Extended Analysis of Binary Congestion Control

Sergey Gorinsky Harrick Vin
Technical Report TR2002-28
Department of Computer Sciences
The University of Texas at Austin
Taylor Hall 2.124, Austin, TX 78712, USA
{gorinsky,vin}@cs.utexas.edu

January 16, 2002

ABSTRACT

Congestion control in the Internet relies on binary adjustment algorithms. For example, Transmission Control Protocol (TCP) in its congestion avoidance mode behaves similarly to Additive-Increase Multiplicative-Decrease (AIMD) algorithm. The classical analysis by Chiu and Jain recommends AIMD based on the assertion that among stable linear algorithms, AIMD ensures the quickest convergence to fair states. We demonstrate incorrectness of this assertion. For an asynchronous version of Chiu-Jain model, we show that AIMD is sensitive to initial conditions and has multiple unfair attractors. Our findings question the appropriateness of AIMD for binary congestion control. We attribute some of our observations to unrealistic features of Chiu-Jain model and argue that binary adjustment algorithms should be analyzed in a more realistic model.

1. INTRODUCTION

The Internet serves a multitude of users that spread all over the globe, compete for numerous network resources, and have changing communication demands. In such a complex system, it is arduous to provide every user with up-to-date information about its fair and efficient load on the network. Instead, congestion control in the Internet relies on *binary adjustment algorithms*: a user adjusts its load in response to binary signals that indicate whether the user must decrease or can increase the load. For example, Transmission Control Protocol (TCP) exercises binary congestion control – the TCP sender steps up its transmission after receiving a new acknowledgment; the sender reduces its load upon a retransmission timeout or after receiving three duplicate acknowledgments [1, 5]. Until the first indication of congestion, each TCP connection raises its load in a manner resembling the Multiplicative-Increase (MI) algorithm [3]. This reliance on MI is supposed to enable quick convergence to efficient states. Once efficiency is achieved, the TCP connection switches to the congestion avoidance mode and adjusts the load similarly to Additive-Increase Multiplicative-Decrease (AIMD) algorithm [3]. The choice of AIMD is supposed to provide stability, i.e., convergence to fair efficient states.

To our knowledge, the only theoretical justification for favoring AIMD appears in the classical work by Chiu and Jain [3]. According to their analysis of linear adjustment algorithms, AIMD provides the quickest convergence to fairness. In this paper, we review Chiu-Jain analysis and derive several surprising results. In particular, we refute the assertion that AIMD guarantees the fastest convergence to fair states. We also examine a model that allows different users to have different round-trip times. In this asynchronous version of Chiu-Jain model, AIMD behaves chaotically: it is sensitive to initial conditions and has multiple unfair attractors. We attribute some of our observations to unrealistic features of Chiu-Jain model and argue that the problem of choosing an appropriate algorithm for binary adjustments should be examined in a more realistic model.

The rest of our paper is structured as follows. Section 2 presents Chiu-Jain analysis. Section 3 examines the issue of convergence to fair states. Section 4 extends the analysis to the asynchronous version of Chiu-Jain model. Finally, Section 5 gives a summary of our conclusions.

2. CHIU-JAIN MODEL AND ANALYSIS

In [3], Chiu and Jain use a simple model to analyze binary congestion control. They represent the network as a single resource shared by cooperative users. The model assumes that all users have the *same* round-trip time and adjust their loads *simultaneously*. Consequently, the model employs a discrete timescale where every instant t corresponds to the moment when each user i adjusts its load to $x_i(t)$. The network provides the users with a binary feedback $y(t)$ which indicates whether the total load $X(t-1)$ after the previous adjustment exceeds an optimal value X_{goal}:

$$y(t) = \begin{cases} 1 & \text{if } X(t-1) > X_{goal}, \\ 0 & \text{if } X(t-1) \leq X_{goal} \end{cases} \tag{1}$$

where $X(t)$ is the combined load of all n users at time t:

$$X(t) = \sum_{i=1}^{n} x_i(t). \tag{2}$$

Note that the model assumes *uniform* feedback – all the users receive the same bit $y(t)$. The users have no access to other external information including n, X_{goal}, or $X(t-1)$.

Chiu and Jain perform a static analysis for the following class of linear adjustment algorithms:

$$\forall i \quad x_i(t) = \begin{cases} a_I + b_I x_i(t-1) & \text{if } y(t) = 0, \\ a_D + b_D x_i(t-1) & \text{if } y(t) = 1 \end{cases} \tag{3}$$

where a_I, b_I, a_D, and b_D are real constants.

The criteria for selecting an appropriate algorithm include its *stability*: for any initial loads of the users, load $x_i(t)$ of