasing

w further shows that one of the branches collide, with the upper border of the map showing a chaos type phenomena. This is basically a bifurcation sequence expressed as $1 \to 2 \to chaos$. We suspect this to be a case of *border collision* bifurcation [12]. Border collision bifurcation is a well understood phenomenon in piecewise linear systems and has been shown responsible for chaos in different electrical circuits and economic system models. A technical proof for the border collision bifurcation will be reported somewhere else.

0. $p_{max} = 0.1$,

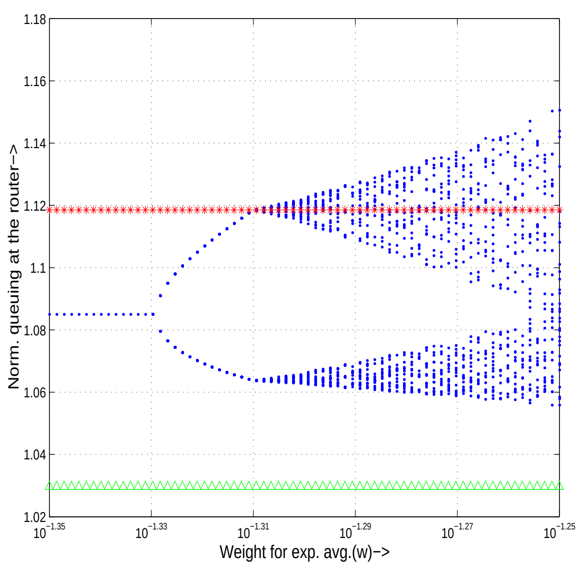

Fig. 5. Bifurcation diagram of average queue length w.r.t. w, $p_{max} = 0.1$

1. $p_{max} = 0.3$,

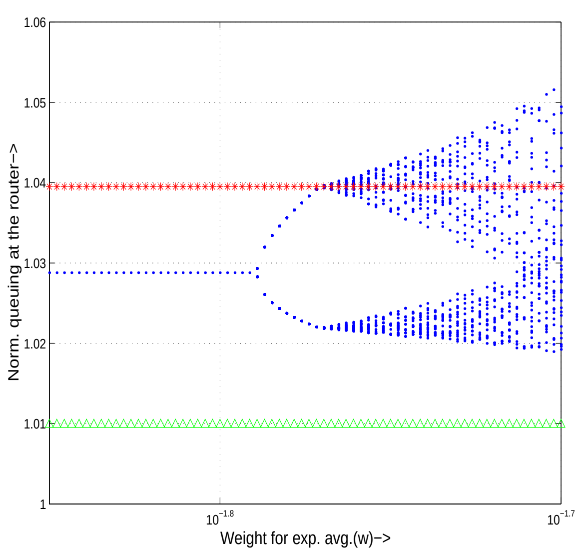

Fig. 6. Bifurcation diagram of average queue length w.r.t.w, $p_{max} = 0.3$

2. $p_{max} = 1$

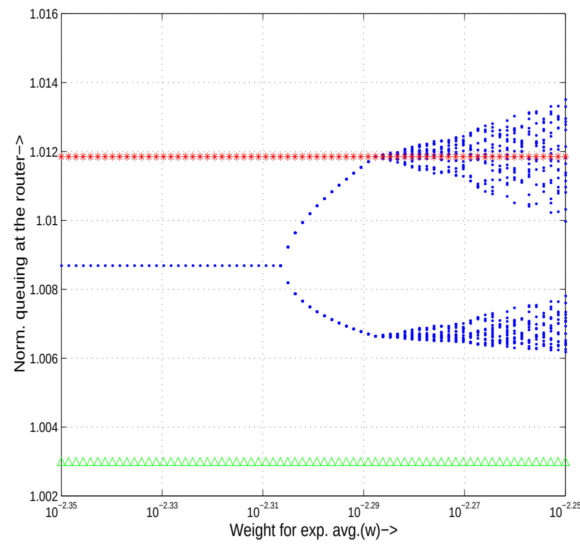

Fig. 7. Bifurcation diagram of average queue length w.r.t. w, $p_{max} = 1$

We also plot the Lyapunov exponents for the bifurcation scenario in fig. 5 where $p_{max} = 0.1$. Fig. 8 shows that in the beginning expoent is negative which corresponds to the fixed point. It slowly increases to zero near period doubling bifurcation and then goes negative again. Finally, it jumps to a positive value when the border collides with the periodic solution. Positive Lyapunov exponent conmfirms the existence of chaos.

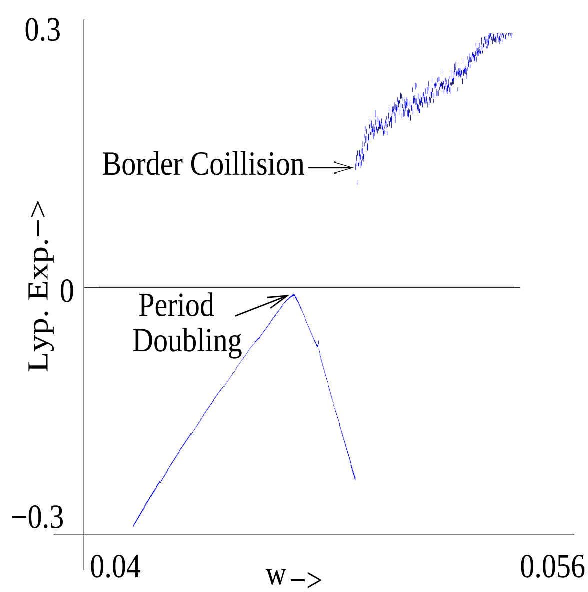

Fig. 8. Lyapunov exponent computed for average queue length w.r.t. w, $p_{max} = 0.1$

Lyapunov exponents for other two scenarios also exhibit similar behavior.

C. Effect of RED Control Parameter q_{min}

The following parameters are common to the next four bifurcation plots [5].

$$p_{max} = 0.3, \quad q_{max} = 100, \quad c = 1500kbps, \quad k = \sqrt{8/3}$$
$$B = 300 \text{ packets}, \quad R_0 = 0.1sec, \quad M = 0.5kb,$$
$$n = 20, \quad q_{min} = \text{bifurcation parameter}$$

0. $w = 2^{-5}$

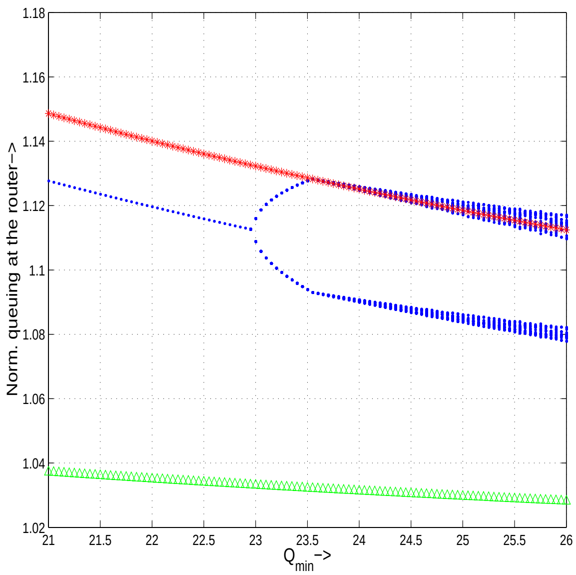

Fig. 9. Bifurcation diagram of average queue length w.r.t. q_{min}, $w = 2^{-5}$

1. $w = 2^{-6}$

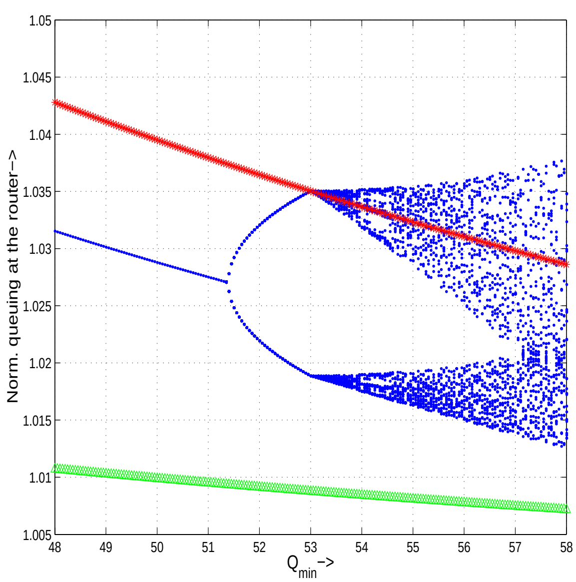

Fig. 10. Bifurcation diagram of average queue length w.r.t. q_{min}, $w = 2^{-6}$

Similar phenomena are exhibited in these four scenarios. Here also there is bifurcation sequence like $1 \longrightarrow 2 \longrightarrow chaos$ in figs. 9, 10 and 11 but scenario of fig. 12 shows $1 \longrightarrow 2 \longrightarrow 4 \longrightarrow chaos$. It should be noted that $2 \longrightarrow 4$ is not a smooth bifurcation like period doubling rather, it is a border collision bifurcation.

2. $w = 2^{-7}$

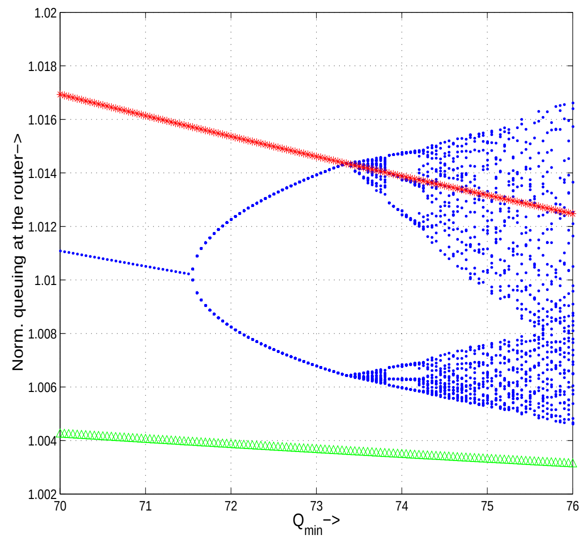

Fig. 11. Bifurcation diagram of average queue length w.r.t. q_{min},$w = 2^{-7}$

3. $w = 2^{-8}$

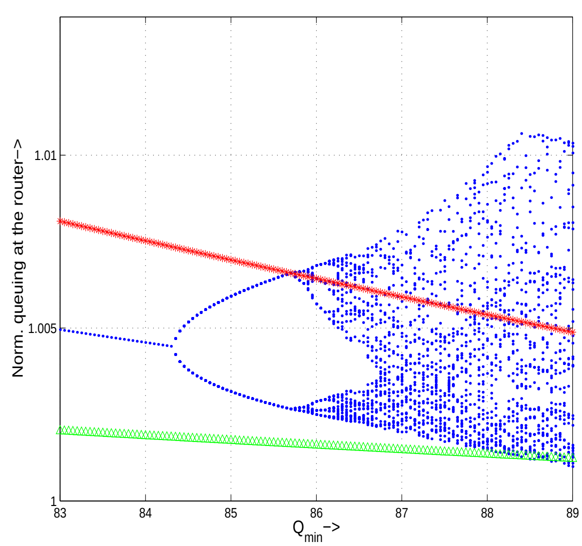

Fig. 12. Bifurcation diagram of average queue length w.r.t. q_{min},$w = 2^{-8}$

Finally, we plot the Lyapunov exponent corresponding to the the bifurcation scenario in fig. 10. Lyapunov exponent shown in fig. 13 also stays negative in the beginning like the other one plotted in fig. 8. In a similar fashion it increases to zero when the system goes through a period doubling bifurcation and again decreases when the system has a stable period two trajectory. Finally, it jumps to a positive value after border coliision bifurcation and stays there.

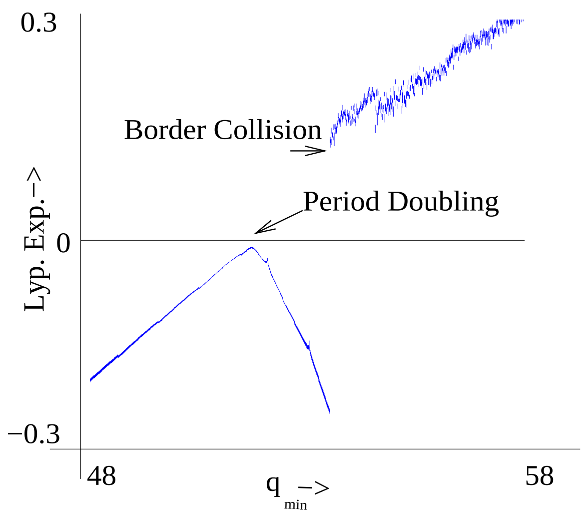

Fig. 13. Lyapunov exponent computed for average queue length w.r.t. q_{min}, $w = 2^{-6}$

VI. Discussion

We have demonstrated in this paper that instability in TCP-RED can be induced by the inherent nonlinear behavior of the network, rather than by discontinuity in the "queue or the control law" as has been believed so far citeaqm1. The subharmonic load batching very clearly indicates that the system can oscillate if the parameters are not properly tuned. We have also given a conservative criterion for stable parameter settings based on linearized stability analysis.

Although the simulations have indicated a period-two oscillation with amplitude within five percent of the nominal amplitude, the appearance of period doubling behavior is of significance. The importance of period doubling bifurcation is not in predicting the amplitude of the oscillation, but rather in explaining the routes to larger and more pronounced oscillations including chaos. Also, the results become still more significant when we think of the network as a cascade involving multiple routers. If one router starts oscillating, it may transfer this instability on the routers up or downstream depending on the major traffic direction. Also depending on traffic conditions this instability may propagate throughout the network with different time scales and a systemic oscillation can develop. This needs further study. In addition, although only analysis has been studied here, it would be interesting to consider implications of the approach for control and for protocol design.

Acknowledgments

This research has been supported in part by the Institute for Systems Research, University of Maryland and by the Office of Naval Research under Multidisciplinary University Research Initiative (MURI) Grant N00014-96-1-1123.

References

[1] Y. Korilis and A. Lazar, "Why is Flow Control Hard: Optimality, Fairness, Partial and Delayed Information," *Proc. 2nd ORSA Telecommunications Conference*, March 1992.

[2] S. Floyd and V. Jacobson, " Random Early Detection Gateways for Congestion Avoidance," *IEEE Trans. on Networking*, Vol.1, no. 7, pp. 397-413, 1993.

[3] M. Mathis, J. Semke, J. Mahdavi, and T. Ott, "The Macroscopic Behavior of the TCP Congestion Avoidance Algorithm," *Computer Communications Review*, Vol. 27, no. 3, 1997.

[4] J. P. Hespanha, S. Bohacek, K. Obraczka and J. Lee, "Hybrid Modeling of TCP Congestion Control," *Lecture notes in Computer Science* no. 2034 pp. 291-304, 2001.

[5] V. Firoiu and M. Borden, "A Study of Active Queue Management for Congestion Control," *Proc. of Infocom* 2000.

[6] V. Misra, W. Gong and D. Towsley, "A Study of Active Queue Management For Congestion Control," *Proc. of Sigcomm* 2000.

[7] C. Hollot, V. Misra, D. Towsley and W. Gong, "A Control Theoretic Analysis of RED," *CMPSCI Technical Report TR 00-41, 2000 and in Proc. of IEEE Infocom*, 2001.

[8] P. Kuusela, P. Lassilaand and J. Virtamo, "Stability of TCP-RED Congestion Control," *Sumitted for publication, available at http://www.tct.hut.fi/tutkimus/com2/publ/*, 2000.

[9] M. May, J. Bolot, C. Diot, B. Lyles , "Reasons Not to Deploy RED," *Proc. IWQoS'97*, 1997.

[10] D. Lin and R. Morris, "Dynamics of Random Early Detection," *Proc. of SIGCOMM*, 1997.

[11] M. di Bernardo, F. Garofalo, L. Glielmo and F. Vasca, "Analysis of Chaotic Buck, Boost and Buck-Boost Converters through Impact Maps," *Proceedings PESC97 (IEEE Power Electronics Specialist Conf.), St. Louis, USA*, 1997.

[12] S. Banerjee, P. Ranjan and C. Grebogi, "Bifurcations in two-dimensional piecewise smooth maps - Theory and applications in switching circuits," *IEEE Trans. on Circuits and Systems–I: Fundamental Theory and Applications*, Vol. 47, no. 5, pp. 633-643, 2000.

[13] S. Grishechkin, M. Devetsikiotis,I. Lambadaris and C. Hobbs, "On Catastrophic Behavior of Queueing Networks," *Workshop on Analysis and Simulation of Communication Networks, The Fields Institute for Research in Mathematical Sciences, Toronto*, 1998.

[14] A. Veres and M. Boda, "The Chaotic Nature of TCP Congestion Control," *Proc. of Infocom*,2000.

[15] A. Erramilli, L. J. Forys, "Oscillations and Chaos in a Flow Model of A Switching System," *IEEE Journal on Selected Areas in Communication*, Vol. 9, No. 2, pp. 171-178, Feb. 1991.

[16] A. Erramilli, L. J. Forys, "Traffic Synchronization Effects in Teletraffic Systems," *Proc. ITC-13, Copenhagen*, 1991.

[17] R. Johari and D. Tan, "End-to-End Congestion Control for the Internet: Delays and Stability," *available at http://www.statslab.cam.ac.uk/Reports/2000/2000-2.pdf*, 2000.