On Fair Window Control for TCP with ECN using Congestion Pricing

Ngo Dong Hai
Posts and Telecommunications Institute of Technology (PTIT), Vietnam
Vu Ngoc Phan
Vietnamese academy of science and technology

Abstract

This paper focuses on a TCP window-based flow control mechanism with Explicit Congestion Notification (ECN). We investigate the fundamental problem of achieving a fair window control for TCP, which cooperates with ECN. This is done by using feedback congestion pricing as a means of estimating the state of bottleneck router. The problem is solved by achieving network optimal performance, which maximize the total user utilities. We then look at the simulation of such scheme.

1. Introduction

Transmission Control Protocol (TCP) is the most widely used transport protocol in the Internet. The original TCP does not have any explicit feedback information for the congestion state from the network. TCP source hosts have to themselves adjust their transmission rate with respect to the timeouts and the receipt of the duplicate acknowledgments (ACKs).

To avoid delays and packet drops, ECN (Explicit Congestion Notification) has been introduced. However, as is widely recognized, TCP with ECN does not generally lead to a fair or efficient resource allocation among connections [3], [7].

The classical resource sharing principle for data networks is a max-min fairness criteria [10]. The problem of fair resource allocation to u sers can be solved by pricing scheme, as discussed in [10] Congestion control algorithm using pricing scheme with explicit feedback have been proposed by Kelly et al [6]. However, pricing scheme is mainly used for rate control algorithms, in which users adapt their transmission rates to balance the revenue they obtain from the network. There are not many arguments for preferring a window-based congestion control algorithm over a rate-based algorithm

This paper proposes a modified TCP window control algorithm, which is based on pricing. Our goal is to find a new ECN marking algorithm that solves the system problem of maximizing the aggregate utility of the users. The changes to TCP discussed in this paper all adhere to the underlying framework of the congestion avoidance component.

This paper is organized as follows. Session 2 briefly reviews the problem of using ECN in TCP window control algorithm. The network modeling and assumption are presented in session 3. In session 4, the used technique to implement fair window control for TCP with ECN are described. Presented in session 5 are the results of simulation. Conclusions are presented in session 6.

2. Background: TCP with ECN

The fast growing Internet has shown the need for efficient congestion control algorithms. TCP is a window-based congestion control mechanism. A window-based algorithm adjusts its sending rate based on congestion window size, which is the maximum number of sending packets that the source host has not yet received acknowledgements for.

TCP has three distinct phases: Slow Start, Congestion Avoidance and Retransmission/Recovery. Slow Start implements the initial probing for available resource in the network. Retransmission/Recovery includes the exponential backoff of the retransmit timer when a retransmitted packet is itself dropped. The basic component of TCP congestion control is Congestion Avoidance.

TCP is widely known as Additive Increase Multiplicative Decrease (AIMD). AIMD, the traditional congestion control algorithm of the Internet, operates within that scope: it increases additively the rate of the users by increasing the congestion window by one per round-trip time until the network reaches congestion. Then it decreases the users' sending rate by halving their window size multiplicatively using a decrease ratio. This algorithm allows users to probe the network to find out the maximum rate at which data can be sent without incurring packet drops.

TCP connection adjust its transmission rate by updating its window size based on the estimated congestion state of the network. One of the fundamental characteristics of TCP congestion control is that packet losses are used as indication of congestion. Whenever a packet loss has occurred, there are timeout or duplicate acknowledgments that the source host would experience The original TCP depends on only timeouts and duplicate acknowledgements (ACKs) as the feedback information However, this mechanism suffer from multiple packet losses and global synchronization [1].

To improve these behaviors of TCP under congestion, Random Early Drop (RED) [2] and Explicit Congestion Notification (ECN) [3] have introduced. RED prevents network from multiple packet losses and global synchronization by dropping packets randomly before the gateway experiences queue overflow. With ECN, the sources are explicitly notified of congestion occurrence. ECN enables routers to probabilistically mark a bit in the IP header, rather than drop the packet, to inform end hosts of pending congestion when the length of the queue exceeds a threshold. The destinations then correlatively mark a bit in the header of the ACK, which is sent back to the source host. Thanks to the explicit feedback from the

network, the TCP congestion control algorithm is transformed to a congestion avoidance algorithm. By this way, the performance of network has been improved. However, RED/ECN cannot solve the problem of unfair sharing of network resource [4].

TCP with ECN has been proposed by Floyd [3]. In [5], authors extract end-to-end congestion information by simply counting the ECN marks in the ACKs. The probability that a packet is marked is used to estimate the average queue length. In [4], the window size of the sources are adjust based on calculating the level of congestion, which is indicated by the ratio of market and unmarked ECN-ACKs. In [5] and [7], the users use the number of marked ECN messages during a round-trip time to extract queue length at the router. However, estimating queue length is impractical for large number of sources.

This paper analyzes and proposes a TCP congestion control algorithm based on fairness and pricing concepts of Kelly. However, in our scheme, the problem does not be decomposed into separate subproblems for the router and for the individual users. The goal of the network is to allocate resources to maximize the total utility over all users. Router is the best place to detect congestion in the network. So that, the router, not the users, takes part of solving fair resource sharing problem. The result is used to calculate the packet marking ratio for each individual connection. Source hosts do not be required to take any computation of pricing, but simple adjustment to window size based on ECN marks.

3. Network modeling and assumptions:

3.1 Network modeling

Consider a simple network consist of N source hosts. Each source host is connected to a corresponding destination through a single bottleneck router. The network is treated as a discrete-time system, where the duration of one slot corresponds to the round-trip time of TCP connections. Each source host adjusts the window size once per round-trip time.

3.2 Assumptions:

In this paper, the assumptions are:

- All TCP connection have an identical round-trip time, which is the Least Common Multiple (LCM) value of all round-trip time obtained when the network is not congested.
- Network updates its state every round-trip time.
- All source hosts and router use TCP and are capable of RED/ECN.
- Each source host always has packets to transmit.
- All packets are of equal length.
- The service discipline of the router is First Come First Serve (FCFS).
- All sources use identical utility function.

4. Problem formulation

4.1 Fair resource sharing

Let x_i (packet per second) be the flow rate of the *i-th* connection for *i=1,...,N*. When user *i* gain a throughput of x_i, it receives utility $U_i(x_i)$ and has to pay $C_i x_i$, where C_i is the cost per unit of throughput. $U_i(x_i)$ is assumed to be an increasing, strictly concave and continuously differentiable function of x_i over the range of $x_i \geq 0$.

Let μ (packet per second) be the processing rate of the router; w_i (packet) be the congestion window of source *i*, $w_i = x_i RTT$; min_{th}, max_{th} be the minimum and maximum queue length threshold of the router; α is proportional weight of each utility function.

The fair resource sharing scheme is the solution of the following optimal problem.

$$Maximize \qquad \sum_i (\alpha_i U_i(x_i) - C_i x_i) \quad (1)$$

Subject to

$$RTT \sum_i x_i \geq min_{th} + RTT\mu \quad (2)$$

$$RTT \sum_i x_i \leq max_{th} + RTT\mu \quad (3)$$

The constrains (2) and (3) guarantee that the aggregate flow do not exceed the processing capability of the router and packets do not have to wait in the queue for a long time. They also guarantee that there are always packets transmitted in the network.

While this optimization problem is mathematically tractable, its solution relies on knowledge of the sources' utility functions $U_i(x_i)$, which are typically not known to the system. F. Kelly [7] argues that resource should rather be shared in such a way to maximize an objective function representing the overall utility of the flows in the network. The overall utility is assume to be additive, meaning that it is $\sum_i U_i(x_i)$.

According to [9], the utility function of TCP Reno is

$$U_i(x_i) = \frac{\sqrt{2}}{d_i} arctan\left(\frac{d_i x_i}{\sqrt{2}}\right) \quad (4)$$

where d_i is backward delay of *i-th* connection.

The utility function of TCP Vegas is

$$U_i(x_i) = log(x_i) \quad (5)$$

In our model, (5) is taken to be the utility function for all connections. In [7], they have proved that the proportional fairness maximizes the overall utility of all flows with utility function is logarithmic. Proportional fairness means, that if all sources use identical utility function with identical weight, the source which uses the most resources in the network will get the lowest rate. By choosing the appropriate weights to each connection, we can get the weighted proportionally fair [7].

The above maximization problem has a strictly concave objective function and the maximization is performed over a compact set. Therefore, it has a unique solution, which is the optimal rate x_i^{opt} at which each connection should operate.

4.2 Algorithms

Suppose that the network is at the end of *(t-1)-th* time slot, the router's and the sources' algorithms are set up as follow.

Router's algorithm:

1. *Estimates the window size* w_i^{t-1} *of each source which had packets to transmit during (t-1) time slot. To simplify,* w_i^{t-1} *is treated as the number of packets that i-th source has transmitted during (t-1) time slot. The actual flow is* $x_i^{t-1} = w_i^{t-1}/RTT$.
2. *Solves the problem of maximization utilization of aggregated sources (1)-(3). The optimal solution* x_i^{opt} *is used to compute the t-th optimal window size of each source:* $w_i^{opt} = x_i^{opt}.RTT$.
3. *Marking algorithm: The packets from i-th source arriving at router during t-th time slot will be marked with probability* $p_i(t)$ *as*

$$p_i(t) = \begin{cases} \dfrac{w_i^{t-1}}{w_i^{opt}} & \text{if } w_i^{t-1} < \eta w_i^{opt} \\ 1 & \text{if } w_i^{t-1} \geq \eta w_i^{opt} \end{cases}$$

where $\eta \approx 1$ *is control coefficient.*

The optimal problem (1)-(3) results in a resource sharing among multiple connections in a possibly weighted fair manner. The result of the problem is that each user is shared a fair portion of the router's buffer. The per-connection marking ratio presents the matter that how much each connection has consumed its sharing resource.

Sources' algorithm:

The source host adjusts its window size based on the feedback information returned as a series of ECN mark packets. This is done by simply counting the ECN marks in the traversing packets.

At the beginning of *t-th* time slot, the source host carries out the following steps:

1. *Count the number of marked ACKs received during previous time slot.*
2. *Estimate the marking probability by computing the ratio of the number of marked ACKs to the number of all received ACKs*

$$e_i^{t-1} = \frac{NE_i}{NA_i}, \qquad (0 \leq e_i^{t-1} \leq 1).$$

where NA_i, *be the number of all received ACKs at i-th source host.*

NE_i *be the number of received ACKs at i-th source host with ECN bit set.*

3. *If* e_i^{t-1} *is close to 1, it implies that the i-th source host has consumed almost all of its allocated resource. Then, the window size should be reduced quickly. On the contrary, if* e_i^{t-1} *is close to 0, there is so much resource that the user hasn't used yet. So, the window size should be increased quickly. If* $0 < e_i^{t-1} < 1$, *the window size should be additively increased. The proposed window size adjustment is:*

$$w_i^t = \begin{cases} w_i^{t-1}(1 - \gamma \ln e_i^{t-1}) & \text{if } e_i^{t-1} < 1 \\ \kappa w_i^{t-1} & \text{if } e_i^{t-1} = 1 \end{cases}$$

where $\gamma > 0$ *and* $0 < \kappa < 1$ *are control coefficients.*

4. Simulation

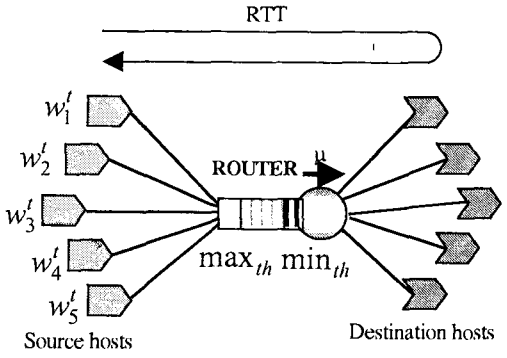

Figure 1. Analytic model.

In order to carry out a simulation model, we look at the Lagrangian for the problem (1)-(3). In session 3.2, we have assumed that each source host always has packets to transmit, so that the constrain (2) can be omitted.

The Lagrangian for the problem is:

$$L(x,z,u) = \sum_i (\alpha_i \log x_i - C_i) + u\left(max_{th} + RTT\mu - RTT\sum_i x_i - \sum_i z_i\right)$$

where $z_i > 0$ is slack variables and u is Lagrange multiplier. Then we have

$$\frac{\partial L}{\partial x_i} = \frac{\alpha_i}{x_i} - C_i - uRTT$$

which leads to the optimum of the problem:

$$x_i = \frac{\alpha_i}{C_i + uRTT}$$

In our simulation model, to simplify, it is assumed that all α_i (and all C_i) are same.

The aim of this simulation is to compare the proposed algorithm with the present ECN-TCP algorithm in term of the window size and the queue length at the router. The model has the following parameters (Figure 1). The processing speed of the router is 150 packet/ms. The round-trip time is 1 ms. The number of TCP connection is 5. The other parameters are $max_{th}=100$, $min_{th}=20$.

In the model, at the beginning of the transmission process, the sender transmits data according to slow start stage. When the first duplicative ACKs or timeout occur, it reduces its window size by 50% and transforms into the congestion avoidance stage. During this stage, it implements the window control according to equation (6). If it gets a heavy congestion marking (the marking ratio is close to 1), it reduces its congestion window by 20%. The new window size is computed after every round-trip time.

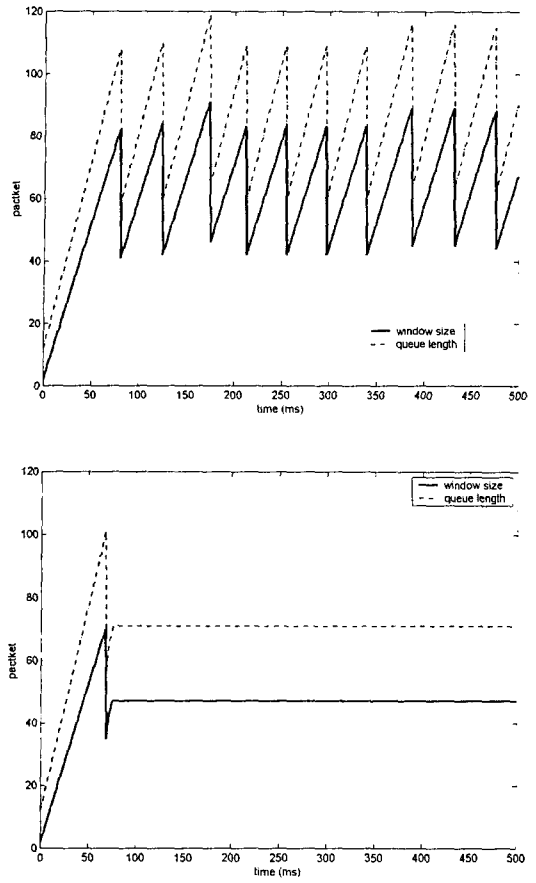

Figure 2. Window size, queue length: (a) ECN algorithm; (b) proposed algorithm.

Figure (2) illustrates the window size of source host, the queue length at the router for the ECN algorithm (2.a) and for the proposed algorithm (2.b). With assumption of identical initial window size, the change of each window size is the same. The result shows that the queue length converges to a level between min_{th} and max_{th}. The oscillations of the window size and the queue length are eliminated.

5. Conclusion

This paper proposes a TCP window congestion control mechanism using pricing with ECN. In our scheme, the senders do not have to extract the network information. They have just to compute the reasonable window size based on the fair sharing of the network resources. This scheme shows that the oscillations of the window size and the queue length are eliminated.

The algorithm was tested by means of Matlab simulation. The results obtained confirmed that the weighted fairness in resource allocation among multiple users was archieved.

As a future work, we should extend our analysis to more realistic networks with variation of round-trip time or multiple bottleneck routers.

References

[1] L. Zhang and D. Clark, "Oscillation behavior of network traffic: A case study similation", Internetworking: Research and Experience, vol.1, no.2, pp.101-112, 1990.

[2] S. Floyd and V. Jacobson, "Random Early Detection gateways for congestion avoidance", IEEE/ACM Trans. Netw., vol.1, no.4, pp.397-413, 1993.

[3] S. Floyd, "TCP and explicit congestion notification", ACM computer Communication Review, vol.24, no.5, pp.8-23, 1994.

[4] H. Choi and J. Lim, "On fair window control for TCP with ECN using congestion level", IEICE Trans. Commun., vol.E86-B, No.12, Dec 2003.

[5] R. Pletka and M. Waldvogel and S. Mannal, "PURPLE: Predictive active queue management utilizing congestion information", IBM research, Zurich Research Laboratory, 2003

[6] F. Kelly and A. Maullo and D. Tan, "Rate control in communication networks: shadow prices, proportional fairness and stability", Journal of the Operation Research Society, vol.49, pp.237-252, 1998.

[7] H. Byun and J. Lim, "On window-baesd congestion control with Explicit congestion notification", IEICE Trans. Commun., vol.E86-B, no.1, Jan 2003.

[8] L. Zhu and G. Cheng and N. Ansari, "Local stability of random exponential marking", IEE proc. commun., vol.150, no.5, Oct 2003.

[9] D. Luong and J. Biro, "Bandwidth sharing scheme of end-to-end congestion control protocol", Dept. of Telecom. And Telematics, Budapest University of Technology and Economics, May 2002.

[10] N. D. Hai, "Optimization problems in telecommunication networks: A classification sutdy", Journal of Computer science and Cybernetics, vol.19, no.3, pp.281-289, 2003.