

Utility Function of TCP and Its Application to the Design of Minimum Rate Guaranteed Control Law

Lei Ye¹, Zhijun Wang¹, Hao Che², Henry B.C. Chan¹, Indradip Ghosh², and Constantino M. Lagoa³

¹Department of Computing, The Hong Kong Polytechnic University, Hong Kong

²Department of Computer Science and Engineering, University of Texas at Arlington, TX, USA

³Department of Electrical Engineering, The Pennsylvania State University, PA, USA

Emails: {cseyei, cszjwang, cshchan}@comp.polyu.edu.hk, {hche, ighosh}@cse.uta.edu, lagoa@enr.psu.edu

Abstract—Understanding the TCP congestion control mechanism from a global optimization point of view is not only important in its own right, but also crucial to the design of other transport layer traffic control protocols with provable properties. In this paper, we derive a global utility function and the corresponding optimal control law, known as TCP control law, which maximizes the global utility. The TCP control law captures the essential behaviors of TCP, including slow start, congestion avoidance, and the binary nature of congestion feedback in TCP. We find that the utility function of TCP is linear in the slow start phase and approaches the well-known logarithm function as the data rate becomes large in the congestion avoidance phase. We also find that understanding the slow start phase with a fixed threshold is critical to the design of new transport layer control protocols to enable quality of service features. Finally, we design an optimal, minimum-rate-guaranteed (MRG) traffic control law that shares the same utility function with the TCP control law, aiming at achieving fair share of network resources with TCP. By design, the MRG control law and the TCP control law jointly maximize the global utility and achieve globally stable control. Our preliminary simulation study of the MRG control law indicates that it is indeed TCP friendly and can provide minimum rate guarantee as long as the percentage of network resource consumed by the MRG flows is moderately small. Consequently, the MRG control law has the potential to be used to provide minimum rate guarantee for low-bandwidth Internet applications, such as live streaming media.

I. INTRODUCTION

The window-based congestion control used in the current Transmission Control Protocol (TCP) is a fully distributed, end-to-end traffic control mechanism. It relies solely on single-bit binary information feedback as input for the control, i.e., whether the forwarding path is congested or not. This binary information is acquired on the basis of source inferable information only, such as repetitive acknowledgments (ACKs) of the same segment, measured round-trip delay, and/or ACK timeout, without the assistance of the network nodes for the control and regardless of the link technology and queuing mechanism used in the network nodes. This makes TCP highly scalable and deployable at global scale.

The outstanding performance of TCP in dealing with network congestion has motivated researchers to look for a possible interpretation of TCP congestion control from an optimization-based, distributed traffic control framework [9] [10] [19]. The implication of finding such an interpretation is significant for two reasons. First, such an interpretation will unveil whether or not the TCP congestion control leads to globally stable operation and if so, what is the underlying global design objective the TCP congestion control protocol strives to achieve. Second, such an interpretation will pave the way for the design of new end-to-end congestion control protocols, which jointly with TCP, achieve globally optimal and stable traffic control.

Significant research efforts have been made on the general understanding of optimization-based, distributed control laws [1-4][6-12][17-21]. Some work also focuses on the understanding of TCP behavior from an optimization perspective [9] [10] [19]. In their seminal work, Kelly et al. [9] demonstrated that a user utility function of $\log(x)$ form (x is a flow data rate) leads to a control law that exhibits additive-increase-and-multiplicative-decrease (AIMD) behavior, resembling the TCP congestion avoidance phase. This encouraging observation has triggered significant research interests in an attempt to further understand the TCP behavior from an optimization point of view. In [10], Kunniyur, et al. showed that the TCP behavior without the slow start phase can be modeled using a framework based on a utility function minus a cost function. By taking into account randomness of packet loss, their model leads to a user utility function of the form $-1/x$ as x becomes large. Low [19] studied the various variations of TCP using a primal-dual nonlinear programming technique, that directly solves the original utility maximization problem. While the primal algorithms capture the TCP window control behavior (without the slow start phase) at the TCP source node, the dual algorithms translate into given active queue management (AQM) algorithms running in the network nodes.

The above TCP models successfully capture important as-

pects of the TCP congestion control, especially some variations of TCP that involve network nodes. However, in the existing work, the aperiodic, binary nature of end-to-end TCP congestion control is not well captured and the slow start phase of TCP control is not modeled. Based on the Sliding Mode technique in control theory, the family of optimization-based, distributed control laws found in [22] successfully addresses the case where information feedback for the control is discontinuous, aperiodic, and binary. On the basis of the family of control laws given in [22], in this paper, we aim at developing a more accurate model for end-to-end TCP congestion control and also a new end-to-end minimum-rate-guaranteed (MRG) control law.

This paper makes two major contributions. First, a utility function of TCP and the corresponding control law, called TCP control law, are derived. The control law captures the slow start, congestion avoidance, and the aperiodic, binary nature of the information feedback in TCP. The slow start phase is found to correspond to a linear term in the utility function. This term indicates that the user fairness can be improved by setting the slow start threshold to a larger fixed value for a flow with longer round-trip-time (RTT). In the congestion avoidance phase, the utility function converges to the well-known logarithm function as the data rate becomes large, which provides the so called weighted proportional fairness with the weight equal to the ratio between the additive increase rate and multiplicative decrease factor. These findings are tested against the ns-2 simulation results. Second, an MRG control law is derived, based on the same utility function as TCP, which by design, ensures fairness to TCP. The MRG control law and the TCP control law jointly achieve globally stable and optimal control. The simulation results indicate that the MRG control law is indeed TCP friendly and can provide minimum rate guarantee as long as the percentage of network resource consumed by the MRG flows is moderately small. These results provide insights on the possible design of an optimal, end-to-end MRG transport layer protocol to support low-bandwidth Internet applications, such as live streaming media and Voice-over-IP.

The rest of the paper is organized as follows: Section II derives the utility function of TCP and TCP control law. The proposed TCP model is tested against ns-2 simulation results in Section III and IV. Section V designs and tests an MRG control law. Finally, Section VI concludes the paper and describes our future work.

II. UTILITY FUNCTION OF TCP

In this section, we derive the global utility function of TCP based on a large family of distributed control laws that maximize the global utility of a very general form proposed in [22]. We first state the results in [22], which are relevant to the modeling of TCP. The problem can be stated as follows:

$$\max \sum_{i=1}^F U_i(x_i) \quad (1)$$

subject to network constraints

$$\sum_{i:l \in \mathcal{L}} x_i \leq B_l, \quad l \in \mathcal{L} \quad (2)$$

where $U_i(x_i)$ is the utility function for flow i ; F is the total number of flows in the network; x_i is the rate of flow i ; \mathcal{L} is the set of links in the network; and B_l is the bandwidth of link l in \mathcal{L} .

For the best effort (BE) service, the distributed control law [22] that solves the above problem is given by (for simplicity, we omit the index i),

$$\dot{x} = (z(t, x)[f(x) - (1 - \bar{c}g)])_{x=0}^+ \quad (3)$$

with

$$f(x) = 1 - e^{-\partial U(x)/\partial x} \quad (4)$$

and

$$(y)_{x=0}^+ = \begin{cases} \max(y, 0) & \text{if } x = 0 \\ y & \text{if } x \neq 0 \end{cases} \quad (5)$$

where x is the flow rate; $U(x)$ is a differentiable concave and strictly increasing function of x ; $z(x, t)$ is a strictly positive function; and cg is the binary congestion indicator ($cg=1$ if the packet forwarding path is congested and 0 otherwise); $\bar{c}g$ is the logical negation of cg .

Note that the above control law can faithfully capture the discontinuous, aperiodic, and binary nature of any end-to-end traffic control, such as the end-to-end TCP congestion control. This is because the required minimum input for the above family of control laws is a binary indicator, cg , indicating whether the forwarding path is congested or not. Moreover, it is shown in [22] that the above control laws are globally stable and converge to the optimal rate allocation. It is also shown that the corresponding control laws in discrete time domain with a finite time interval lead to stable and near optimal control.

On one hand, for any given concave user utility functions, a family of distributed control laws with positive $z(x, t)$ functions can be readily derived from Eqns. (3)-(5). On the other hand, an empirically designed, elastic control protocol, such as the TCP congestion control mechanism, can be reverse engineered by matching its behavior with the above family of control laws. This matching process may lead to a control law that best describes its behavior and an underlying utility function this control law maximizes. We take this latter approach. Namely, we derive the utility function and the control law that closely matches with the TCP congestion control behavior.

In this paper, we do not model the timeout effect of TCP. In other words, we model multiplicative-increase-and-multiplicative-decrease (MIMD) behavior in the slow start phase and additive-increase-and-multiplicative-decrease (AIMD) behavior in the congestion avoidance phase. Moreover, we assume that the threshold rate x_s between the slow start phase and the congestion avoidance phase is fixed. Namely, we model the steady state TCP behaviors where the threshold converges to a constant.

For $x < x_s$, i.e., in the slow start phase, assume that the increase rate is αx ($\alpha > 0$) and the decrease rate is βx ($0 < \beta \leq 1$). In the absence of congestion, i.e., $cg = 0$, the control law in Eqn. (3) is given by ¹

$$\dot{x} = z(t, x)f(x) = \alpha x \quad (6)$$

In the presence of congestion, i.e., $cg = 1$, we have

$$\dot{x} = z(t, x)[f(x) - 1] = -\beta x \quad (7)$$

Then we have the utility function

$$U(x) = x \log\left(1 + \frac{\alpha}{\beta}\right) \quad (8)$$

and

$$z(t, x) = \frac{\alpha x}{f(x)} = (\alpha + \beta)x \quad (9)$$

$U(x)$ is a differentiable, concave, and strictly increasing function, and $z(t, x)$ is positive for $x > 0$. So the MIMD control law achieves globally optimal rate allocation, maximizing a linear utility. The scale factor of the utility depends on the ratio α/β . A flow with a larger increase factor (α) or smaller decrease factor (β) has higher utility. For the slow start phase in TCP, assume $\alpha=1$ and $\beta=1/2$, then $U(x) = x \log 3$.

For $x > x_s$, i.e., in the congestion avoidance phase, assume the additive-increase rate is $\mu > 0$, and multiplicative-decrease rate is βx . In the absence of congestion, i.e., $cg = 0$, the control law is given by

$$\dot{x} = z(t, x)f(x) = \mu \quad (10)$$

In the presence of congestion, i.e., $cg = 1$, we have

$$\dot{x} = z(t, x)[f(x) - 1] = -\beta x \quad (11)$$

Then we have

$$U(x) = \left(\frac{\mu}{\beta} + x\right)[\log(\mu + \beta x) - 1] - x[\log(\beta x) - 1] + C \quad (12)$$

¹Here we assume the rate of a flow is positive after the flow initiates. In other words, the flow has a positive initial rate. For example, the initial flow rate is two packets per RTT in TCP-Reno

and

$$z(t, x) = \mu + \beta x \quad (13)$$

where C is a constant; $U(x)$ is a differentiable, strict concave, and increasing function; and $z(t, x)$ is positive. Hence the AIMD control law achieves globally optimal rate allocation maximizing utility function described in Eqn. (12). In the TCP congestion avoidance phase, $\beta = 1/2$, then $U(x) = (2\mu + x)[\log(\mu + x/2) - 1] - x[\log(x/2) - 1] + C$. Note that the β value is found to be larger than $1/2$ in [10] and [16]. However, our analysis in the next section shows that β value has limited effect on the equilibrium rate allocation and hence, we simply let $\beta = 1/2$.

We set C at a value such that $U(x)$ is continuous at this point. Then we have

$$C = x_s \log\left[\frac{(\alpha + \beta)x_s}{\mu + \beta x_s}\right] - \frac{\mu}{\beta} \log(\mu + \beta x_s) + \frac{\mu}{\beta} \quad (14)$$

Eqns. (6), (7) (for $x < x_s$) and (10), (11) (for $x > x_s$) are called the TCP control law, which maximizes the utility function given by Eqns. (8) (for $x < x_s$) and (12) (for $x > x_s$). The TCP control law captures the steady state TCP behaviors where the slow start threshold x_s is a constant. More specifically, TCP uses a dynamic slow start threshold. In the absence of TCP timeout, a flow always adjusts its slow start threshold to be half of the peak rate (the rate at the instance of congestion). When TCP is in the steady state, the peak rate is fixed and so is the slow start threshold. Whenever a congestion occurs, the rate is set to the slow start threshold and the flow immediately enters the congestion avoidance phase. As a result, the steady state TCP behavior is modeled by Eqn. (10), (11), and (12). In other words, the slow start phase plays an important role only during the initial TCP ramp-up phase.

The proposed, normalized utility function with $\mu = 0.1$ and $x_s = 1$, as well as the logarithm utility function ($\log(1 + x)$) are presented in Fig. 1. We see that the utility in the slow start phase constitutes a significant part of the proposed utility function. A flow achieves more utility in the slow start phase than that in the congestion avoidance phase. Hence, flows with larger slow start thresholds tend to grab the network resource more aggressively.

In the congestion avoidance phase, when $\beta x \gg \mu$, $U(x) \simeq (\mu/\beta)\log(x) + const$. For TCP, $U \simeq 2\mu\log(x) + const$. Namely, when the flow rate is large, TCP achieves the well-known weighted proportional fairness [9] with the weight value equal to μ/β , explicitly proportional to the additive increase rate μ . For TCP, the congestion window size is increased by one packet per RTT. As a result, the additive increase rate is inversely proportional to its RTT. A flow with a shorter RTT has higher additive increase rate. Hence, the utility, and thus the allocated rate, for a flow with a shorter RTT in a shared link is higher than that of a flow with a

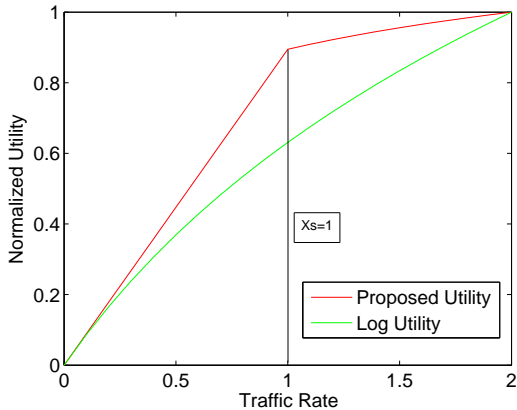


Fig. 1. Utility Functions

longer RTT. If the additive-increase rate is set to a constant value for every flow, then each flow has the same utility function and hence achieves the same allocated rate in a shared link, independent of RTT. This explains why keeping constant additive-increase rate in TCP leads to fair allocation of traffic rate to every flow, as observed in [5] and [13]. Utility functions with dependence on RTT were also proposed in [23] in the context of pricing models and in [19] based on AQM algorithms.

III. RATE ALLOCATION BASED ON UTILITY FUNCTION

The TCP control law (i.e., MIMD in the slow start phase and AIMD in the congestion avoidance phase) and the corresponding utility function in the previous section are derived in the continuous time domain without taking into account the effect of TCP timeout. A natural question to be asked is how accurate the control law and the corresponding utility function describe the TCP behavior. In this section, we test the accuracy of the proposed model by first comparing the measured TCP utility against the optimal utility, and then the measured rate allocation of TCP sessions against the optimal rate allocation.

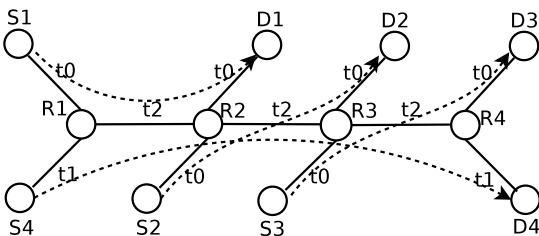


Fig. 2. Network topology I

Let us consider a network with a total number of 12 nodes, as shown in Figure 2. Assume there are 4 flows in the net-

work. Flow i that starts at the source node S_i and ends at the destination node D_i ($i=1, 2, 3$ and 4). The link bandwidth for each of the links between nodes R_1 and R_2 , R_2 and R_3 , and R_3 and R_4 is 100 Mbps (Mega bit per second) and the link bandwidth for each of the rest of links is 200 Mbps. The three 100 Mbps links are the bottleneck links, each of them is shared by two flows. Three different propagation delays (t_0 , t_1 and t_2) are assumed for different links as shown in Figure 2. We study three different cases with different link propagation delays as shown in Table I.

TABLE I
LINK PROPAGATION DELAY OF THREE CASES

	t_0	t_1	t_2
Case I	5 ms	5 ms	5 ms
Case II	2 ms	5 ms	5 ms
Case III	5 ms	5 ms	2 ms

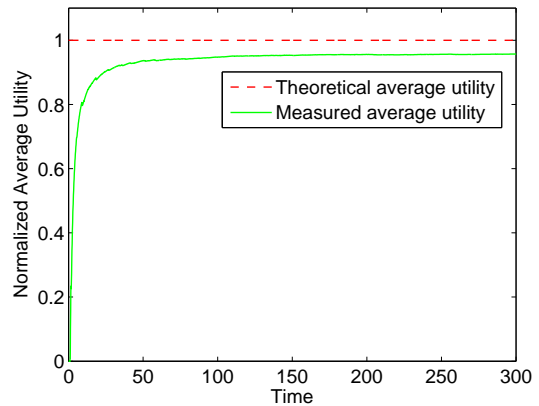


Fig. 3. Measured average TCP Utility

We first estimate the utility function in Eqns. (8) and (12) for the equilibrium rate allocations of TCP-Reno and compare it with the theoretical optimal utility. In TCP, the threshold of the slow start phase is dynamically adjusted based on the current congestion window size. Whenever congestion is detected, the slow start threshold is set to half of the current window size. In the simulation, we first measure an average threshold value (in Mbps) for each flow which is computed as the average threshold window size (in Mega bits) divided by the average measured RTT (in seconds). Then the theoretical maximum utility is calculated based on the measured average thresholds, and the measured utility of TCP at time t is estimated based on both the measured average thresholds and the average rates from time 0 to t of all the TCP flows. Both theoretical and measured average network utility for Case I are given in Figure 3 (similar results are obtained for Cases

II and III, which are not given here). The close match of the two curves indicates that the proposed model captures the underlying utility of TCP well.

To further test the accuracy of the proposed TCP model, we compare the measured rate allocations of TCP flows against the theoretical ones that maximize the proposed utility function. As mentioned before, a TCP flow is always in the congestion avoidance phase in the steady state, assuming that the timeout is ignored. Hence, the utility of each flow can be calculated based on Eqn. (12), assuming it is always in the congestion avoidance phase. Denote the additive increase rate and slow start threshold of flow i are μ_i and x_s^i , respectively. Then the utility of the system is

$$U(x_1, x_2, x_3, x_4) = \sum_{i=1}^4 \left\{ \left(\frac{\mu_i}{\beta} + x_i \right) [\log(\mu_i + \beta x_i) - 1] - x_i [\log(\beta x_i) - 1] + C_i \right\} \quad (15)$$

subject to

$$x_i + x_4 \leq B \quad \text{for } i=1, 2 \text{ and } 3$$

where B is the shared link bandwidth. Since flows 1, 2 and 3 have the same RTT, the flow rates and the additive increase rates of these flows are the same, i.e., $x_1 = x_2 = x_3 = x = B - x_4$ and $\mu_1 = \mu_2 = \mu_3 = \mu$. Then the maximum utility is achieved by setting $\partial U(x_1, x_2, x_3, x_4) / \partial x_4 = 0$, i.e., the optimal rate, x_4 , satisfies the following equation:

$$\left[\frac{\mu/\beta + B - x_4}{B - x_4} \right]^3 = \frac{\mu_4/\beta + x_4}{x_4} \quad (16)$$

In Eqn. (16), the power coefficient takes value 3 because flow 4 traverses 3 bottleneck links. In general, the power coefficient will be n if flow 4 traverses n bottleneck links. When $x = B - x_4 \gg \mu/\beta$ (in all three cases of the simulation, $\beta x/\mu > 50$), Eqn. (16) can be approximated by

$$\frac{x_4}{x} = \frac{x_4}{B - x_4} \simeq \frac{\mu_4}{3\mu} \quad (17)$$

The flow allocation is dependent on the additive increase rate and the number of bottleneck links, but is independent of β and B . For that reason, we have simply set $\beta = 1/2$. The flow rate is proportional to the additive increase rate and inversely proportional to the number of shared bottleneck links. Hence TCP achieves proportional fairness: inversely proportional to the RTT and the number of shared bottleneck links. When $\mu = \mu_4$, the flow ratio $x_4 : x \simeq 1 : 3$, the same as the optimal rate allocation for the logarithm utility function with a constant scale factor.

In the congestion avoidance phase, the congestion window is increased by one packet per RTT. If the packet size is w

bits, the additive increase rate of a flow is w/RTT . In our simulation, the packet size is set to 1,000 bytes, i.e., $w = 8000$. Table II gives the theoretical rate ratios based on Eqn. (16) for all three cases. Here the RTT is simply taken to be the round-trip propagation delay. The RTTs for flow 4 and other 3 flows in all three cases are: 30 ms and 50 ms in Case I; 18 ms and 50 ms in Case II; and 30 ms and 38 ms in Case III.

TABLE II
THEORETICAL OPTIMUM RATE ALLOCATION FOR THREE CASES

Rate Ratio	Case I	Case II	Case III
$x_4:x$	0.2	0.12	0.26

Although not tightly coupled with the queuing mechanisms in use in network nodes, the family of control laws presented in the previous section does implicitly assume that different flows sharing the same congested link should have equal opportunity to sense the congestion. Hence we let all the flows share a single Random Early Detection (RED) first-in-first-out (FIFO) queue at each router. Flows are generated by TCP-Reno. Each simulation run is 300 seconds and the statistics are collected during the last 100 seconds.

Figures 4 - 6 show the flow ratio versus time for three cases, respectively. The flow rate x for the three cases is computed as the average rate of flows 1, 2 and 3. Here, we observe that the measured TCP flow ratios are close to the theoretical ratios for all three cases. These results demonstrate that the derived utility function and hence the TCP control law captures the essential behavior of TCP.

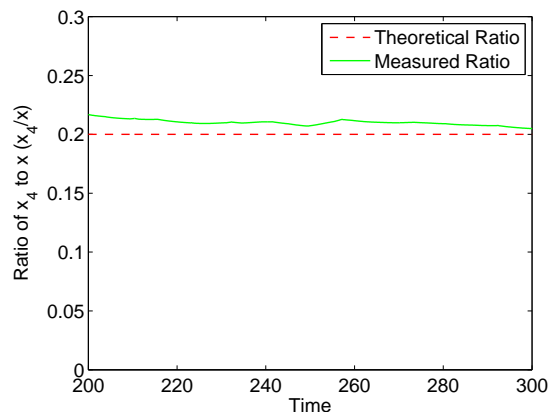


Fig. 4. Rate Ratio, Case I

IV. EFFECT OF SLOW START THRESHOLD

For TCP using dynamical slow start threshold, the slow start phase has no impact on the equilibrium rate allocation

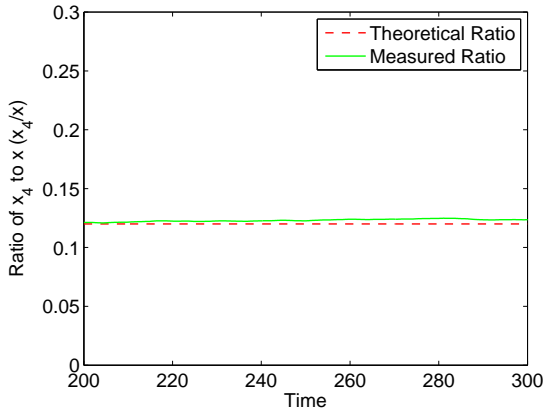


Fig. 5. Rate ratio, Case II

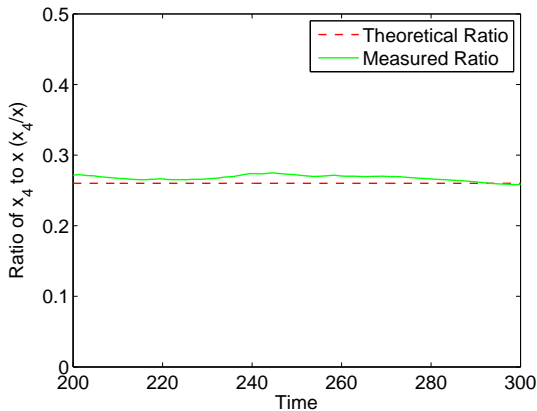
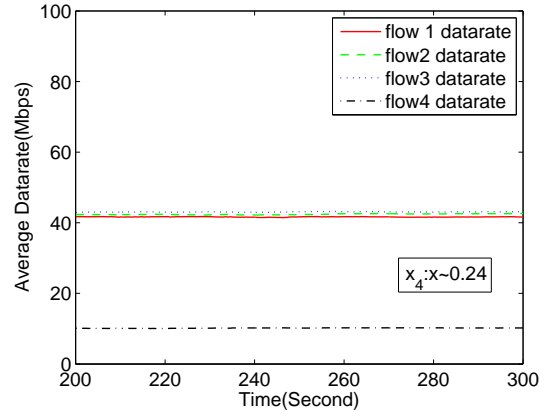
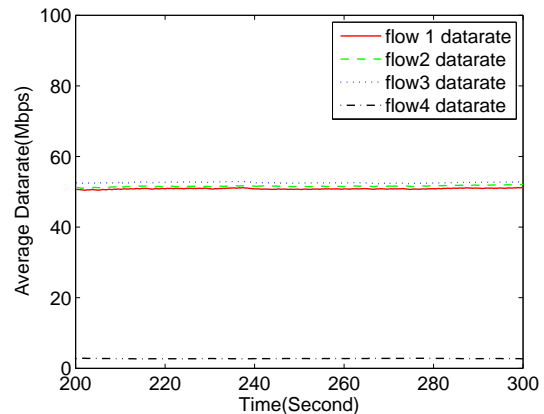


Fig. 6. Rate Ratio, Case III

among flows, given that the TCP timeout is a rare event. To test the linear utility function for the slow start phase, we modify TCP-Reno by setting a fixed threshold value x_s for every flow in the network. Then we run this modified TCP-Reno on the same network topology with propagation delay given in Case I. We consider $x_s > B/2$ and $x_s < B/2$, separately.

For fixed x_s and $x_s > B/2$, due to the linear utility function in the slow start phase, the theoretical optimal rate allocations are approximately: $x_1 = x_2 = x_3 \simeq x_s$ and $x_4 \simeq B - x_s$, independent of RTT. For $x_s = 80$ Mbps, $x_4 \simeq 20$ Mbps, $x_1 = x_2 = x_3 = x \simeq 80$ Mbps, i.e., the ratio $x_4 : x \simeq 0.25$; For $x_s = 100$ Mbps, $x_4 \simeq 0$, $x_1 = x_2 = x_3 \simeq 100$ Mbps. Figures 7 and 8 present the simulation results for TCP-Reno with fixed threshold at $x_s = 80$ and 100 Mbps. We see that the rate ratio $x_4 : x$ is around 0.24 for $x_s = 80$ Mbps, very close to the theoretical ratio. The rate of flow 4 is close to 0 for $x_s = 100$ Mbps. For the linear utility, a flow sharing a single bottleneck link with

a flow that traverses multiple bottleneck links tends to grab all the available link bandwidth. These results strongly suggest that the utility of TCP in the slow start phase is a linear function.

Fig. 7. Rate allocation, with fixed threshold $x_s=80$ MbpsFig. 8. Rate allocation, with fixed threshold $x_s=100$ Mbps

Now let's look at the case where x_s is fixed and $x_s < B/2$. Due to the linear part in the utility function, the optimal rate of each flow in the steady state is above the slow start threshold x_s . Among all the flows, flow 4 has the longest RTT and thus the smallest average and peak rate (x_p^4) in the steady state. First consider the case where $x_s < x_p^4/2$. When congestion occurs, the rates of all flows are reduced to half of their respective peak rates. The rate of flow 4 is reduced to $x_p^4/2$, larger than x_s . By ignoring the timeout effect, when congestion occurs, the flow rates are reduced by half and then all the flows enter the congestion avoidance phase immediately. So the utility function for flows with a fixed small slow start threshold is expected to be the same as that for TCP with a dynamic threshold.

For the case where $x_s < B/2$ and $x_s > x_p^4/2$, the peak rate of flow 4 (i.e., x_p^4) is less than $2x_s$ and greater than x_s (above the threshold) in the steady state. When congestion occurs, flow 4 goes to the slow start phase due to the fact that the rate is reduced to $x_p^4/2$ which is below the threshold. In this case, flow 4 goes back and forth between the congestion avoidance phase and the slow start phase. In this particular case, it is difficult to analytically derive the optimal rate allocations using the utility function given in Eqns. (8) and (12). To simplify the analysis, here we derive an approximate utility function, which can be easily used to estimate the optimal rate allocation. Note that, after flow 4 goes to the slow start phase, it enters the congestion avoidance phase at flow rate x_s after one RTT. So flow 4 is in the additive increase phase most of the time. If we consider the rate change from the peak value to the slow start threshold to be a single rate decrease at rate $x - x_s$, then the flow can be viewed as always in the congestion avoidance phase. The control law can be approximately modeled as: (1) the rate increases at rate μ in the absence of congestion (the same as in Eqn. (10)), and (2) the rate decreases at $x - x_s$ in the presence of congestion, i.e.,

$$\dot{x} = z(t, x)f(x) = -x + x_s \quad (18)$$

Then the utility function derived from Eqns. (10) and (18) for this approximated control law can be expressed as

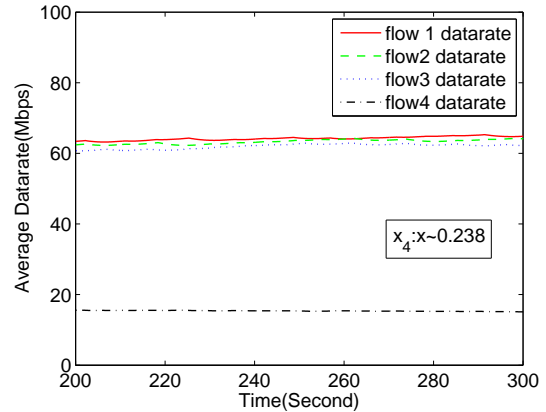
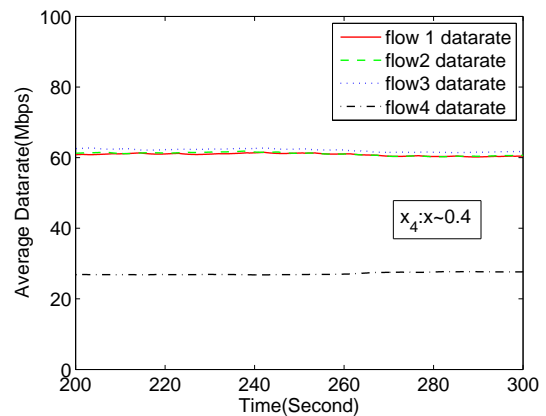
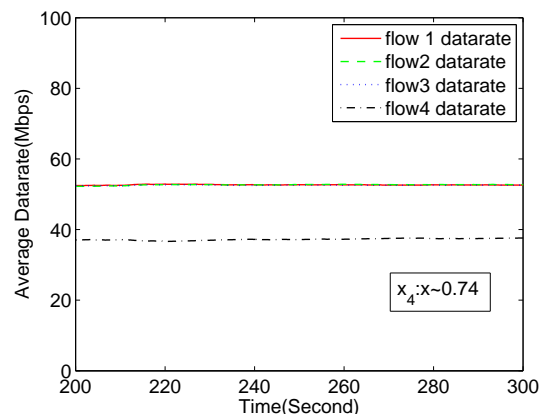
$$U^2(x) = (\mu + x - x_s)\log(\mu + x - x_s) - (x - x_s)\log(x - x_s) \quad (19)$$

Now the optimal rate of flow 4 can be computed from the utility function expressed in Eqn. (19). Let us study the rate allocation in Case I using fixed x_s at 10, 20, and 40 Mbps. For $x_s = 10$ and 20, the peak rates of flows 1, 2, and 3 in steady state are over $2x_s$, and the utility function of these flows is given by Eqn. (12), denoted as $U^1(x)$. For flow 4, the peak rate in steady state is less than $2x_s$, and its utility function is given by Eqn. (19). In this case, the optimal rate allocation maximizes the total utility $U(x_1, x_2, x_3, x_4) = \sum_{i=1}^3 U^1(x_i) + U^2(x_4)$. For $x_s=40$, the peak rates of all four flows in steady state are less than $2x_s$. In this case, we have $U(x_1, x_2, x_3, x_4) = \sum_{i=1}^4 U^2(x_i)$. The optimal rate allocation with different slow start thresholds are given in Table III.

TABLE III

THEORETICAL OPTIMUM RATE RATIO WITH DIFFERENT SLOW START THRESHOLDS

Rate Ratio	$x_s=10$	$x_s=20$	$x_s=40$
$x_4:x$	0.223	0.373	0.769

Fig. 9. Rate Allocation for fixed threshold $x_s=10$ Fig. 10. Rate Allocation for fixed threshold $x_s=20$ Fig. 11. Rate Allocation for fixed threshold $x_s=40$

Figures 9-11 show the simulated rate allocation of four flows at $x_s = 10, 20$ and 40 Mbps, respectively. The rate ratios ($x_4 : x$) are close to their theoretical counterparts for all

three cases. We observe that despite the fact that flow 4 traverses three critical links and has a larger RTT value than the rest of the flows, its rate increases as the threshold increases and stays above or close to the threshold value. This suggests that the fixed slow start threshold could play a key role for providing user fairness, provided that it does not exceed in certain value ($x_s < B/2$ for the current case).

Based on these results, we conclude that the linear part in the utility function plays a key role for rate allocation. Although the slow start phase does not play an important role in TCP (assuming TCP timeout is a rare event) due to its use of a dynamic slow start threshold, understanding the slow start phase provides valuable insights on how to design new end-to-end transport layer protocols, in particular, the protocols that can provide guaranteed rate services. In summary, we verify that the utility functions in the slow start phase is linear.

V. CONGESTION CONTROL LAW FOR MINIMUM RATE GUARANTEED SERVICE

The above TCP model indicates that a flow with a longer RTT and sharing more bottleneck links achieve lower rate. As a result, it cannot be used to support applications requiring rate guarantees. So an interesting question to be answered is the following: is it possible to design an end-to-end congestion control protocol to support Minimum Rate Guaranteed Service (MRGS)? As a first step towards answering this question, in this section, we derive a minimum-rate-guaranteed (MRG) congestion control law. This control law possesses the following three desirable features. First, it maximizes the same utility function as the TCP control law does, i.e., the utility function given in Eqns. (8) and (12). Hence, by design, the MRG control law is TCP fair and friendly. Second, just like end-to-end TCP, it is end-to-end in the sense that it can be implemented without involving the network nodes. Third, flows using the MRG control law can coexist with the flows using the TCP control law, and they jointly drive the network to a globally stable and optimal state. In what follows, we present this control law and provide preliminary testing results based on simulation.

According to [22], the optimal control law for single path MRG service (MRGS) can be generally expressed as follows,

$$\dot{x} = (z(t, x)[f(x) - (1 - \bar{c}g \times r^m)])_{x=0}^+ \quad (20)$$

with

$$r^m(x) = \begin{cases} 1 & \text{if } x \geq \Theta \\ r_{max}^m > 1 & \text{if } x < \Theta \end{cases} \quad (21)$$

where Θ is the targeted minimum rate and r_{max}^m is a tunable parameter. Letting the utility function be the one in Eqns. (8) and (12), we have the MRG control law as follows: (1) when

the rate x is above the targeted minimum rate Θ , the control law is the same as the TCP control law; (2) when the rate x is smaller than the targeted minimum rate Θ and the path is congested, the decrease rate is $-x/2$, the same as the TCP control law. Without congestion, the increase rate is

$$\dot{x} = \begin{cases} (3r_{max}^m - 1)x/2 & \text{if } x < x_s \\ (r_{max}^m - 1)x/2 + \mu r_{max}^m & \text{otherwise} \end{cases} \quad (22)$$

If $r_{max}^m = 1$, the MRG control law degenerates to the TCP control law. If r_{max}^m is greater than 1, the exponential increase rate for the MRG control law is faster than TCP in the slow start phase, and the increase rate is still exponential in the congestion avoidance phase, but at a lower rate. In our implementation of the MRG control law, the rate of a flow at time t is calculated as the congestion window size at time t divided by the average measured RTT at time t . We also use a dynamic slow start threshold. In other words, when the path is congested, the slow start rate is set to half of the current flow rate and the rate is reduced by half. However, the increase rate is always exponential if the rate is smaller than the target rate. Hence the target rate plays a role similar to the fixed threshold and hence it may provide rate guarantee.

From Eqn. (22), we note that if $r_{max}^m = 3$, the increase rate is $x + 3\mu \simeq x$, i.e., the flow rate is doubled every RTT. If the rate is reduced by half from the target rate when congestion occurs, such increase rate brings the flow rate back to the target rate in one RTT. A larger r_{max}^m results in a larger increase rate. However, a larger r_{max}^m still takes the flow rate back to the target rate in one RTT, the same as the case of $r_{max}^m = 3$. Hence any r_{max}^m for $r_{max}^m > 3$ has the similar effect as $r_{max}^m = 3$. $r_{max}^m > 3$ has an impact only for the case that a flow senses continuous congestions and goes back to initial slow start rate. Hence we only consider r_{max}^m in a range between 1 and 3.

We conducted preliminary simulation testing of the MRG control law. Consider a network topology with 14 nodes as shown in Figure 12. In this network, each link has a bandwidth of 20 Mbps and the propagation delay of each link is given in Figure 12. There are 5 flows running in this network. Flow i starts at the source S_i and ends at the destination D_i ($i = 1, 2, 3, 4$ and 5). The RTTs excluding queue delay for flows 1 to 5 are 12, 14, 50, 22, and 80 ms, respectively. Assume that flow 4 requires 3 Mbps minimum rate guarantee, whereas all other flows are elastic BE flows.

First, we run all five flows using TCP-Reno. Figure 13 shows the average rate of the five flows. The rates of flows 1, 2 and 3 are around 12 Mbps, the rate of flow 5 is about 5 Mbps. The rate of flow 4 is below 2 Mbps, much smaller than its required minimum rate of 3 Mbps.

Now we run MRG control law for flow 4 and TCP-Reno for other flows. We find that the target rate for flow 4 can be

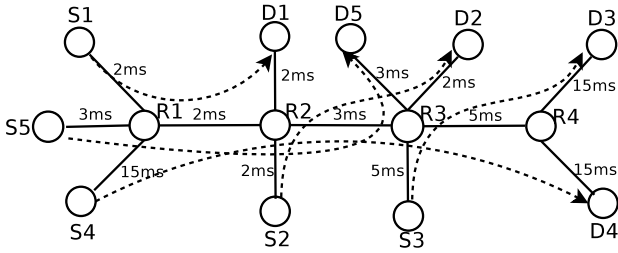


Fig. 12. Network topology II

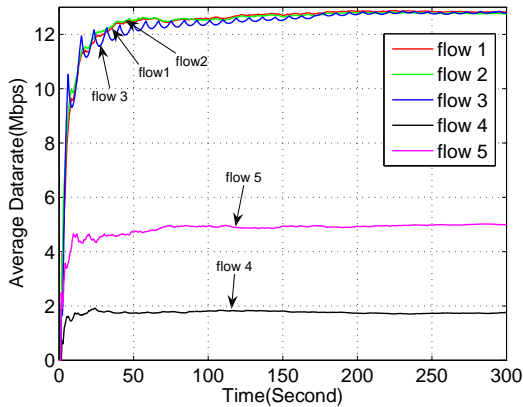
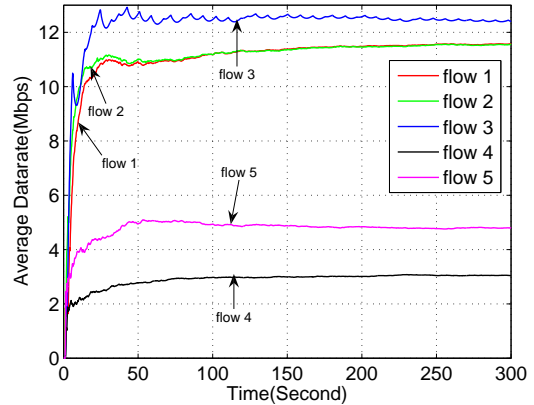
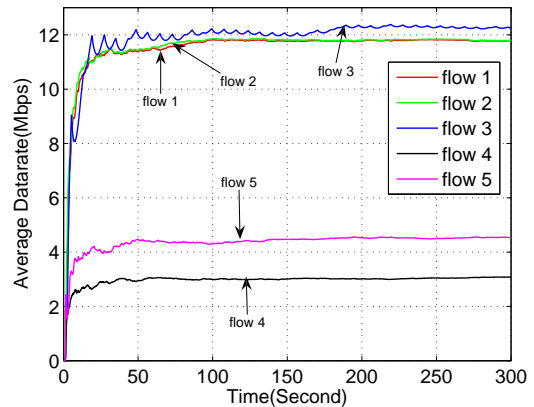


Fig. 13. Flow Rate Allocation, TCP-Reno for all Flows

achieved when r_{max}^m is 2 or above. When r_{max}^m is less than 2, the average rate of flow 4 is increased but cannot reach its target rate. Figures 14 and 15 show the average rate of five flows. The rate of flow 4 reaches its target rate at both $r_{max}^m = 2$ and 3, respectively. Note that in both cases (see Figures 14 and 15), the four TCP flows incur minor rate losses compared with the respective rates in Figure 13. When the target rate of flow 4 (not presented here) is increased further to 4 Mbps, the MRG control law can provide about 3.5 Mbps, less than its target rate, while the other flows incur very little further rate losses. This indicates that the MRG control law is TCP friendly. In other words, it has the ability to better exploit the available network resource than TCP to achieve its target rate. In the meantime, when the network resource is not sufficient to sustain its target rate, it has the ability to back off to ensure friendliness to TCP flows. These observations are encouraging but preliminary. It is of practical importance to understand, from a theoretical point of view, under what conditions, the target rate can be achieved, which will be exploited in our future work.

The proposed MRG control law has great potential to be used to support low-bandwidth Internet applications with minimum rate guarantee, such as live streaming media and Voice-over-IP. For example, it can be applied to the popular

Fig. 14. Flow Rate Allocation, MRG Control Law for flow 4 with $r_{max}^m = 2.0$ Fig. 15. Flow Rate Allocation, MRG Control Law for flow 4 with $r_{max}^m = 3.0$

P2P live streaming media applications, such as PPLive [14] and PPStream [15]. There are mainly two types of P2P multimedia applications: live streaming data and multimedia file downloads. Both applications use TCP as its reliable data transport. The first type of applications requires guaranteed average throughput rate performance, which is usually a few hundreds of Kbps. The received data in a peer is buffered for a few minutes before its replay. Hence, a short rate decrease does not affect the replay quality if the average rate reaches its target rate at a timescale of tens of seconds. The second type of applications is elastic. Due to the relatively small bandwidth demand of live streaming applications, support such applications using the proposed MRG control law is expected to have limited impact on other elastic traffic.

VI. CONCLUSIONS AND FUTURE WORK

The TCP congestion control mechanism used in today's Internet is a fully distributed, end-to-end traffic control mechanism. It uses only resource inferable, single-bit binary information as input for the source control, i.e., whether the forwarding path is congested or not. These salient features make TCP highly robust and deployable at global scale.

This paper aimed at understanding, interpreting, and modeling of end-to-end TCP behavior on the basis of a global optimization framework. First, we derived a utility function of TCP and its corresponding control law, called TCP control law. The TCP control law captures the TCP slow start, congestion avoidance, and the aperiodic, binary nature of the TCP information feedbacks. We found that the utility function in the slow start phase is linear. This term indicates that the user fairness in terms of achievable user utilities can be improved by setting the slow start threshold at a fixed, larger values for flows with longer RTT values. Second, a Minimum Rate Guaranteed (MRG) control law is derived based on the same utility function of TCP. The MRG control law and the TCP control law jointly achieve globally stable and optimal control. The preliminary simulation results indicated that the MRG control law can provide minimum rate guarantee under certain conditions. These results provide significant insights on the possible design of an MRG transport layer protocol to support emerging Internet applications, such as P2P live streaming media.

As our future work, we will study, from both theoretical and engineering perspectives, under what conditions the target rate of a flow using the proposed MRG control law can be achieved. We also plan to develop other possible optimal, end-to-end traffic control laws and protocols to meet various application needs, such as assured forwarding service and upper bounded rate service.

REFERENCES

- [1] T. Alpcan and T. Basar, "A Utility-Based Congestion Control Scheme for Internet-Style Networks with Delay", *IEEE INFOCOM*, 2003
- [2] D. Andersen, H. Balakrishnan, F. Kasshoek, and R. Morris, "Resilient Overlay Networks", *ACM Symposium on Operating Systems Principles*, 2001.
- [3] Z. Duan, Z. Zhang, and Y. T. Hou, "Service Overlay Networks: Slas, QoS and Bandwidth Provisioning", *IEEE/ACM Transactions on Networking*, v13(6), pp 870-883, 2003.
- [4] A. Elwalid, C. Jin, S. Low, and I. Widjaja, "Mate: Mpls adaptive traffic engineering", *IEEE INFOCOM*, 2001.
- [5] S. Floyd. "Connections with Multiple Congested Gateways in packet-Switched Networks, Part 1: one-way Traffic", *ACM Computer Communications Review*, v21(5), pp30-37, 1991.
- [6] S. J. Golestani and S. Bhattacharyya, "A class of end-to-end congestion control algorithms for the internet", *IEEE International conference on Network Protocols (ICNP)*, 1998.
- [7] S. Kandula, D. Katabi, B. Davie, and A. Charney, "Texcp: Responsive yet stable traffic engineering", *ACM SIGCOMM*, 2005.
- [8] K. Kar, S. Sarkar and L. Tassiulas, "A Simple Rate Control Algorithm for MAXimizing Total User Utility", *IEEE INFOCOM*, 2001.
- [9] F. Kelly, A. Maulloo, and D. Tan, "Rate Control in Communication Networks: Shadow Prices, Proportional Fairness and Stability", *Journal of the Operational Research Society*, v49, pp237-252, 1998.
- [10] S. Kunniyur and A. Srikant, "End-to-End Congestion Control Schemes: Utility Functions, Random Losses and ECN Marks", *IEEE/ACM Transactions on Networking*, 11(5), 2003.
- [11] T. Harks and T. Poschwatta, "Priority Pricing in Utility Fair Networks", *IEEE ICNP*, 2005.
- [12] T. Harks and T. Poschwatta, "Utility Fair Congestion Control for Real-Time Traffic", *IEEE INFOCOM*, 2005.
- [13] T. R. Henderson, E. Sahouria, S. McCanne and R. H. Katz, "On Improving the Fairness of TCP Congestion Avoidance", *IEEE GLOBECOM*, 1999.
- [14] <http://www.pplive.com>
- [15] <http://www.ppstream.com>
- [16] P. Hurlley, J. L. Boudec, and P. Thiran, "A Note on the Fairness of Additive Increase and Multiplicative Decrease", *16th International Teletraffic Congress*, 1999.
- [17] R. La. V. Anantharam, "Utility-Based Rate Control in the Internet for Elastic Traffic", *IEEE Transactions on Networking*, v10(2), pp272-286, 2002.
- [18] C. M. Lagoa, H. Che, and B. A. Movsichoff, "Adaptive control algorithms for decentralized optimal traffic engineering in the internet", *IEEE/ACM Transactions on Networking*, v12(3), pp. 415428, 2004.
- [19] S. H. Low, "A Duality Model of TCP and Queue Management Algorithms", *IEEE/ACM Transactions on Networking*, 11(4), pp525-536, 2003.
- [20] S. Low and D. Lapsley, "Optimization flow control, i: Basic algorithm and convergence", *IEEE/ACM Transactions on Networking*, v7(6), pp. 861874, 1999.
- [21] B. Movsichoff, C. Lagoa, and H. Che, "Decentralized optimal traffic engineering in connectionless networks", *IEEE Journal on Selected Areas in Communications*, v23(2), pp 293303, 2005.
- [22] B. A. Movsichoff, C. M. Lagoa and H. Che, "End-to-End Optimal Algorithms for Integrated QoS, Traffic Engineering, and Failure Recovery", *IEEE/ACM Transactions on Networking*, to appear, available at <http://crystal.uta.edu/hche/PUBLICATIONS/papers/binary.pdf>
- [23] J. Pongsajapan and S. Low, "Reverse Engineering TCP/IP-like Networks Using Delay-Sensitive Utility Functions", *IEEE INFOCOM*, 2007.