Chapter 6

TCP Symbiosis: Bio-Inspired Congestion Control Mechanism for TCP

Go Hasegawa
Osaka University, Japan

Masayuki Murata
Osaka University, Japan

ABSTRACT

In this chapter, we introduce a robust, self-adaptive and scalable congestion control mechanism for TCP. We change the window size of a TCP connection according to the information of the physical and available bandwidths of the end-to-end network path. The bandwidth information is obtained by an inline network measurement technique. We also borrowed algorithms from biophysics to update the window size: the logistic growth model and the Lotka-Volterra competition model. The greatest advantage of using these models is that we can refer previous discussions and results for various characteristics of the mathematical models, including scalability, convergence, fairness and stability in these models. Through mathematical analysis and simulation experiments, we compare the proposed mechanism with traditional TCP Reno, HighSpeed TCP, Scalable TCP and FAST TCP, and exhibit its effectiveness in terms of scalability to the network bandwidth and delay, convergence time, fairness among competing connections, and stability.

INTRODUCTION AND RESEARCH BACKGROUND

Transmission Control Protocol (TCP) (Stevens, 1994) is the de facto standard transport layer protocol of the current Internet first designed in the 1970s: the first Request for Comments (RFC) on TCP was released in 1981 (Postel). TCP has also been frequently modified and enhanced to accommodate the development of the Internet (Jacobsen et al., 1992; Allman et al., 2001; Blanton et al., 2003).

TCP has various functions to realize reliable and efficient data transmission on the network. The congestion control mechanism (Stevens, 1994) is one of the most important. Its main purpose is to avoid and resolve network congestion and to distribute network bandwidth fairly.
TCP Symbiosis

among competing connections. To do that, TCP employs a window-based congestion control mechanism that adjusts data transmission speed by changing congestion window size. A congestion window indicates the maximum amount of data that can be sent out on a connection without being acknowledged.

TCP Reno is the most popular version of TCP in current operating systems. Its window size control algorithm allows a TCP sender to continue to additively increase its congestion window size until it detects a packet loss (or losses), decreasing it multiplicatively when a packet loss occurs. This is called an Additive Increase Multiplicative Decrease (AIMD) policy. In (Chiu & Jain, 1989), the authors argue that an AIMD policy is suitable for efficient and fair bandwidth usage in a distributed environment, if congestion indication signals are simultaneously distributed to all connections.

With increases in the heterogeneity and the complexity of the Internet, many problems have emerged in TCP Reno’s congestion control mechanism (Shenker et al, 1990; Hoe, 1996; Guo & Matta, 2001; Fu et al, 2003; Floyd, 2003 for example). The primary reasons for these problems are that the congestion signals are only indicated by packet loss and that TCP Reno uses fixed AIMD parameter values to increase and decrease window size, whereas the window size should be changed according to the network environment. For example, it is well-known problem that the throughput of TCP connections decreases when it traverses wireless links, since TCP cannot distinguish congestion-oriented packet loss caused by network congestion, and wireless-oriented packet loss caused by link loss and/or handoff. Another problem is the low throughput of TCP connections in large bandwidth and long delay networks. In (Floyd, 2003), the authors argued that a TCP Reno connection cannot fully utilize the link bandwidth of such networks, since the increasing parameter, which is one packet per a Round Trip Time (RTT), is too small and the decreasing parameter, which halves the window size when a packet loss occurs, is too large for networks with a large bandwidth-delay product. Although many solutions have been proposed for their problems (Floyd, 2003; Kelly, 2003; Jin et al, 2004), most of them inherit the fundamental congestion control mechanism of TCP Reno: the AIMD mechanism triggered by the detection of packet losses in the network. The congestion control mechanism improves the throughput by adjusting the increasing and decreasing parameters statically and/or dynamically. However, most previous studies have focused on changing the AIMD parameters to accommodate particular network environments. Since these methods employ ad hoc modifications for a certain network situation, their performance when applied to other network environments is unclear.

Because window size indicates the maximum amount of packets that TCP can transmit for one Round Trip Time (RTT), an adequate window size for a TCP connection is equal to the product of the available bandwidth and the round-trip propagation delay between the sender and receiver hosts. TCP Reno measures the RTTs of the network path between sender and receiver hosts by checking the departure times of the data packets and the arrival times of the corresponding ACK packets. However, TCP Reno does not have an effective mechanism to recognize the available bandwidth. This is an explanation of the fundamental problem: TCP Reno cannot adjust window size to an adequate value under various network environments. In a sense, traditional TCP Reno can be considered to be a tool that measures available bandwidth because of its ability to adjust the congestion window size to achieve a transmission rate appropriate to the available bandwidth. However, it is ineffective because it only increases the window size until packet loss occurs. In other words, TCP Reno induces packet loss in order to obtain information about the available bandwidth (delay product) of the network. That is, even when the congestion control mechanism of TCP works perfectly,
Related Content

Nature-Inspired Knowledge Mining Algorithms for Emergent Behaviour Discovery in Economic Models
www.igi-global.com/chapter/nature-inspired-knowledge-mining-algorithms/21127?camid=4v1a

On Quasi Discrete Topological Spaces in Information Systems
www.igi-global.com/article/quasi-discrete-topological-spaces-information/74335?camid=4v1a

A Multiscale Computational Model of Chemotactic Axon Guidance
www.igi-global.com/chapter/multiscale-computational-model-chemotactic-axon/52336?camid=4v1a

Development of Highly Sensitive Compact Chemical Sensor System Employing a Microcantilever Array and a Thermal Preconcentrator
www.igi-global.com/chapter/development-highly-sensitive-compact-chemical/71925?camid=4v1a