Design and Evaluation of Congestion Control Algorithms in the Future Internet

Narayanan Venkitaraman Tae-eun Kim Kang-Won Lee
Songwu Lu Vaduvur Bharghavan*

Abstract

Linear Increase/Multiplicative Decrease (LIMD) has typically been the congestion control paradigm of choice in the traditional Internet with tail-drop routers. In recent years, routers with fair packet dropping policies (e.g. RED) are increasingly being deployed in the Internet. In the context of a network with fair packet dropping routers, we challenge the popular notion that Linear Increase/Linear Decrease (LILD) is an unsuitable paradigm for congestion control, and present two key results:

(a) LILD is both *efficient* and *fair* in the target environment. In fact, LILD outperforms LIMD in most respects.

(b) By maintaining the *history* of transmission rates and loss profiles, congestion control algorithms can effectively distinguish between congestion-induced and non-congestion-induced loss.

We present the LILD/H algorithm that combines the above features. LILD/H reacts gently to random/channel probe loss, and aggressively to congestion loss, thereby adapting effectively to the dynamics of the network. Consequently, we believe that LILD/H is a viable congestion control algorithm at the end host or the edge router in the future Internet.

1 Introduction

Congestion control algorithms that are deployed in the Internet today typically use the linear increase/multiplicative decrease (LIMD) paradigm for adapting the transmission rate of a connection to reflect the available bandwidth. Briefly, LIMD periodically adapts the sending rate of a connection by gently increasing the rate upon observing no packet losses in order to probe for additional bandwidth (i.e. $r \leftarrow r + \alpha$), and aggressively decreasing the sending rate upon observing packet losses in order to alleviate congestion (i.e. $r \leftarrow r(1-\beta)$). In essence, LIMD assumes that all packet losses are due to congestion, that all the contending connections traversing through a bottleneck link experience packet drops during congestion, and that aggressively throttling down the transmission rate can quickly alleviate congestion. LIMD is known to be robust and asymptotically provide fairness among competing connections. However, it has several drawbacks, all of which are caused by the fact that it throttles the sending rate by a factor that is typically both aggressive and independent of the number of packets dropped. The known problems include the following: (a) since β *must* be large (e.g. 0.5 in TCP) to quickly alleviate congestion, the average sending rate is suboptimal (e.g. 75% in TCP) even when the delay-bandwidth product of the connection is invariant, (b) LIMD reacts wrongly to losses that are not caused by congestion, and (c) LIMD becomes highly inefficient if there are packet losses in successive periods of rate adaptation. In spite of all the known problems of LIMD, it is still the predominant paradigm for congestion control because the alternatives, such as linear increase/linear decrease (LILD) have been shown not to converge asymptotically to fairness for an arbitrary start state, given the tail-drop packet dropping policies of typical Internet routers to date [1].

In recent years, researchers have perceived the need to introduce router-level mechanisms for ensuring *fair packet dropping* in the router during congestion, in order to punish non-adaptive senders that fail to react to congestion by throttling down their sending rates. With fair packet dropping, the number of packets of a connection dropped during congestion at a router will be proportional to its sending rate in the long term. Popular examples of such mechanisms include random early detection (RED) and its variants (FRED, WRED). In this paper, we assume that fair packet dropping will be implemented in the future Internet (not an unreasonable assumption, given that RED is already being deployed very successfully in the Internet). With this assumption in place, we show that LILD is both fair and efficient, hence a viable paradigm for end-to-end congestion control in the future Internet.

2 Convergence of LILD with fair packet dropping

There are two ways of defining the linear decrease of LILD: (a) decrease rate by a constant upon packet loss in the last measurement period, called *epoch*, (i.e. $r \leftarrow r - \beta$) or (b) decrease rate in proportion to the number of packet losses observed during the last measurement period (i.e. $r \leftarrow r - \beta f$, where f is the loss ratio). We will adopt the latter definition, and show that this leads to fair rate allocation with fair packet dropping.

Figure 1 shows the trajectory of rate allocation in LILD with fair packet dropping. In the figure, X_0 represents the initial rate allocations of flow 1 and flow 2. Both flows increase their sending rates linearly until the system reaches $X_1 = (x_1, x_2)$. At this point, *the expected packet loss that each flow will observe is in proportion to its sending rate if all routers implement fair packet dropping.* Thus, flow 1 and flow 2 will see an expected loss of $(\frac{x_1}{x_1+x_2}) \times n$ packets and $(\frac{x_2}{x_1+x_2}) \times n$ packets, respectively, where n is the total packet loss. This brings the system to X_2 which is closer to the fairness line. By repeating this cycle, we see that the system converges to the optimal point, where rate allocation is both

*The authors are with the Coordinated Science Laboratory, University of Illinois, Urbana, IL, 61801.

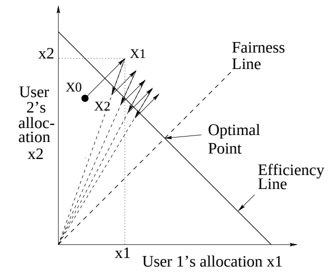

Figure 1: Convergence of LILD with fair packet dropping

efficient and *fair*. (Note that with definition (a), the system will oscillate along the X_0 and X_1, and do not converge to the optimal point.) It turns out the LILD overcomes a number of problems associated with LIMD congestion control. Specifically, LILD is less susceptible to random packet loss, leads to significantly higher link utilization, does not cause large variations in transmission rate or buffer utilization, and is less susceptible to loss synchronization. However, LILD is also less sensitive to the onset of sudden congestion because it does not react to packet drops as aggressively as LIMD, and takes longer to converge to the fairness point. These drawbacks of LILD can be overcome by the maintenance of congestion history, as described below.

3 LILD with history (LILD/H)

Packet losses may generally occur because of three reasons: (a) congestion losses, (b) probe losses, and (c) random losses. Irrespective of the specific paradigm for congestion control, there is a fundamental conflict in achieving both efficiency and fairness so long as the congestion control algorithm does not distinguish between random/probe loss and congestion loss. We observed that both LIMD and LILD suffer from the consequences of not distinguishing the cause of packet loss and reacting appropriately: the utilization of LIMD is suboptimal in steady state, whereas the responsiveness of LILD is poor upon sudden congestion as a result of their design choices. What we need is to react gently to random and probe loss, but react aggressively to congestion loss. This can be done fairly effectively by maintaining the profile of packet losses and sending rates in the recent past.

In LILD/H, a conscious effort is made to keep track of the progress of the transmission rate and loss in each epoch, and hence, keep track of the profile of the *effective transmission rate* $R(= r(1-f))$, i.e. the virtual transmission rate at which all the packets in the last epoch would have been received successfully at the receiver. LILD/H maintains three variables: the moving average of the effective transmission rate $\bar{R}$, the average deviation of the effective transmission rate $\bar{D}$, and the decrease variable d, which denotes the degree of estimated congestion. (The variable d is multiplied by 2 when there is packet losses in an epoch, and reset to 1 otherwise. Thus, at the end of k consecutive epochs with packet loss, $d = 2^k$.) At any time, LILD/H predicts that so long as the effective transmission rate in epoch i is within $\bar{R} - \eta\bar{D}$, packet losses were either random loss or channel probe loss (where η is typically set to 1.5), and decreases linearly according to the loss ratio. If the effective transmission rate falls below $\bar{R} - \eta\bar{D}$, then LILD/H charges all losses within the $\bar{R} - \eta\bar{D}$ bound to random/probe loss, and the remainder of the losses to the decrease in the network bandwidth. In this case, LILD/H decreases multiplicatively by d for the congestion-induced loss (specifically, $r \leftarrow \{\bar{R} - \eta\bar{D} - d(\bar{R} - \eta\bar{D} - R)\}$). LILD/H thus tries to preserve LILD's property of graceful variation of the transmission rate for random/probe loss, but at the same time, throttle the transmission rate quickly for the decrease in the available bandwidth for a connection due to reduction of network resources, or the advent of new connections.

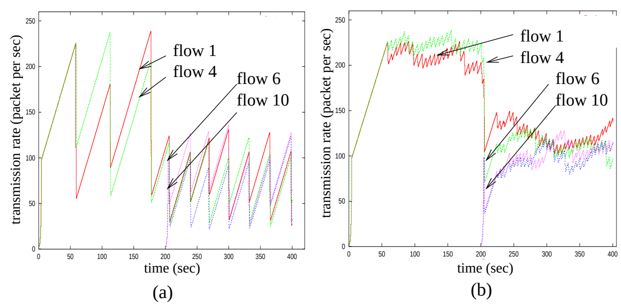

Figure 2: Transmission rate: (a) LIMD and (b) LILD/H

4 Performance Results

We performed a simulation of LIMD, LILD, LILD/H, and their variants. In this version of paper, we only show the performance of LIMD and LILD/H in a simple single-hop network topology to illustrate the advantage of LILD based congestion control with history information. There are ten flows sharing one bottleneck link. The capacity of the bottleneck link is 9 Mbps with 260 msec delay. For each flow, there are separate entrance and exit data links with 40 Mbps bandwidth and 20 msec delay.

For the result shown in Figure 2, five flows started first at time 0 sec. Then the other five flows started at time 200 sec. when the first set of flows have arrived at a steady state. Figure 2.(a) and 2.(b) show the transmission rate changes of LIMD and LILD/H, respectively. Due to space constraints, only four randomly selected flows (flow 1 and 4 started at time 0, and flow 6 and 10 started at time 200) are presented. As shown in the figure, LIMD exhibits a larger variation of transmission rates even when the network bandwidth is invariant. However, with LILD/H the variation of transmission rate is small as long as the network bandwidth does not change (0 – 200 sec). When the new flows start transmission (using slow start), both LIMD and LILD/H react quickly and decrease the transmission rate of the existing flows in order to make room for new flows. Essentially, we observe that LILD/H achieves a better utilization of network resources than LIMD, when there is no congestion. At the same time, LILD/H achieves approximately the same degree of responsiveness as LIMD when there occurs actual congestion.

Thus, we believe that LILD/H shows the most desirable features of LILD and LIMD, and is also able to effectively distinguish between random/probe loss, and losses due to the reduction of the available bandwidth.

References

[1] D.-M. Chiu and R. Jain. Analysis of the increase and decrease algorithms for congestion avoidance in computer networks. *Journal of Computer Networks and ISDN Systems*, 17(1), June 1989.