Analysis of Binomial Congestion Control *

Y. Richard Yang, Simon S. Lam
Department of Computer Sciences
The University of Texas at Austin
Austin, TX 78712-1188
{*yangyang, lam*}*@cs.utexas.edu*

TR-2000-14
June 8, 2000

1 Introduction

Binomial congestion control was proposed by Bansal and Balakrishnan in [2]. However, the sending rate derivation in [2] is greatly simplified and does not consider the effect of timeouts. Further, even though the authors use $\alpha = 1$ and $\beta = 0.6$ for TCP-friendliness in their experiments; this selection is not justified by their analysis. On the contrary, according to the authors, for $\alpha = 1$, they should select β such that $\frac{\alpha}{\beta} = \frac{1}{0.5}$, therefore, $\beta = 0.5$.

The motivation of this paper is to analyze the sending rate of binomial congestion window adjustment policy, considering both tripli-duplicate loss indications and timeout loss indications. We also consider the selection of α and β for IIAD and SQRT congestion control strategies [2] to be TCP-friendly. This paper suggests that the authors of Binomial should test their protocol under higher loss scenarios.

The balance of this paper is as follows. In Section 2, we describe the Binomial congestion control and state the analysis assumptions. The detail of the derivations is put in the Appendix. In Section 3, we use the sending rate formula to derive conditions under which a Binomial flow is TCP-friendly.

*Research sponsored in part by National Science Foundation grant No. ANI-9977267 and grant no. ANI-9506048. Experiments were performed on equipment procured with NSF grant no. CDA-9624082.

2 Model and Analysis Assumptions

Formally, the Binomial window adjustment policy is

$$\begin{cases} w_{t+R} & \leftarrow & w_t + \alpha/w_t^k & \text{if no loss} \\ w_{t+\delta t} & \leftarrow & w_t - \beta w_t^l & \text{when loss} \end{cases} \tag{1}$$

We can see that TCP is a special case when $k = 0$, $l = 1$. In this paper, we analysis the two cases considered by the authors: when $k = 1$, $l = 0$, which is called IIAD (inverse-increase/additive decrease) and $k = l = 0.5$, which is called SQRT (square-root).

Window adjustment policy, however, is only one component of a complete congestion control protocol. Other mechanisms such as congestion detection and round-trip time estimation are needed to make a complete protocol. Since TCP congestion control has been studied extensively for many years, Binomial adopts these other mechanisms from TCP Reno [5, 6, 8, 1]. In the next subsection, we give a brief description of the Binomial congestion window adjustment algorithm. All other algorithms are the same as those of TCP Reno.

2.1 Congestion window adjustment

A Binomial session begins in the *slowstart* state. In this state, the congestion window size is doubled for every window of packets acknowledged. Upon the first congestion indication, the congestion window size is cut in half and the session enters the *congestion avoidance* state. In this state, the congestion window size is increased by α/W^k in each round-trip time, where W is the current congestion window size. Notice that in this analysis we assume that the receiver returns one new ACK for each received data packet. It is straightforward to extend the analysis to consider delayed ACK. Binomial reduces the window size when congestion is detected. Same as TCP Reno, Binomial detects congestion by two events: *triple-duplicate ACK* and *timeout*. If congestion is detected by a triple-duplicate ACK, Binomial changes the window size to $W - \beta W^l$. If the congestion indication is a timeout, the window size is set to 1.

2.2 Modeling assumptions

The assumptions and simplifications made in this analysis are summarized below.

- We assume that the sender always has data to send (i.e., a saturated sender). The receiver always advertises a large enough receiver window size such that the send window size is determined by the Binomial congestion window size.

- The sending rate is a random process. We have limited our efforts to modeling the mean value of the sending rate. An interesting future topic will be to study the variance of the sending rate which is beyond the scope of this paper.

- We focus on Binomial's congestion avoidance mechanisms. The impact of slowstart has been ignored.

- We model Binomial's congestion avoidance behavior in terms of rounds. A round starts with the back-to-back transmission of W packets, where W is the current window size. Once all packets falling within the congestion window have been sent in this back-to-back manner, no more packet is sent until the first ACK is received for one of the W packets. This ACK reception marks the end of the current round and the beginning of the next round. In this model, the duration of a round is equal to the round-trip time and is assumed to be independent of the window size. Also, it is assumed that the time needed to send all of the packets in a window is smaller than the round-trip time.

- We assume that losses in different rounds are independent. When a packet in a round is lost, however, we assume all packets following it in the same round are also lost. Therefore, p is defined to be the probability that a packet is lost, given that it is either the first packet in its round or the preceding packet in its round is not lost [7].

- To void having too many parameters, we assume that the receiver returns one new ACK for each received data packet, i.e., no delayed ACK. To model the effect of delayed ACK, we can simply replace all α with α/b, where α is the increasing parameter, and b is the number of data packets before an ACK is sent.

- To derive an analytic result, sometimes in the analysis we assume $E[W^t] \approx E[W]^t$, where W is the window size and $t \in (0, \infty)$.

3 TCP-friendly Binomial Congestion Control

As derived in Appendix, the sending rate of both IIAD and SQRT can be expressed as

$$T_{Binomial}(\alpha, \beta, p, R, T_0) \approx \frac{1}{R\sqrt{\frac{\beta}{\alpha}p} + T_0 \min\left(1, 3\sqrt{\frac{\beta}{\alpha}p}\right)p(1+32p^2)} \qquad (2)$$

where p is the loss rate, R the mean round-trip time, and T_0 the timeout. We should emphasize that to derive (2), in some cases we have assumed p is small. For detail, refer to the Appendix.

To be TCP-friendly, we need to match $T_{Binomial}(\alpha, \beta, p, R, T_0)$ to that of TCP sending rate formula, which is

$$T_{TCP}(p, R, T_0) \approx \frac{1}{R\sqrt{\frac{2}{3}p} + T_0 \min\left(1, 3\sqrt{\frac{3}{8}p}\right) p(1 + 32p^2)} \tag{3}$$

Under low loss scenario, the first terms in the denominators of (2) and (3) dominates, and we have the expression:

$$\frac{\beta}{\alpha} = \frac{2}{3} \tag{4}$$

For example, when the Binomial congestion control uses $\alpha = 1$, we select $\beta = 0.66$ so that the control is TCP-friendly.

To consider the sensitivity of the TCP-friendliness on the β parameters, we define

$$\begin{aligned} F(\alpha, \beta) &= \frac{\frac{1}{R\sqrt{\frac{2}{3}p}}}{\frac{1}{R\sqrt{\frac{\beta}{\alpha}p}}} && (5) \\ &= \sqrt{\frac{3\beta}{2\alpha}} && (6) \end{aligned}$$

Under small loss rate p, F is the relative throughput of a IIAD/SQRT flow and a TCP flow. Figure 1 plots F as a function of β when $\alpha = 1$. Compare Figure 1 with the experimental results in Figure 16 of [2], we find that the two figures are very similar. This can be considered a validation of (2).

However, it is important to point out that F is valid only when loss rate p is small. When loss rate is high, we should use the complete sending rate formula to derive the TCP-friendly α and β, using the methods as in [**?**]. It also suggests that the authors of Binomial should evaluate Binomial under high loss scenarios.