# Buffer Requirements for Zero Loss Flow Control with Explicit Congestion Notification

Chunlei Liu Raj Jain
Department of Computer and Information Science
Raj Jain is now at Washington University in Saint Louis, jain@cse.wustl.edu http://www.cse.wustl.edu/~jain/

**Abstract**

Current TCP flow control depends on packet losses to find the workload that a network can support. Packet drops not only reduce TCP performance, but also add large transfer delay to the packets. Instead of dropping the overflowed packets, Explicit Congestion Notification (ECN) detects incipient congestion and notifies the sources to reduce their windows when the queue length exceeds a threshold. In this paper, we derive closed-form formulae for requirements on threshold and buffer size to achieve zero packet loss and full link utilization. Simulation results that verify our analysis are presented. The impacts of average queue length and Random Early Detection are also discussed.

## 1 Introduction

Current TCP flow control depends on packet losses to find the workload that a network can support. The source starts with a window of one packet and doubles window in every RTT until a packet is lost. Then the source reduces the window and performs congestion avoidance [1, 2]. Studies [3, 4, 5, 6, 7] show the bandwidth of the TCP connection is limited by packet loss probability. Packet losses not only increase the traffic in the network, but also add large transfer delay.

Explicit Congestion Notification (ECN) proposed in [8, 9] provides a light-weight mechanism for routers to send direct congestion indication to the source. It makes use of two experimental bits in the IP header and two experimental bits in the TCP header. When the queue length exceeds a threshold, the incoming packet is marked. When the marked packet is received, the receiver marks the acknowledgment (called an *ECN-Echo*) to send congestion notification back to the source. Upon receiving the ECN-Echo, the source halves its congestion window to alleviate the congestion. The window reduction is done only once in a window of packets [8]. In the next RTT period, the window will not be increased in response to acknowledgment [10].

Choosing an appropriate threshold and buffer size is critical to the performance. However, RFC 2481 and other papers did not specify the threshold and buffer requirements. Based on a simplified congestion detection model, this paper analyzes the queue dynamics at the congested router. Closed-form formulae for threshold and buffer requirements to achieve zero packet loss and full link utilization are derived.

The content of this paper is organized as follows. In section 2, we describe the assumptions and goals for our analysis. Dynamics of queue growth for one connection and multiple connections is studied in section 3 and 4. In section 5, simulation results verify our conclusions are presented. Finally, in section 6, we discuss some refinements and directions for further study.

## 2 Assumptions and Goals

ECN is used together with TCP flow control mechanisms like slow start and congestion avoidance [2]. When the acknowledgment is not marked, the source follows existing TCP algorithms to send data and update window. Upon the receipt of an ECN-Echo, the source halves its congestion window and reduces the slow start threshold. Although our algorithm aims for zero loss, in case of a packet loss, the source still follows TCP algorithm to reduce the window and retransmit the lost packet.



Ohio State University, Computer Science and Engineering Department
Technical Report, September 1999

*Chunlei*

are received by the source. All packets between the source and router have entered the congested router or have been sent downstream. As shown in Figure 1, the pipe length from the congested router to the receiver, and then back to the source is $r - t_p$. The number of downstream packets and outstanding acks are $(r - t_p)d$. The rest of the $w(t - t_p)$ unacknowledged packets are still in the congested router. So the queue length is

$$Q(t) = w(t - t_p) - t_p d - (r - t_p)d = w(t - t_p) - rd. \tag{2}$$

Q.E.D.

Notice that in the above theorem, we did not use the number of packets between the source and the congested router to estimate the queue length, because the packets downstream from the congested router and the acks on the reverse link are equally spaced, but the packets between the source and the congested router are not.

## 3.1 Slow Start Phase

Using Theorem 1, we can study the queue growth from window changes. In slow start phase, the congestion window doubles in every RTT. Suppose the packet $P$ that increases the queue length over the threshold $T$ is sent at time $s_0$, and it arrives at the congested router at time $t_0$, its acknowledgment, which is an ECN-echo, is received at the source at time $s_1$. The window is reduced at time $s_1$. Also suppose the last packet before the window reduction is sent at time $s_1^-$ and arrives at the congested router at time $t_1^-$.

We need to consider two cases separately: when $T$ is large and when $T$ is small. When $T$ is reasonably large (about $rd$) such that the buildup of a queue of size $T$ needs $r$ time, the assumption in Theorem 1 is satisfied. We have

$$T + 1 = Q(t_0) = w(t_0 - t_p) - rd = w(s_0) - rd, \tag{3}$$

so

$$w(s_0) = T + rd + 1. \tag{4}$$

Since the time elapse between $s_0$ and $s_1$ is an RTT, if $P$ were not marked, the congestion window would increase to $2w(s_0)$. Since $P$ is marked, the congestion window before receiving the ECN-Echo is

$$w(s_1^-) = 2w(s_0) - 1 = 2(T + rd) + 1. \tag{5}$$

When the last packet sent under this window reaches the router at time $t_1^-$, the queue length is

$$Q(t_1^-) = w(s_1^-) - rd = 2w(s_0) - 1 - rd = 2T + rd + 1. \tag{6}$$

Upon the receipt of ECN-Echo, congestion window is halved. The source can not send any more packets before half of the packets are acknowledged. So $2T + rd + 1$ is the maximum queue length.

When $T$ is small, $rd$ is an overestimate of the downstream packets and acks on the reverse link $r$ time from the router.

$$w(s_0) = T + 1 + \textit{number of downstream packets and acks } \leq T + rd + 1. \tag{7}$$

Therefore,

$$Q(t_1^-) = w(s_1^-) - rd = (2w(s_0) - 1) - rd \leq 2(T + rd + 1) - 1 = 2T + rd + 1. \tag{8}$$

So, in both cases, $2T + rd + 1$ is an upper bound of queue length that can be reached in slow start phase.

**Theorem 2** *In a TCP connection with ECN congestion control, if the fixed round trip time is $r$, the bottleneck link rate is $d$, and the bottleneck router uses threshold $T$ for congestion detection, then the maximum queue length can be reached in slow start phase is less than or equal to $2T + rd + 1$.*

As we analyze above, when $T$ is large, the bound $2T + rd + 1$ is tight. Since the queue length in congestion avoidance phase is smaller, this bound is actually the buffer size requirement.

### 3.2 Congestion Avoidance Phase

In the congestion avoidance phase, congestion window increases roughly by one in every RTT. Using the same timing variables as in the previous subsection, we have

$$w(s_0) = Q(t_0) + rd = T + rd + 1. \tag{9}$$

The congestion window increases roughly by one in an RTT but packet $P$ is marked, so

$$w(s_1^-) = T + rd + 1. \tag{10}$$

The last packet sent before the window reduction still sees a queue length of $T + 1$:

$$Q(t_1^-) = w(s_1^-) - rd = T + 1. \tag{11}$$

After the window is reduced,

$$w(s_1) = (T + rd + 1)/2. \tag{12}$$

The first packet sent under the new window see a queue length of

$$Q(t_1) = w(s_1) - rd = (T + rd + 1)/2 - rd = (T - rd + 1)/2. \tag{13}$$

Henceforce, the window increases by one in every RTT and the queue length begins to increase. So we have

**Theorem 3** *Under the conditions of Theorem 2, in congestion avoidance phase, the maximum queue length is* $T + 1$ *and the minimum queue length is* $(T - rd + 1)/2$.

In order to avoid link idling, we should have $(T - rd + 1)/2 \geq 0$, $T \geq rd - 1$. On the other hand, if $minQ$ is positive, the router may have an unnecessarily large queue and cause long queueing delay. Therefore, the best choice of threshold should satisfy

$$(T - rd + 1)/2 = 0, \tag{14}$$

or

$$T = rd - 1. \tag{15}$$

**Theorem 4** *In a path with only one connection, the optimal threshold that achieves full link utilization while keeping queueing delay minimal in congestion avoidance phase is* $rd - 1$. *If the threshold is smaller than this value, the link will be under-utilized. If the threshold is greater than this value, the link can be full utilized, but packets will suffer an unnecessarily large queueing delay.*

## 4 Queue Dynamics for Multiple Connections

When there are multiple connections, we can follow the argument in the previous section for each connection and get the following estimate.

**Theorem 5** *Suppose there are* $m$ *connections sharing the same bottleneck link, the* $i$*-th connection has a fixed round trip time* $r_i$*, and its average share of the bottleneck bandwidth is* $d_i$*, the window size at time* $t$ *is* $w_i(t)$*, the propagation time from the source to the bottleneck link is* $t_p^i$*, also suppose the bottleneck link has been busy for at least* $\max(r_i)$ *time, then the queue length at the congested router is*

$$Q(t) = \sum_{i=1}^{m} \left( w_i(t - t_p^i) - r_i d_i \right). \tag{16}$$

The difficulty to use this theorem is to find $d_i$. Misra [13] claims that a connection's share of bandwidth is inversely proportional to its RTT, but our simulation results do not support this claim. However, the connection with the shortest RTT has the largest share. If one connection has an RTT significantly shorter than others, the changes of aggregate window size is dominated by that connection, and the queue dynamics is similar to the single connection case. So in this section, we only study the case where all connections have comparable RTTs.

### 4.1 Slow Start Phase

Multiple connections are unlikely to start at the same time. If some connections are in slow start phase and others are in congestion avoidance phase, the queue increase is not as fast as all connections are in slow start phase. If all connections start simultaneously, we can consider them as one aggregate connection. The aggregate window doubles in an RTT. The argument for one connection still holds, so the buffer size requirement is $2T + rd + 1$.

### 4.2 Congestion Avoidance Phase

In congestion avoidance phase, the window size of one connection increases roughly by one in an RTT, so the aggregate window for multiple connections increases by a variable between 1 and $m$, depending on the distributions of RTTs and window sizes. Similar argument as in the previous section finds the maximum and minimum queue length to be

$$maxQ = T + X, \qquad minQ = (T + X - rd)/2. \tag{17}$$

where $X$ is a variable between 1 and $m$. So we have

**Theorem 6** *In a path with $m$ sources, the maximum queue length in congestion avoidance is between $T + 1$ and $T + m$, and the minimum queue length is between $(T + 1 - rd)/2$ and $(T + m - rd)/2$.*

Considering all possible values of $X$, the best choice of threshold that can avoid link idling and excessive delay should satisfy

$$minQ = (T - rd + 1)/2 = 0, \tag{18}$$

or

$$T = rd - 1. \tag{19}$$

**Theorem 7** *In a path with $m$ connections, the optimal threshold that achieves full link utilization while keeping queueing delay minimal in congestion avoidance phase is between $rd - m$ and $rd - 1$. If the threshold is smaller than $rd - m$, then link will be under-utilized. If the threshold is greater than $rd - 1$, the link will be full utilized, but packets will suffer an unnecessarily large queueing delay.*

## 5 Simulation Results

In order to verify our analysis, a set of simulations are performed with the $ns$ simulator [14]. The basic simulation model is shown in Figure 2.

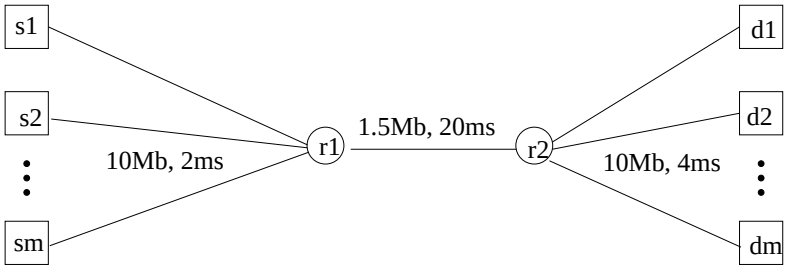


Figure 2: Simulation model.

The fixed round trip time is 59.21ms. Changing the propagation delay between $r_1$ and $r_2$ to 40ms gives an RTT of 99.21 ms. Changing the propagation delays between the sources and $r_1$ can give us configurations of connections with

different RTTs. An FTP application is attached to each source. Reno TCP and ECN are used for flow control. The data packet size is 1000 bytes and the acknowledgment packet size is 40 bytes. The simulations run for 50 seconds. Figure 3 shows the maximum queue length for different number of sources and different RTT values. The measured maximum queue lengths are shown with "*", "$\times$" and "$\circ$", the corresponding analytical estimates from Theorem 2, 3 or 6 are drawn with solid, dashed and dotted lines. When $T$ is small, the buffer size requirement is an upper bound, when $T \geq rd$, the estimate is accurate.

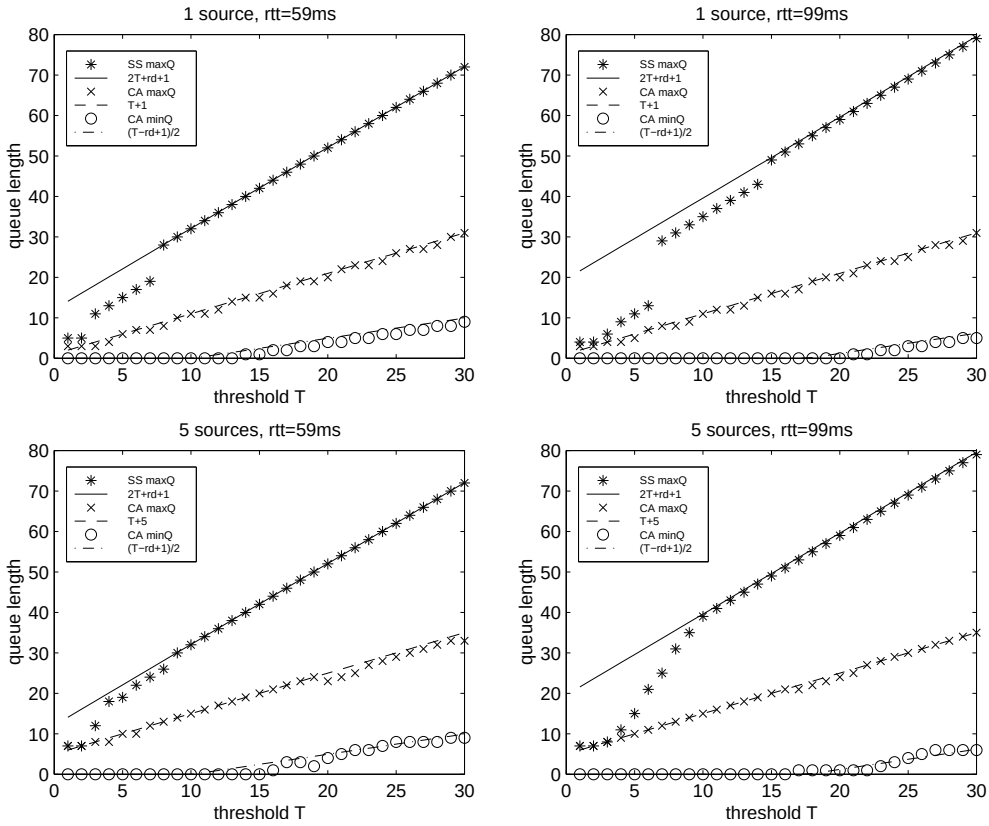


Figure 3: Maximum queue lengths for different number of sources and RTTs

Figure 4 shows the queue length graph of a small threshold ($T = 7$) and a large threshold ($T = 14$) for the configuration with $rd = 10.10$. Small threshold leads to low link utilization, while large threshold results in excessive queueing delay.

Figure 5 shows the link efficiency calculated from the number of packets successfully sent and acknowledged in 50s. The vertical lines in the figures show the optimal threshold range in Theorem 4 and 7. These results confirm the conclusions.

# 6 Refinements and Discussion

In section 2, we discussed the two techniques in congestion detection — average queue length and RED. Mathematical analysis of average queue length and RED is difficult [13]. In our simulation, we experiment with average queue length with different weights. The results indicate that actual queue length has better control over buffer overflow and underflow, for all choices of threshold, actual queue length needs smaller buffer than average queue length. When the threshold is small, average queue length has better link utilization; but when the threshold is not too small (about half of $rd$), actual queue length has better link utilization.

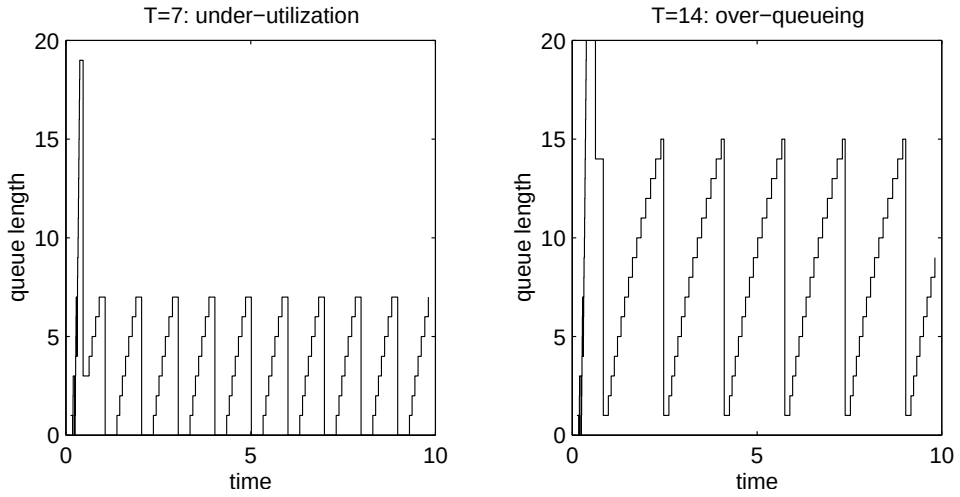


Figure 4: Queue length for small and large thresholds

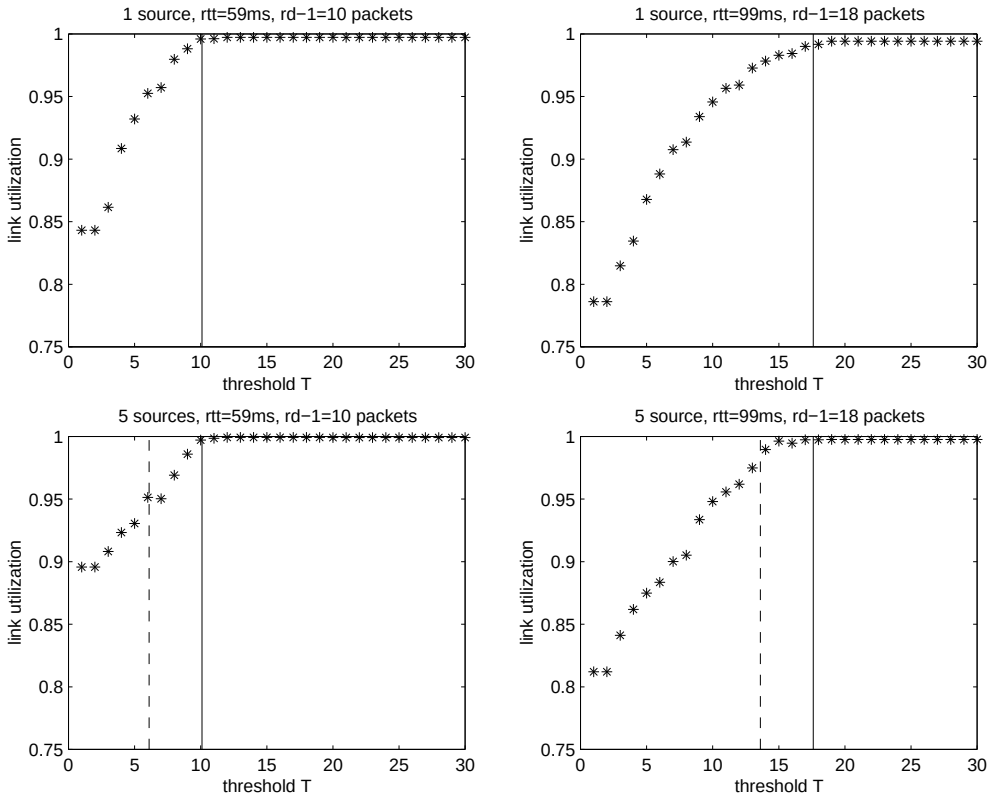


Figure 5: Link efficiency for different choices of threshold.

One important observation from our analysis is the global synchronization of congestion windows. When RED is not performed, no matter how the connections start, global synchronization will happen. Our simulation shows that RED is very effective in desynchronizing the windows and thus significantly reduces the average and variation of the queue length. How to choose the RED parameters to minimize the buffer requirement and transfer delay will be left as a

topic for further study.

# 7 Conclusion

In this paper, we study the buffer size and threshold requirement for zero loss flow control with ECN. The analysis shows that the buffer requirement is $2T + rd + 1$, where $T$ is the threshold, $r$ is the fixed round trip time, and $d$ is the data rate of the bottleneck link. The optimal threshold for ECN is $rd - 1$, this threshold can achieve full link utilization while keeping the queue delay minimal.

# References

[1] V. Jacobson, "Congestion avoidance and control", Proc. ACM SIGCOMM'88, pp. 314-329.

[2] W. Stevens, "TCP slow start, congestion avoidance, fast retransmit, and fast recovery algorithms", *RFC 2001*, January 1997.

[3] S. Floyd, "Connections with multiple congestion gateways in packet-switched networks: part 1: one-way traffic", *Computer Communication Review*, 21(5), October 1991.

[4] T. V. Lakshman, U. Madhow, "Performance analysis of window-based flow control using TCP/IP: effect of high bandwidth-delay products and random loss", *IFIP Transactions C: Communication Systems*, C-26, 1994, p.135-149.

[5] M. Mathis, J. Semke, J. Mahdavi, T. Ott, "The macroscopic behavior of the TCP congestion avoidance algorithm", *Computer Communication Review*, volume 27, number3, July 1997.

[6] T. Ott, J. Kemperman, and M. Mathis, "The stationary behavior of ideal TCP congestion avoidance", ftp://ftp.bellcore.com/pub/tjo/TCPwindow.ps, August 1996.

[7] J. Padhye, V. Firoiu, D. Towsley, J. Kurose, "Modeling TCP throughput: a simple model and its empirical validation", *Computer Communication Review*, 28(4), 1998, p. 303-314.

[8] K. Ramakrishnan and S. Floyd, "A proposal to add Explicit Congestion Notification (ECN) to IP", *RFC 2481*, January 1999.

[9] S. Floyd, "TCP and explicit congestion notification", *ACM Computer Communication Review*, V. 24 N. 5, October 1994, p. 10-23.

[10] S. Floyd and K. K. Ramakrishnam, "Explicit Congestion Notification — ECN birds of a feather session", http://www-nrg.ee.lbl.gov/floyd/ecn/kk-ecn-mar98.pdf

[11] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Transactions on Networking*, Vol. 1, No. 4, August 1993, pp. 397-413.

[12] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang and W. Weiss, "An architecture for differentiated services,", *RFC 2475*, December 1998.

[13] A. Mistra, T. Ott, J. Baras, "The window distribution of multiple TCPs with random loss queues", Globecom'99, Rio de Janeiro, Brazil, December, 1999

[14] UCB/LBNL/VINT Network Simulator - ns (version 2), http://www-mash.CS.Berkeley.EDU/ns/.