

Stability Study of the TCP-RED System Using Detrended Fluctuation Analysis

Xi Chen*, Siu-Chung Wong*, Chi K. Tse*, and Ljiljana Trajković†

*The Hong Kong Polytechnic University, Hong Kong

†Simon Fraser University, Vancouver, Canada

Abstract—It has been observed that the TCP-RED system may exhibit instability and oscillatory behavior. Control methods proposed in the past have been based on the analytical models that rely on statistical measurements of network parameters. In this paper, we apply the detrended fluctuation analysis (DFA) method to analyze stability of the TCP-RED system. The DFA has been used for detecting long-range correlations in seemingly non-stationary noisy signals. The key indicator emanating from DFA is known as the scaling exponent. By examining the variations of the DFA scaling exponent when varying system parameters, we quantify the stability of the TCP-RED system in terms of system's characteristics.

I. INTRODUCTION

Internet applications, such as World Wide Web, file transfer, and email, are transmitted using the Transmission Control Protocol (TCP) [1]–[3]. TCP allows multiple information sources to compete for a fair share of physical bandwidth allocated between two end-points of a connection. Active Queue Management (AQM) in bottleneck gateways plays an important role in controlling the traffic flow when the number of sources increases. The Random Early Detection (RED) [4] gateway was introduced to provide early feedback to the TCP senders by marking (or dropping) packets before buffers become full thus preventing large number of packets from being lost. However, it has been reported that TCP gateways employing RED may exhibit instability and oscillatory behavior [5]–[8] that degrade the transmission performance.

The TCP-RED algorithms have been implemented in the *ns-2* network simulator [9]. The *ns-2* simulator enables analysis and verification of communication networks in a controlled environment. A number of analytical models have been developed to characterize the TCP-RED system and analyze its oscillatory behavior [10]–[14]. The stability conditions derived using these analytical models often depend on statistical measurements of various parameters and their upper and/or lower bounds. These measured parameters may be non-stationary. Hence, a universal and reliable method, robust and simple for implementation, would facilitate characterizing the TCP-RED system for various scenarios of network connections.

Detrended fluctuation analysis (DFA) is a method for detecting long-range power-law correlations in seemingly non-stationary noisy/randomized signals. It has been widely used in computer science, biodynamics, bioinformatics, economics, and meteorology [15]–[18]. The outcome is usually given in terms of a parameter called *power-law scaling exponent*.

Given a time series (waveform), a chosen statistical variable is observed for various time scales. When plotted over various time scales, a typically seen straight line on a log-log scale implies a long-range power-law correlation for the observed variable. Its slope is equal to the scaling exponent. This line may display a deflection (*crossover*) at a certain time scale where the slope abruptly changes. The interpretations of scaling exponents and the crossovers are system dependent. In this paper, we apply the DFA method to the individual and aggregated traffic flows in a bottleneck RED gateway of a TCP-RED system which may exhibit a change of stability under variation of system parameters. We analyze the change in scaling exponents and the locations of the crossovers.

The paper is organized as follows. In Section II, we describe the TCP-RED system. In Section III, we illustrate the DFA method and analyze the long-range power-law correlation property of the queue length. A physical interpretation of the obtained scaling exponent values in relation to the waveforms of the queue length is provided in Section IV. We conclude with Section V.

II. OVERVIEW OF TCP-RED ALGORITHMS

A. Congestion Control Algorithm

We assume a window based flow control between the two end-points of a TCP connection. The TCP sender organizes a stream of data into a sequence of windows of packets. Each window contains w packets. When the TCP receives a packet from the sender, it returns an acknowledgment packet. Upon receiving the first acknowledgment packet from the last window of data sent, the TCP sender begins sending packets within the next window. The duration between sending a packet and receiving the acknowledgment is called the *round-trip-time* (RTT). Therefore, a TCP sender may send w packets for each RTT. If the window size w is adaptively selected to reach its maximum share of throughput according to the network conditions, the TCP communication eventually reaches its steady-state as a self-regulated RTT clocked system. Several TCP algorithms have been employed for adjusting the window size according to network conditions (Tahoe [1], Reno [2], Vegas [3], and their variants). Both TCP Tahoe and TCP Reno employ an Additive Increase Multiplicative Decrease (AIMD) algorithm for congestion avoidance. TCP Vegas uses a more proactive approach for controlling window size. In this study,

we consider TCP Reno, being the most widely implemented version of TCP.

The TCP congestion control mechanism employs four algorithms: slow start, congestion avoidance, fast retransmit, and fast recovery. TCP uses slow start to detect network bandwidth immediately following connection establishment or timeout. The algorithm starts with a window size w of one or two packets that doubles for each RTT when acknowledgments are received. When w exceeds a threshold, the algorithm enters the congestion avoidance phase. In a typical congestion avoidance phase, the window size w increases linearly by $1/w$ for each returned acknowledgment (additive increase). Thus, after a window of w packets is transmitted, w will be increased by 1 within one RTT. Window size w is reduced to half its current value (multiplicative decrease) if there is an indication of congestion (the receipt of three duplicated acknowledgments of a packet or the presence of an Explicit Congestion Notification (ECN) header information received from a gateway and forwarded in the acknowledgment sent by the receiver). The window then follows the additive increase algorithm. When three duplicated acknowledgments of a packet are received, the TCP sender enters the fast retransmit phase where lost packets are retransmitted without waiting for timeout. The fast recovery algorithm then retransmits all lost packets until a non-duplicate acknowledgment is received. The congestion avoidance phase follows, where the TCP sender attempts to optimize its fair share of the link bandwidth connecting the two TCP end-points. If at least one link on this communication path becomes the bottleneck, an AQM gateway regulates the traffic flow in this bottleneck link at its maximum capacity.

B. The RED Algorithm

The RED mechanism calculates a weighted moving average x_k of the current queue size q_k as:

$$x_k = (1 - \alpha)x_{k-1} + \alpha q_k, \quad (1)$$

where $\alpha \in (0, 1)$ is the filter resolution. A probability p_k is calculated based on the current value of x_k :

$$p_b = \begin{cases} 0 & 0 \leq x_k < X_{\min} \\ \frac{x_k - X_{\min}}{X_{\max} - X_{\min}} p_{\max} & X_{\min} \leq x_k \leq X_{\max} \\ p_{\max} + \frac{1 - p_{\max}}{X_{\max}} (x_k - X_{\max}) & X_{\max} < x_k \leq 2X_{\max} \\ 1 & 2X_{\max} \leq x_k \leq B \end{cases} \quad (2)$$

$$p_k = \frac{p_b}{1 - c_m p_b}, \quad (3)$$

where X_{\min} and X_{\max} are minimum and maximum thresholds of x_k , respectively; B is the buffer size; p_{\max} is the maximum threshold of p_k ; and c_m is the number of packets that arrived after the last marking. Each incoming packet is marked (or dropped) with probability p_k . The *target-queue-length* q_0 can, therefore, be maintained by properly choosing system parameters, such as X_{\min} , X_{\max} , and p_{\max} .

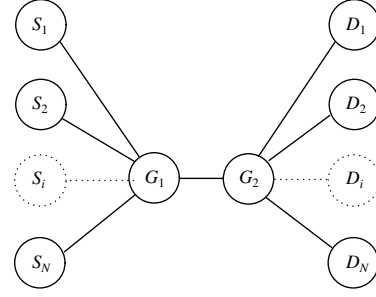


Fig. 1. A system of N TCP flows, from S_i to D_i , where $i = 1, 2, \dots, N$, passing through a common bottleneck link between G_1 and G_2 .

TABLE I

ns-2 SIMULATION PARAMETERS

<i>ns-2</i> parameter	value
N	170
C/N	1.5 Mbps
R_o	12 ms
r_o	19.46 ms
X_{\min}	256 packets
q_0	384 packets
X_{\max}	512 packets
Bottleneck delay	6 ms
RED redwait	false
TCP ECN	1

C. The TCP-RED System

We consider a system of N TCP flows passing through a bottleneck RED gateway shown in Fig. 1 [6], [8]. We assume that the system operates in its desired stable condition, satisfying a designed objective [4]. Without loss of generality, the system is assumed ECN capable. The N long-lived TCP connections are controlled by the ECN markings of the RED gateway, which aims to maintain an even distribution of the bandwidth in the bottleneck link among the N TCP flows.

The average queue size can be maintained at the target queue level $(X_{\min} + X_{\max})/2$. The system can be analyzed using small-signal linearization and perturbation for the calculation of instability boundaries [10].

The nonlinear dynamics of the aggregated flow at the RED gateway was analyzed using the *ns-2* network simulator and the parameters shown in Table I. As the RED filter resolution α varies, the system crosses the boundary of stability. Stable and unstable RED queue length waveforms of the TCP-RED system for two values of α are shown in Fig. 2 [10]. The waveforms observed over various time scales exhibit *self-similarity*. In this paper, we analyze its effect on the queue stability.

III. LONG-RANGE POWER LAW CORRELATIONS IN TCP FLOWS

A. Self-Similarity and DFA Method

Detrended fluctuation analysis (DFA) method [17], [18] is a popular approach to analyze self-similarity and long-range dependence (LRD) properties of signals.

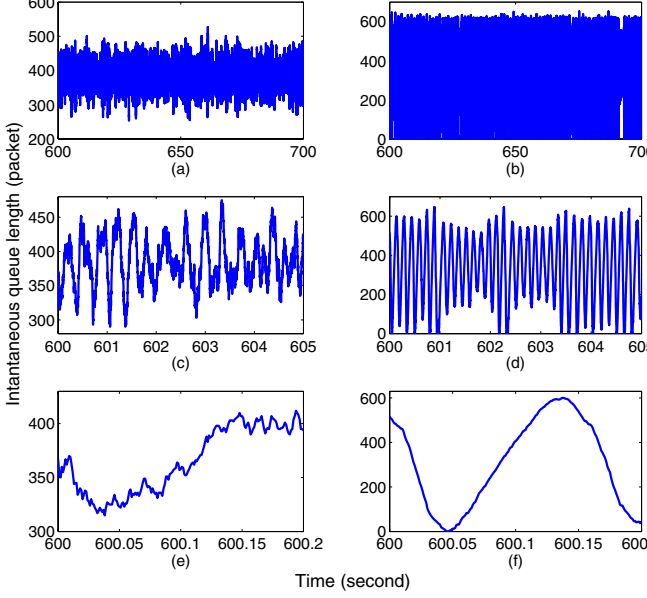


Fig. 2. Simulated RED queue length waveforms using *ns-2* simulator for filter resolutions $\alpha = 0.0001$ and $\alpha = 0.0008$ illustrate stable waveforms (a), (c), and (e) and unstable waveforms (b), (d), and (f), respectively. Figures (c) and (e) are enlarged views of (a) while (d) and (f) are enlarged views of (b). There are 170 TCP connections. Each connection shares a fixed bandwidth of 1.5 Mbps in the bottleneck link.

Consider a signal $s(i)$, where $i = 1, \dots, L$, and L is the length of the signal. An integrated time series $y(i)$ of $s(i)$ is obtained as:

$$y(i) = \sum_{j=1}^i (s(j) - \bar{s}), \quad (4)$$

where \bar{s} is the mean of $s(j)$ given as:

$$\bar{s} = \frac{1}{L} \sum_{j=1}^L s(j). \quad (5)$$

The integrated time series $y(i)$ is divided into m bins of equal length l , where $ml = L$. A least-squares fitted line $y_{k_l}(i)$ of $y(i)$, also called the local trend of $y(i)$, is calculated for the k th bin of length l , where $k = 1, \dots, m$. The series $y(i)$ is detrended by subtracting the local trend $y_{k_l}(i)$, for $i = (k-1)l + 1, \dots, kl$, as:

$$Y_l(i) = y(i) - y_{k_l}(i). \quad (6)$$

For the k th bin of length l , the root mean square (rms) fluctuation for the integrated and detrended time series is calculated as:

$$f_k(l) = \sqrt{\frac{1}{l} \sum_{j=(k-1)l+1}^{kl} [Y_l(j)]^2}. \quad (7)$$

Thus, the averaged rms fluctuation for the entire time series

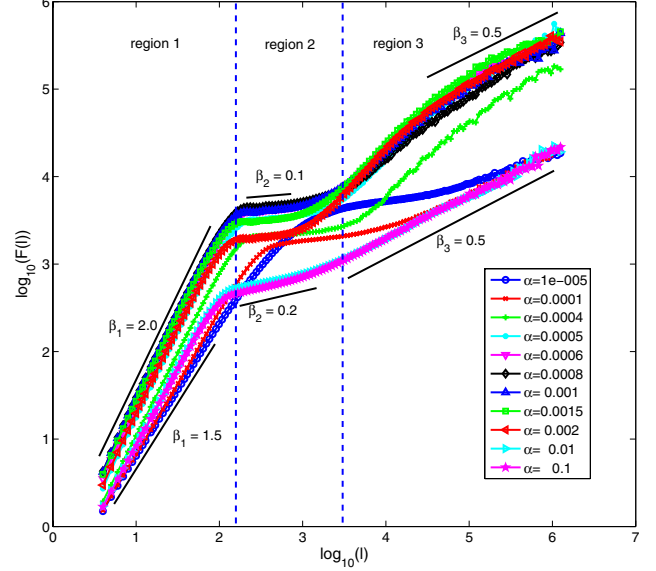


Fig. 3. DFA scaling functions obtained from RED instantaneous queue length for various choices of α .

is given as:

$$F(l) = \sqrt{\frac{1}{m} \sum_{k=1}^m f_k^2(l)}. \quad (8)$$

This computation is repeated for all time scales l to obtain the behavior of $F(l)$. Scale invariant signals with power-law correlations possess a power-law relationship between the rms fluctuation function $F(l)$ and the scale l , given by::

$$F(l) \sim l^\beta. \quad (9)$$

If $\log F(l)$ increases linearly with $\log l$, the slope of the line $\log F(l)$ versus $\log l$ gives the *scaling exponent* β . $F(l)$ is also called the scaling function. We analyze the scale exponent β for the TCP-RED system.

B. Scaling Exponent and Stability

Fig. 3 illustrates application of the described DFA method to the waveforms of the queue length in the TCP-RED system with parameters shown in Table I. The instantaneous queue length in the RED gateway is recorded every 1 millisecond over the 5,000 seconds interval. The scaling exponent (slope of the line) changes with varying α . For each DFA curve, we observe two deflections (crossovers) and three linear regions. We thus obtain three scaling exponents for each DFA curve over the entire range of bin size l , labeled β_1 , β_2 , and β_3 for regions 1, 2, and 3, respectively. A plot of exponent β_1 vs. the RED filter resolution α is shown in Fig. 4. Also shown are the stability boundaries for the TCP-RED system [10]. The value of β_1 , which varies between 1.5 and 2.0, provides a clear indication of the stability of the TCP-RED system. Values of β_1 below ~ 1.85 correspond to stable region and above ~ 1.85 correspond to unstable region of operation.

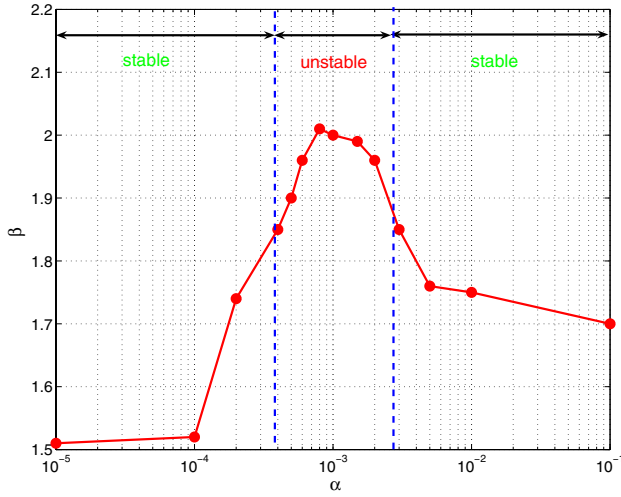


Fig. 4. DFA exponents in region 1 of Fig. 3 for varying α .

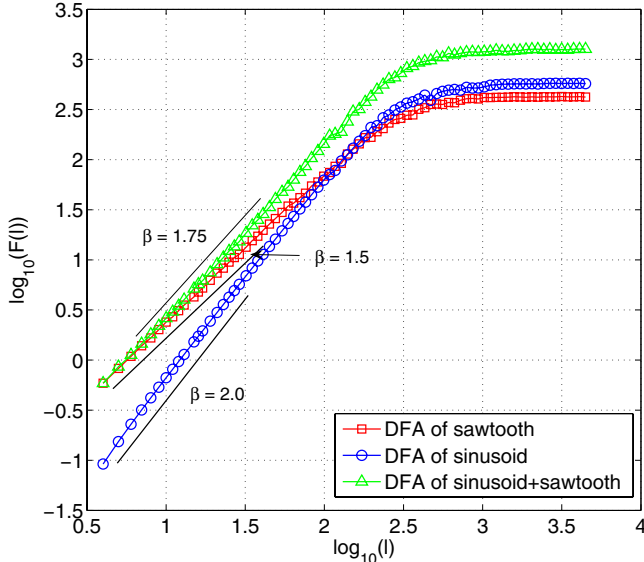


Fig. 5. DFA method applied to stationary signals.

IV. INTERPRETATION FROM A WAVEFORM VIEWPOINT

In the analysis of time series, the DFA exponent value allows a consistency characterization of seemingly nonstationary time series [18]. Two time series are *DFA similar* if they have nearly equal DFA exponent values. Hence, if a stationary time series that is *DFA similar* to the RED queue length time series could be identified, the two would have consistent characteristics.

The unstable RED queue length waveform shown in Fig. 2(e) resembles a sinusoidal wave. The DFA curve for a sinusoidal waveform with scaling exponent 2 is shown in Fig. 5. The crossover depends on the frequency of the sinusoidal wave. Also shown in Fig. 5 is the DFA curve for a sawtooth waveform with a scaling exponent 1.5. Such a sawtooth waveform is the native waveform of the stable

AIMD congestion algorithm. DFA exponent values between 1.5 and 2.0 may be generated by combining “sinusoid” and “sawtooth”. For example, a DFA exponent value 1.75 may be obtained by combining “sinusoid” and “sawtooth” with an amplitude ratio of 2:1, as shown in Fig. 5.

V. CONCLUSION

Based on the data collected from the *ns-2* simulations, long-range power-law correlations of the queue length waveforms in the RED gateway have been studied using the DFA method. The results show that the scaling exponent varies with the stability of the RED gateway. The scaling exponent is independent of the stationarity of the queue length and, hence, may be used as an indicator for the stability of the TCP-RED system.

REFERENCES

- [1] V. Jacobson, “Congestion avoidance and control,” *ACM Computer Communication Review*, vol. 18, no. 4, pp. 314–329, Aug. 1988.
- [2] V. Jacobson (1990, Apr. 30). “Modified TCP congestion avoidance algorithm” [Online]. Available: <ftp://ftp.ee.lbl.gov/email/vanji.90apr30.txt>.
- [3] L. S. Brakmo and L. L. Peterson, “TCP Vegas: end-to-end congestion avoidance on a global Internet,” *IEEE J. Selected Areas in Communications*, vol. 13, no. 8, pp. 1465–1480, Oct. 1995.
- [4] 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.
- [5] V. Misra, W. B. Gong, and D. Towsley, “Fluid-based analysis of a network of AQM routers supporting TCP flows with an application to RED,” in *Proc. ACM/SIGCOMM*, Stockholm, Sweden, Aug. 2000, pp. 151–160.
- [6] 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.
- [7] S. Floyd, R. Gummadi, and S. Shenker, (2001) “Adaptive RED: an algorithm for increasing the robustness of RED’s active queue management” [Online]. Available: <http://www.icir.org/floyd/red.html>.
- [8] P. Ranjan, E. H. Abed, and R. J. La, “Nonlinear instabilities in TCP-RED,” *IEEE/ACM Trans. Netw.*, vol. 12, no. 6, pp. 1079–1092, Dec. 2004.
- [9] The network simulator ns-2 [Online]. Available: <http://www.isi.edu/nsnam/ns>.
- [10] X. Chen, S. C. Wong, C. K. Tse, and Lj. Trajković, “Stability analysis of RED gateway with multiple TCP Reno connections,” in *Proc. Int. Symp. Circ. Syst.*, New Orleans, USA, May 2007, pp. 1429–1432.
- [11] P. Tinnakornsrisuphap and A. M. Makowski, “Limit behavior of ECN/RED gateways under a large number of TCP flows,” in *Proc. IEEE INFOCOM*, San Francisco, CA, USA, Mar. 2003, pp. 873–883.
- [12] P. Kuusela, P. Lassila, J. Virtamo, and P. Key, “Modeling RED with idealized TCP sources,” in *Proc. 9th IFIP Conf. Performance Modeling and Evaluation of ATM & IP Networks*, Budapest, Hungary, June 2001, pp. 155–166.
- [13] P. Kuusela, P. Lassila, and J. Virtamo, “Stability of TCP-RED congestion control,” in *Proc. ITC-17*, Brazil, Dec. 2001, pp. 655–666.
- [14] T. Ziegler, S. Fdida, and C. Brandauer, (2000). “Stability criteria for RED with TCP traffic,” Technical Report [Online]. Available: http://www.salzburgresearch.at/research/gfx/report_redstability_00.pdf.
- [15] D. R. Figueiredo, B. Liu, A. Feldmann, V. Misra, D. Towsley, and W. Willinger, “On TCP and self-similar traffic,” *Performance Evaluation*, vol. 61 no. 2-3, pp. 129–141, July 2005.
- [16] M. Masugi, “Recurrence plot-based approach to the analysis of IP-network traffic in terms of assessing nonstationary transitions over time,” *IEEE Trans. Circ. Syst. I, Reg. Papers*, vol. 53, no. 10, pp. 2318–2326, Oct. 2006.
- [17] C. K. Peng, S. Havlin, H. E. Stanley, and A. L. Goldberger, “Quantification of scaling exponents and crossover phenomena in nonstationary heartbeat time series,” *Chaos*, vol. 5, pp. 82–90, Mar. 1995.
- [18] K. Hu, P. C. Ivanov, Z. Chen, P. Carpena, and H. E. Stanley, “Effect of trends on detrended fluctuation analysis,” *Phys. Rev. E*, vol. 64, 011114, July 2001.