2005 International Symposium on Nonlinear
Theory and its Applications (NOLTA2005)
Bruges, Belgium, October 18-21, 2005

Characteristic Oscillation and Intermittency in TCP-RED Gateway

Siu C. Wong, Chi K. Tse and Francis C.M. Lau

Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, Hong Kong
URL: http://chaos.eie.polyu.edu.hk

Abstract—It has been found that a bottleneck RED gateway can become oscillatory when regulating a number of TCP flows. We develop a dynamic model to qualitatively describe the characteristic frequency and intermittency of the oscillation in the RED gateway. The results have been verified with the **ns** simulator.

1. Introduction

The Internet has become an important medium of information transfer nowadays. The TCP/IP protocol suite and the interconnected gateways provide reliable channels for the flow of information which are evenly shared among different connections. The utilization of a connection reaches its maximum when there is a bottleneck link formed along the path of the information flow. The bottleneck link is regulated by a gateway which will promote its bandwidth being distributed equally among all connections.

The two endpoints of a TCP connection are communicating using a window based flow. The TCP sender partitions a stream of data into a sequence of separated windows of packets. The packets in a window are sent in one batch of size w. For each successfully received packet, the TCP receiver returns an acknowledgment packet. Upon receiving the first packet acknowledgment from the last window, the TCP sender starts sending packets in the next window. The duration between the sending of a packet and the receiving of the acknowledgment of this packet is called the *round-trip-time* (RTT). Therefore, a TCP sender can 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 will eventually reach its steady state as a self-regulated RTT clocked system.

There are several TCP algorithms for adjusting the window size according to the network conditions, e.g., Tahoe [1], Reno [2], and Vegas [3] or their variants. Both Tahoe and Reno use Additive Increase Multiplicative Decrease (AIMD) algorithm for congestion avoidance. Vegas uses a more proactive approach for controlling window size. Among them, Reno is the most widely implemented version of TCP. Our focus here is the AIMD congestion avoidance algorithm based on Tahoe and Reno implementations.

In a typical congestion avoidance algorithm, the window size w increases linearly by $1/w$ for each acknowledgment returned (Additive Increase). In this way, after successfully transmitted w packets (a window of packets), w will accumulatively increase by 1 within one RTT. When there is congestion indication, either by receiving three duplicated acknowledgments of a particular packet sent or by an Explicit Congestion Notification (ECN) header information injected from a gateway and forwarded in the acknowledgment by the receiver, w is reduced to half its current value (Multiplicative Decrease). After that the window follows the additive increase algorithm.

The congestion avoidance algorithm in a TCP sender tries to optimize its fair share of bandwidth on the links connecting the two TCP end points. In this way, at least one of the links making up the path of the communication channel becomes the bottleneck link. An active queue management (AQM) gateway in this bottleneck link serves the purpose of regulating the flow through the link at its maximum capacity. The drop tail gateway is the simplest AQM gateway. It drops the incoming packets when its queue buffer becomes full. The drop tail gateway may suffer from poor communication with the TCP senders that a large amount of packets can be dropped as the senders keep sending and the gateway keep dropping packets until a retransmission time-out occurs in the senders. Random Early Detection (RED) [4] gateway was introduced to give early feedback to the TCP senders by marking (or dropping) packets before the buffer becomes full hence preventing large amount of packets from being dropped. However, it is found that TCP-RED may have instability and oscillatory problems [5, 6, 7, 8, 9], which certainly degrade the transmission performance. We have performed simulations using the **ns** simulator [12], as shown in Fig. 1, and have found oscillation at frequencies related to the TCP sending rate as well as intermittent behaviors.

In this paper, we will try to develop a dynamic model that adequately explains these behaviors of the TCP-RED algorithm.

2. Overview of Operation

We consider a system of N TCP flows passing through a bottleneck RED gateway [6, 8, 9], as shown in Fig. 2. It is assumed that the system is operating in its desired stable condition, satisfying the usual designed goal [4]. Without loss of generality, the system is assumed ECN capable. The N long-live TCP connections are essentially controlled by the ECN markings of the RED gateway. The ECN markings are controlled by the RED gateway to maintain an evenly distribution among the N TCP flows, making them

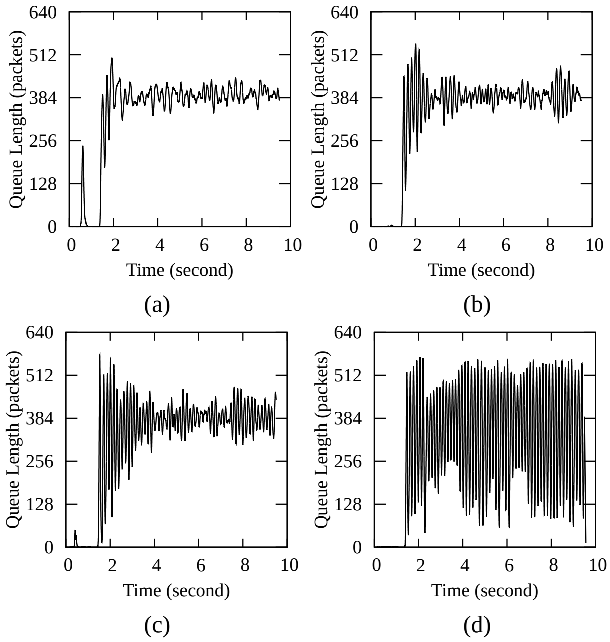

Figure 1: Simulated queue length waveforms of TCP-RED using the **ns** simulator. (a) 96, (b) 128, (c) 160 and (d) 170 TCP connections. Each connection shares a fixed bandwidth of 1.5Mb/s in the bottleneck link.

share evenly the bandwidth of the bottleneck link.

The dynamics of the queue can be explained using the fluid model [5], which can be described as follows:

$$\frac{dw(t)}{dt} = \frac{1}{r(t)} - \frac{w(t)}{2}\frac{w(t-r(t))}{r(t-r(t))}p(t-r(t)) \quad (1)$$

$$\frac{dq(t)}{dt} = N\frac{w(t)}{r(t)} - C \quad (2)$$

$$\frac{dx(t)}{dt} = \frac{\ln(1-\alpha)}{r(t)}(x(t)-q(t)), \quad (3)$$

$$p_b(t) = \begin{cases} 0 & 0 \le x(t) < X_{\min}, \\ \frac{x(t)-X_{\max}}{X_{\max}-X_{\min}}p_{\max} & X_{\min} \le x(t) \le X_{\max}, \\ p_{\max} + \frac{1-p_{\max}}{X_{\max}}(x(t) - X_{\max}) & X_{\max} < x(t) \le 2X_{\max}, \\ 1 & 2X_{\max} \le x(t) \le B \end{cases} \quad (4)$$

$$p(t) = \kappa p_b(t) \quad (5)$$

$$r(t) = \frac{q(t)}{C} + R_0 \quad (6)$$

where

$w(t)$ = averaged instantaneous window size (in packets) of the TCP sources,
$r(t)$ = round trip time,
$q(t)$ = averaged instantaneous queue length (in packets),
$x(t)$ = filtered queue length after removal of short bursts,

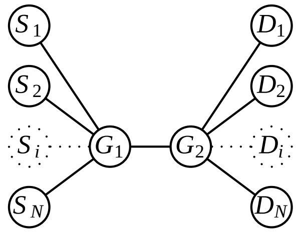

Figure 2: A system of N TCP flows, from S_i to D_i, where $i = 1, 2, \cdots, N$, passing through a common bottleneck link between G_1 and G_2.

$p(t)$ = marking probability,
α = filter resolution ($0 < \alpha < 1$),
κ = a proportionality constant dependent upon the detailed implementation of the RED algorithm,
$X_{\max}$ = maximum threshold of $x(t)$,
$X_{\min}$ = minimum threshold of $x(t)$,
$p_{\max}$ = maximum threshold of $p(t)$,
R_0 = propagation delay,
C = bottleneck bandwidth,
B = maximum physical queue length.

The above model has been analyzed previously using small-signal linearization and perturbation [10, 11]. Our aim is to study the detailed dynamics of the averaged TCP window size w and to explain its influence on the oscillation of the RED queue size.

3. Characteristic Frequency of Oscillation

In the steady state, the TCP algorithm uses congestion avoidance to maintain its share of bandwidth. Figure 3 (a) shows an ideal TCP window steady-state waveform of the TCP algorithm with congestion avoidance. It has been shown [13] that the TCP sources switch their sending rate at a period of T given by

$$T = \frac{w_{\max}R}{2} \quad (7)$$

where $w_{\max}$ is the maximum of $w(t)$ and R is the average of $r(t)$, within a TCP sender period. We recall from Fig. 1 that the **ns** simulations reveal a characteristic frequency of the queue length which is closed to $\frac{1}{T}$ and also intermittence behavior. In the following we try to explain this phenomenon.

We begin with introducing a disturbance to the ECN marking time of a period, such that $w_{\max} \to (1+\delta_0)w_{\max}$, where $-0.5 \le \delta_0 \le 0.5$. In practice, there will always have disturbance arising from the random process associated with the marking probability [4]. As the disturbance develops continuously, the RED algorithm should make sure that the average flow rate is maintained at $3w_{\max}/4R$ after nT, where $n = 1, 2, 3, \cdots$ [13]. Referring to Fig. 3 (c), we formulate two iterative functions to describe the total number of packets sent within a period (i.e., flow rate) and the corresponding period:

$$F(\delta_0,\delta_1) = (3+\delta_0+2\delta_1)\frac{w_{\max}}{4} \quad (8)$$

$$T(\delta_0,\delta_1) = (1-\delta_0+2\delta_1)\frac{w_{\max}R}{2} \quad (9)$$

where $T(\delta_0,\delta_1) > 0$. Mathematically, we may find n such that the average flow rate is maintained at $3w_{\max}/4R$, i.e.,

$$\frac{\sum_{k=1}^{n} F(\delta_{k-1},\delta_k)T(\delta_{k-1},\delta_k)}{\sum_{k=1}^{n} T(\delta_{k-1},\delta_k)} = \frac{3w_{max}}{4R}, \quad (10)$$

$$\sum_{k=1}^{n} T(\delta_{k-1},\delta_k) = nT, \quad \text{and} \quad (11)$$

$$2\delta_k - \delta_{k-1} \geq -1 \quad \text{for } k=1,...,n \quad (12)$$

First, it is readily shown that there is no solution for $n = 1$ and $n = 2$. This implies that the system cannot recover within the next one or two consecutive periods for any disturbance. The solution for $n = 3$ can be calculated as follows:

$$\delta_1 = \frac{3l}{2} + \frac{\delta_0}{2} - \delta_3 \mp \frac{\Lambda}{2\sqrt{3}} \quad (13)$$

$$\delta_2 = \frac{3l}{2} + \frac{\delta_0}{2} - \delta_3 \pm \frac{\Lambda}{2\sqrt{3}} \quad (14)$$

where l is any integer, and

$$\Lambda^2 = -18l - 27l^2 - 8\delta_0 - 18l\delta_0 - \delta_0^2 + 8\delta_3 + 36l\delta_3 + 12\delta_0\delta_3 - 20\delta_3^2 \quad (15)$$

It can be shown that δ_1 and δ_2 are solvable from (12), (13) and (14) only when $l = 0$. A plot of δ_3–δ_0 for $\Lambda = 0$ at $l = 0$ gives the boundary curve of possible solutions. Solutions are confined to the region left of this curve, i.e., region containing point A in Fig. 3 (b). The line $\delta_3 = \delta_0$ touches the boundary at (0,0). Moreover, imposing (12) further restricts the solution area to region S as shown in Fig. 4. Suppose initially $\delta_0 < 0$, we have $\delta_3 > \delta_0$, e.g., point A in Fig. 3 (b), the solutions stay in region S as shown in Fig. 4.

In a likewise manner, we can develop solutions for $n \geq 4$ (omitted here). In all these cases, we observe that while the long-term average flow rate and window period are maintained at $3w_{\max}/4R$ and T, respectively, the instantaneous average flow rate and window period deviate significantly from these long-term average values. In the next section we will discuss how this result affects the characteristic frequency of the TCP flow observed in a RED gateway.

4. Discussions

4.1. Characteristic Frequency

The TCP source window has a period given by (9) and indicated in Fig. 3 (a). When disturbed, the period fluctuates as in Figs. 3 (c) and (d). The averaged period is still given by (7). Now consider paths B and C of Figs. 3 (c) and (d), where there may be some periodic orbits of

$$T(\delta_i,\delta_{i-1}) \approx R \quad \text{for some} \quad i = 1,2,3 \quad (16)$$

When there is disturbance that increases the flow arriving at the gateway, and as the flow reaches its maximum, the gateway will issue more ECN markings to decrease the TCP flow. These additional ECN commands will be effective after a delay of one round-trip-time R. The resulting flow contains small periods and large periods which are randomly interleaved. The smaller periods may disappear as they "merge" with the adjacent much longer periods due to a low filter resolution α of the gateway's queue buffer. This explains why characteristic frequencies of oscillation of the queue size somewhat lower than $1/T$ are observed. On the contrary, for a high filter resolution α, the observed characteristic frequencies become higher.

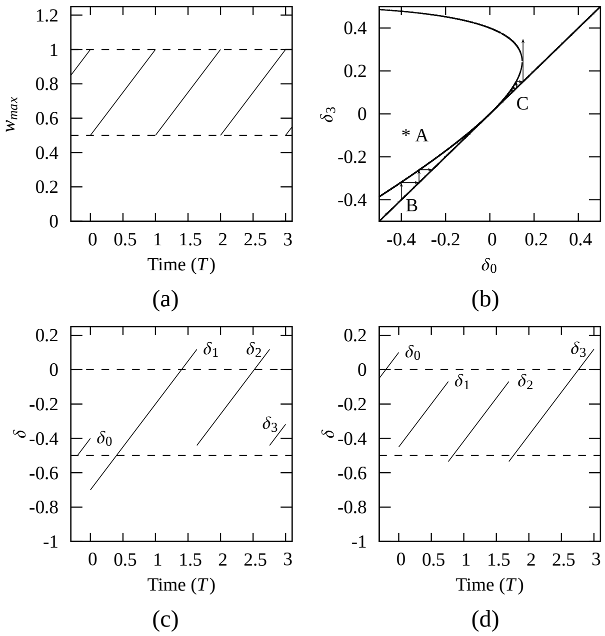

Figure 3: (a) Steady-state waveform of TCP sender's window; (b) boundary of disturbance and its iterative paths; (c) steady-state waveform of TCP sender's window for path B and (d) for path C.

4.2. Intermittency

For the purpose of illustration, let us assume that, for instance, the solution of δ_3 always follows the boundary curve. Thus, we will end up with the iterations indicated by path B as shown in Fig. 3 (b) and the corresponding waveform in Fig. 3 (c). As shown in Fig. 3 (b), the disturbance is converging toward the origin (0,0). If there is a positive disturbance, it will diverge along path C, as illustrated in Figs. 3 (b) and (d). This constitutes a typical intermittency phenomenon.

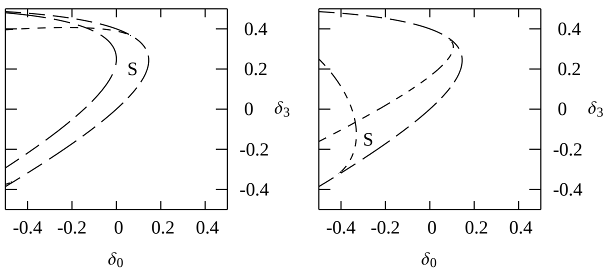

Figure 4: Solution space S of δ_0 and δ_3 when (12) is imposed. (a) First set of solutions of (13) and (14) when "−" in (13) and "+" in (14) are used; (b) second set of solutions of (13) and (14) when "+" in (13) and "−" in (14) are used.

Table 1: Simulation parameters

Parameter	Value
N	96,128,160, and 170
$\frac{C}{N}$	1.5 Mbps
X_{max}	512 packets
q_0	384 packets
X_{min}	256 packets
α	0.001
Bottleneck delay	6 ms
RED redwait_	false
TCP ecn_	1

However, in reality, the iterative path of δ_0 to δ_3 does not follow exactly the boundary curve resembling path B or C in Fig. 3 (b). In fact, the solution is within region S in Fig. 4. Note that solution space S is plotted using (12). The disturbance will either decay or diverge faster with $|\delta_2 - \delta_1| = \Lambda/\sqrt{3}$, as given by (13) and (14) for a non-zero Λ. Thus, we should observe a weaker intermittency in the flow.

4.3. Experimental Observation

The parameters in Table 1 are used for the **ns** simulation shown in Fig. 1. The natural TCP sender's frequency, the observed frequency and the averaged number of periods within an intermittent period are given in Table 2. Also, the effect of filter resolution α is depicted in Table 3, which shows the characteristic frequency of the observed oscillation for different values of α for $N = 128$. As explained earlier, for smaller α, the removal of many shorter periods by the filter makes the characteristic frequency appear lower, and vice versa for larger α.

5. Conclusion

We have developed a model to explain the observed periodicity and intermittency in the instantaneous queue size of the TCP-RED system. The model can qualitatively explain the oscillatory behaviors of RED queue with a number of identical TCP connections.

Table 2: Simulation Results

Parameter	Value	Value	Value	Value
N	96	128	160	170
$\frac{q_0}{N}$	4	3	2.4	1.6
w_0	11.9	10.11	9.292	-
R	34.3 ms	29.1 ms	26.76 ms	-
$\frac{1}{T}$	3.67 Hz	5.10 Hz	6.02 Hz	-
f_q	4.4 Hz	5.28 Hz	6.86 Hz	-

Table 3: Simulation Results for different α at $N = 128$

Parameter	Value	Value	Value	Value
α	0.005	0.001	0.0005	0.0001
$\frac{1}{T}$	5.20 Hz	5.10 Hz	5.25 Hz	5.22 Hz
f_q	7.00 Hz	5.28 Hz	5.25 Hz	2.67 Hz

References

[1] V. Jacobson, "Congestion avoidance and control," *Proc. ACM SIGCOMM*, pp. 314-329, 1988.

[2] V. Jacobson, "Modified TCP congestion avoidance algorithm," ftp://ftp.ee.lbl.gov/email/vanj.90apr30.txt, 1990.

[3] L. S. Brakmo and L. L. Peterson, "TCP Vegas: end-to-end congestion avoidance on a global Internet," *IEEE J. Selected Areas in Comm.*, vol. 13, no. 8, pp.1465-80, 1995.

[4] S. Floyd and V. Jacobson, "Random Early Detection gateways for congestion avoidance," *IEEE/ACM Trans. Networking*, vol. 1, no. 4, pp. 397-423, 1997.

[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," *Proc. ACM/SIGCOMM*, 2000.

[6] V. Firoiu and M. Borden, "A study of active queue management for congestion control," *Proc. IEEE Infocom*, 2000, http://www.ieee-infocom.org/2000.

[7] S. Floyd, R. Gummadi, and S. Shenker, "Adaptive RED: an algorithm for increasing the robustness of RED's active queue management," http://www.icir.org/floyd/red.html, 2001.

[8] P. Ranjan and E. H. Abed, "Nonlinear instabilities in TCP-RED," *Proc. IEEE Infocom*, http://www.ieee-infocom.org/2002.

[9] L. Chen, X. F. Wang, and Z. Z. Han, "Controlling bifurcation and chaos in Internet congestion control model," *Int. J. Bifurcation Chaos*, vol. 14, no. 5, pp. 1863-1876, 2004.

[10] C. V. Hollot, V. Misra, D. Towsley, and E. B. Gong, "A control theoretic analysis of RED," *Proc. IEEE Infocom*, 2001, http://www.ieee-infocom.org/2001.

[11] C. V. Hollot, and Y. Chait, "Nonlinear stability analysis for a class of TCP/AQM networks," *Proc. IEEE Conf. Decision and Control*, 2001.

[12] Network Simulator – ns-2, http://www.isi.edu/nsnam/ns/.

[13] M. Mathis, J. Semke, J. Mahdavi, and T. Ott, "The macroscopic behavior of the TCP congestion avoidance algorithm," *Comp. Comm. Rev.*, vol. 27, no. 3, 1997.