Improving the Convergence and Stability of Congestion Control Algorithm

Xiaomeng Huang*, Chuang Lin†, Fengyuan Ren†, Guangwen Yang*, Peter D. Ungsunan†, and Yuanzhuo Wang†
*Department of Computer Science and Technology, Tsinghua National Laboratory for Information Science and Technology
Tsinghua University, Beijing, 100084, China
Email: huangxiaomeng@mail.tsinghua.edu.cn, ygw@tsinghua.edu.cn
†Department of Computer Science and Technology, Network Institute
Tsinghua University, Beijing, 100084, China
Email: {clin, renfy, hongsunan, yzwang}@csnet1.cs.tsinghua.edu.cn

***Abstract*— The traditional TCP congestion control is inefficient for high speed networks and it is a challenge to design a high speed replacement for TCP. By simulating some existing high speed protocols, we find that these high speed protocols have limitations in convergence and stability. To address these problems, we apply a population ecology model to design a novel congestion control algorithm–Coupling Logistic TCP(CLTCP). It is based on bandwidth pre-assignment that is similar to XCP and MaxNet. The pre-assignment rate factor is computed in the routers based on the information of the router capacity, the aggregate incoming traffic and the queue length. Then the senders adjust the sending rate according to the pre-assignment rate factor which carries by the packet to strengthen the convergence and stability of transport protocol. The theoretical analysis and simulation results show that CLTCP provides not only fast convergence and strong stability, but also high utilization and fair bandwidth allocation regardless of round trip time.**

I. Introduction

The convergence of congestion control algorithms usually studies the time for transport control system transmits from the initial state to the steady state. The two aspects of this issue are convergence to efficiency and convergence to fairness [15]. When a newly-starting flow joins the network, it is anticipated that the new flow should grab the available bandwidth of the link as soon as possible. It is emphasizing the time for convergence to efficiency. As to the time for convergence to fairness, when a newly-starting flow joins the network where the existing flows have taken the whole bandwidth, it is anticipated that the new flow should achieve fair bandwidth allocation as soon as possible.

According to the Additive Increase Multiplicative Decrease(AIMD) algorithm [1] [10] used in TCP and supposing the throughput of a TCP flow in steady state is P, we know that the time for convergence to efficiency and the time for convergence to fairness of TCP is $O(P)$ [24], that is to say the AIMD algorithm converges linearly to efficiency and fairness and it implies TCP will take a long period of time to converge to efficiency and fairness in high speed networks. Therefore TCP attempts to improve the convergence by using a slow-start algorithm in its starting phase. But the convergence speed in congestion avoidance phase is still slow.

This problem motivates the proposal of several novel transport protocols, such as HSTCP [4], STCP [12], XCP [11], EMKC [24], VCP [23], EVLF-TCP [9] and many others, each with their own strengths and limitations. HSTCP and STCP improve convergence by using a more aggressive increasing and more conservative decreasing algorithm with the cost of a higher loss ratio than the AIMD algorithm. Meanwhile, this method makes the RTT unfairness problem of HSTCP and STCP more serious than that of TCP. EMKC, VCP and EVLF-TCP allocate network resources effectively at the end system through explicitly feeding back the state information of the router, such as the loss ratio, load factor and virtual load factor. However, EMKC, VCP and EVLF-TCP only improve the convergence to efficiency from $O(P)$ to $O(\ln P)$, i.e., exponential convergence to efficiency. Their convergence to fairness is still kept as $O(P)$. On the contrary, XCP improves the convergence by allocating the bandwidth of each flow by the router directly, so that the convergence to efficiency and convergence to fairness are both $O(1)$, i.e., constant convergence. As presented in some research, XCP can be unstable and cannot always achieve Max-Min fairness in multi-congested gateway networks. We believe that the key reason is that XCP is too sensitive to network load so that it cannot provide strong stability and fairness in complex topologies.

Another important requirement of congestion control algorithms is stability. The congestion control algorithm needs to be stable and adaptive with a wide range of change in network parameters, such as link bandwidth, flow number and round trip time. A lot of research about the stability of transport protocols have shown stability criterions for different transport protocols [18] [7] [6] [2] [20] [17]. Whether these protocols are stable or not depend on not only the control parameters of the congestion control algorithm, but also the network parameters. So the stability of these protocols is restricted by the network parameters. If the network parameters are not located in the region which satisfies these stability criterions, these protocols may be unstable.

Therefore, it is an urgent issue to enhance the convergence of the transport protocol, and weaken or eliminate the influence

1-4244-1588-8/07/$25.00 ©2007 IEEE

of network parameters to the stability of the transport protocol as much as possible.

This paper proposes a novel congestion control algorithm which provides fast convergence and global asymptotic stability based on the special characteristics of the "Logistic Model" in population ecology [19] [16]. This algorithm is implemented by the explicit rate pre-assignment mechanism. At the same time, theoretical analysis according to stability and convergence has determined the impact of control parameters on the algorithm performance, and favorable performance of the algorithm has been confirmed through simulation on the NS2 simulation platform.

The remainder of this paper is organized as follows. Section II discusses ideal congestion control and the basic relationship between congestion control and the Logistic Model. Section III presents the concrete congestion control algorithm. Section IV analyzes the global asymptotic stability and convergence of the algorithm. Section V introduces the implementation of the corresponding transport protocol and simulation results are given in Section VI. Finally, Section VII concludes.

II. Design Rationale

A. Ideal congestion control

The congestion control algorithm may be divided into the link algorithm and the source algorithm [5] [3] [13]. The link algorithm, running in the router, examines the congestion of the network, and produces congestion signals, such as dropped packets, delay, explicit congestion notification, explicit packet loss rate and explicit load factor. The source algorithm running in the end system, adjusts the sending rate of the end system according to the congestion signal. The main design issues of the congestion control algorithm are to select the appropriate congestion signal for the link algorithm and find the best way to respond to it in the source algorithm.

In general, explicit congestion feedback schemes use direct communication from the router to tell the end system the state of the network precisely, accomplished by sending special packets or by changing some fields in packets as they travel through the routers. The use of explicit congestion feedback usually results in superior congestion control protocols that converge faster and have a lower packet loss rate than protocols using implicit congestion feedback.

Compared to the sawtooth shape of the AIMD mechanism, we believe that the sigmoid curve of the source algorithm is better suited for ideal congestion control in high speed networks. As shown in Fig.1, the sigmoid control curve increases the sending rate gently at the initial phase, accelerates exponentially in the middle stage and finally approach the upper limit of network capacity. The advantages of the sigmoid control curve are:(1) Not sending too many packets at the initial phase to avoid the burst traffic. (2)Exponentially increasing when the available capacity is sufficient. (3)Avoiding congestion as far as possible when the load is heavy. (4) Allocating network resources effectively, and avoiding the waste of network resources caused by the oscillation of the AIMD mechanism.

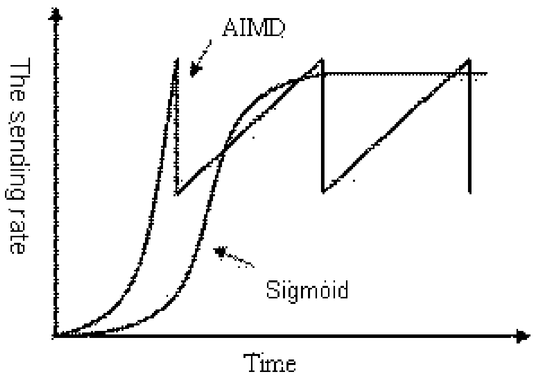


Fig. 1. Comparison of the sigmoid control curve with the AIMD control curve

B. Logistic Model

Firstly, we introduce the foundation of the Logistic Model. It is a population ecology model that studies the dynamics of populations in ecology. In the Logistic Model, the population number $x(t)$ in generations is expressed as:

$$\dot{x} = rx(1 - \frac{x}{K}) \tag{1}$$

Parameter r is the intrinsic rate of increase, which can be interpreted as the difference between the birth rate and the death rate of the population. Parameter K is the upper limit of population growth and it is called carrying capacity. It is usually interpreted as the amount of resources expressed in the number of organisms that can be supported by the resources. The population growth ratio $\frac{dx}{xdt}$ declines with the population number x and reaches 0 when $x = K$. If the population number exceeds K, then the population growth ratio becomes negative and the population number declines. The curve of the Logistic Model is just a sigmoid curve.

Overall, it is easy to see that the Logistic Model consists of: (1)the intrinsic rate of increase r, and (2) the density-dependent factor $(1 - x/K)$. When there is no resource limitation, the population number will exponentially increase with an intrinsic rate of increase. However, when the resources are consumed gradually, the density-dependent factor will have a greater influence on the population growth rate, and finally force the population number to achieve an equilibrium state. The Logistic Model provides a mature mathematical method to analyze how the populations share limited resources so that it is natural to migrate the Logistic Model to do congestion control.

We find that it is difficult to enforce the congestion control using the Logistic Model directly. The objection is based on the fact that when the network aggregate traffic exceeds the limited bandwidth, the packets will queue in the router buffer. The problem is that the queueing phenomenon does not exist in the population ecology models. In order to control the behavior of queues, we need to reconstruct a more reasonable density-dependent factor which links the Logistic Model with the queueing model.

Some researchers also have attempted to study congestion control usingthe Logistic Model. For example, M. Welzl [21] used the Logistic Model directly to design CADPC. However, since it uses the Logistic Model directly, the influence of queuing phenomena on congestion control is not considered.

III. Algorithm Design

Similar to XCP, our novel mechanism is an explicit bandwidth pre-assignment mechanism. Its principle is shown in Fig.2. Each router maintains a pre-assignment rate factor r. The basic adjusting strategy of r is when the link is underloaded, r gradually increases, otherwise decreases gradually. And the minimal r value among all links along with the path will be sent to the end system. After the end system receives the pre-assignment rate factor r, it treats this the value of r as the upper limit of capacity that the network can provide, and then it makes the sending rate x approach the r value quickly. This method is also similar to the "MaxNet" method presented in [22].

Consider a network with a set L of links, and let C_l be the finite capacity of link l, $r_l(t)$ be the pre-assignment rate factor by the link l, $q_l(t)$ be the instantaneous queue size of the link l, for $l \in L$. Let a router be a non-empty subset of L, and write P for the set of possible routes. If $l \in p$, then the link l lies on the route p. If $p \in l$, then the route p passes through the link l. Associate a route with a flow, and suppose that the rate $x_p(t)$ is allocated to user p.

We proposed the following congestion control model, consisting of the source algorithm and the link algorithm.

A. Link Algorithm

Consider the system of differential equations:

$$\dot{r}_l(t) = \beta r_l(t)(1 - \frac{\Sigma_{p\in l} x_p(t) + (q_l(t) - q_0)/T}{C_l}) \qquad (2)$$

where β is a constant parameter, q_0 is the expected queue length in steady state, T is the time constant. The term $\Sigma_{p\in l} x_p(t)$ denotes the whole load on link l. In order to control the queue length, we treat the queueing packets as a special species which also consumes a part of the bandwidth resources. Then the available bandwidth should be equal to the bottleneck bandwidth minus the aggregate traffic and the queuing traffic in the router buffer. The term $(1-\frac{\Sigma_{p\in l} x_p(t)+(q_l(t)-q_0)/T}{C_l})$ is used to represent the normalized available capacity.

Therefore when the available capacity is sufficient, r_l grows quickly, otherwise the available bandwidth is deficient, and r_l grows slowly. When the available capacity is consumed completely, r_l achieves equilibrium, at this time q_l equals q_0, and $\Sigma_{p\in l} x_p(t)$ equals C_l exactly. The main task of the link algorithm is to enable the pre-assignment rate factor that each link maintains to respond quickly to the instantaneous load and queue length. Besides the link control algorithm is independent of the per-flow state, and is only decided by the aggregate effect of flows passing through the link.

B. Source Algorithm

Consider the system of differential equations:

$$\dot{x}_p(t) = \alpha x_p(t)(\ln r_p(t) - \ln x_p(t)) \qquad (3)$$

where

$$r_p(t) = \min\{r_l(t) | l \in p\} \qquad (4)$$

α is a constant parameter. In the real network, the path p often contains multiple links. Because all link l maintains pre-assignment rate factor r_l, in order to obtain the most congested node in the network, we can only choose the minimum value r_p among all pre-assignment rate factors.

For the flow p, the r_p value is the maximum capacity that the network can provide. Generally flow p enters the network with low initial rate, and the pre-assignment rate factor r_p received by the end-system will be larger than x_p, so that x_p exponentially approaches r_p quickly according to equation (3). When x_p equals r_p, the end system reaches equilibrium. We use the logarithm function in the source algorithm to keep the time for convergence to fairness as $O(lnlnP)$. This conclusion is proved in Section IV.

In general, the link algorithm and the source algorithm have the intrinsic rate of increase and the density-dependent factor that are similar to those of the Logistic Model. And the key control variable x and r in the link algorithm and source algorithm are coupled. So we call the whole control model consists of (2) (3) and (4) the Coupling Logistic Model, and the corresponding transport protocol the Coupling Logistic TCP(CLTCP).

XCP uses the direct bandwidth allocation method in the router to get the target assignment instantly and it requires the communication of congestion window in the packet header, as well as the RTT signal. This makes the network more vulnerable to router attack. Different from XCP, CLTCP adopts an exploratory assignment strategy in the router, and continuously adjusts the r factor without any auxiliary information from the end system. We name this mechanism bandwidth pre-assignment. r and x achieve the final target assignment through joint evolution of the link algorithm and the source algorithm.

Another equation of the flow rate and the queue length is shown by the fluid-flow queuing model [18]. For the bottleneck

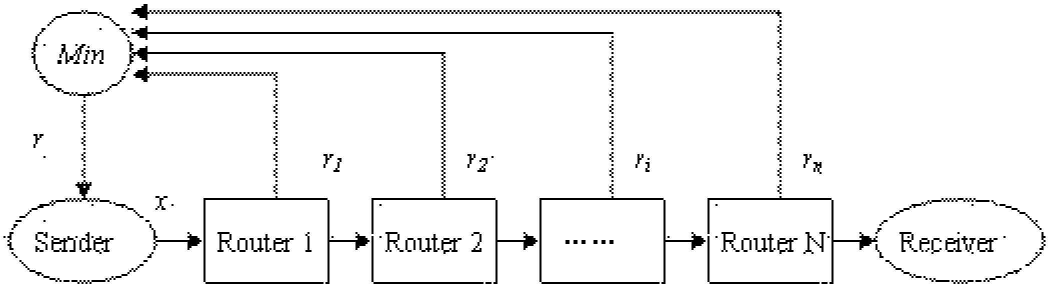


Fig. 2. Principle of bandwidth pre-assignment mechanism

link, given the aggregate arrival rate and link bandwidth, we can calculate instantaneous queue length $q_l(t)$ from:

$$\dot{q}_l(t) = \Sigma_{p\in l} x_p(t) - C_l \qquad (if \;\; q_l(t) > 0) \tag{5}$$

This equation shows that a queue will build up when the aggregate arrival rate exceeds the link bandwidth.

Notice that the pre-assignment rate factor perceived by the sender also has a time delay τ_p, i.e. $\gamma_p = \gamma_p(t-\tau_p)$, therefore the whole control model of the network may be expressed as:

$$\begin{cases} \dot{x}_p(t) = \alpha x_p(t) \cdot (\ln \gamma_p(t-\tau_p) - \ln x_p(t)) \\ \dot{\gamma}_l(t) = \beta \gamma_l(t) \cdot (1 - \frac{\Sigma_{p\in l} x_p(t) + (q_l(t) - q_0)/T}{C_l}) \\ \dot{q}_l(t) = \Sigma_{p\in l} x_p(t) - C_l \qquad (if \;\; q_l(t) > 0) \\ \gamma_p(t) = \min\{\gamma_l(t) | l \in p\} \end{cases} \tag{6}$$

IV. PERFORMANCE ANALYSIS

In this section we establish the global asymptotical stability of the system described by differential equations (6) and determine the time to convergence. We analyze the impact of control parameters on CLTCP's performance, and provide a guideline to determine the appropriate parameters of CLTCP.

A. Stability

To prove that CLTCP can achieve a fair and stable coexisting state, we derived the following theorem:

Theorem 1: The system described by differential equations (6) is globally asymptotically stable independent of the bottleneck capacity, the number of flows and the round trip time.

Proof: Suppose $u_p(t) = \ln x_p(t)$, $v_l(t) = \ln \gamma_l(t)$,and then the equations (6) can be rewritten as:

$$\begin{cases} \dot{u}_p(t) = \alpha \cdot (v_p(t-\tau_p) - u_p(t)) \\ \dot{v}_l(t) = \beta \cdot (1 - \frac{\Sigma_{p\in l} \exp\{u_p(t)\} + (q_l(t) - q_0)/T}{C_l}) \\ \dot{q}_l(t) = \Sigma_{p\in l} \exp\{u_p(t)\} - C_l \qquad (if \;\; q_l(t) > 0) \\ v_p(t) = \min\{v_l(t) | l \in p\} \end{cases} \tag{7}$$

Since $v_l(t)$ is computed by $u_p(t)$ and $q_l(t)$, and $q_l(t)$ is also computed by $u_p(t)$, we define a mapping function f from $u_p(t)$ to $v_l(t)$ which satisfies:

$$\dot{v}_l(t) = \beta f(u_p(t)) \tag{8}$$

Therefore, the equations (7) may be simplified as:

$$\begin{cases} \dot{u}_p(t) = \alpha \cdot (v_p(t-\tau_p) - u_p(t)) \\ \dot{v}_l(t) = \beta f(u_p(t)) \\ v_p(t) = \min\{v_l(t) | l \in p\} \end{cases} \tag{9}$$

Next we use Lyapunov Stability Theory to prove the global asymptotical stability of CLTCP. Firstly, we construct the following positive definite function:

$$\begin{aligned} \mathcal{U}(u_p(t), v_l(t) | p \in P, l \in L) = \beta \cdot \Sigma_{l\in L} \int_0^{-f(u_p(t))} y dy \\ + \alpha \cdot \Sigma_{p\in P} \frac{(v_p(t-\tau_p) - u_p(t))^2}{2} \end{aligned} \tag{10}$$

Observe that:

$$\frac{\partial}{\partial u_p(t)}\mathcal{U} \quad = -\alpha(v_p(t-\tau_p) - u_p(t)) - \beta f(u_p(t)) \tag{11}$$

$$\frac{\partial}{\partial v_l(t)}\mathcal{U} \quad |_{v_l(t)=v_p(t)} = \alpha(v_p(t-\tau_p) - u_p(t)) \tag{12}$$

$$\frac{\partial}{\partial v_l(t)}\mathcal{U} \quad |_{v_l(t)\neq v_p(t)} = 0 \tag{13}$$

Furthermore,

$$\begin{aligned} \frac{d}{dt}\mathcal{U} &= \Sigma_{p\in P} \frac{\partial}{\partial u_p(t)}\mathcal{U} \cdot \frac{d}{dt}u_p(t) + \Sigma_{l\in L} \frac{\partial}{\partial v_l(t)}\mathcal{U} \cdot \frac{d}{dt}v_l(t) \\ &= \Sigma_{p\in P}[-\alpha(v_p(t-\tau_p) - u_p(t)) - \beta f(u_p(t))] \cdot \\ &\quad [\alpha \cdot (v_p(t-\tau_p) - u_p(t))] \\ &\quad + \Sigma_{p\in P} \alpha\beta(v_p(t-\tau_p) - u_p(t)) \cdot f(u_p(t)) \\ &= -[\alpha(v_p(t-\tau_p) - u_p(t)]^2 \end{aligned} \tag{14}$$

Clearly, the function $\dot{\mathcal{U}}$ is negative by definition. Thus function $\mathcal{U}$ is a Lyapunov function for the system of differential equations (7). According to Lyapunov stability theory, the system is globally asymptotically stable. ■

Next consider that N long-lived flows share the bottleneck capacity C, and suppose the equilibrium point of differential equations (6) is $M(x_1^*, x_2^*, ..., x_N^*, \gamma^*, q^*)$. Then we have,

$$\begin{cases} \alpha x_i^* \cdot (\ln \gamma^* - \ln x_i^*) = 0 \\ \beta \gamma^* \cdot (1 - N x_i^* + \frac{(q^* - q_0)/T}{C}) = 0 \\ N x_i^* - C = 0 \end{cases} \tag{15}$$

Since $x_p^* = 0$ is not a stable point, we have,

$$\begin{cases} x_i^* = r^* = \frac{C}{N} \\ q^* = q_0 \end{cases} \quad (16)$$

In steady state, each flow gets the same rate and the queue length equals q_0. Namely, CLTCP guarantees reasonable fairness and full link utilization.

B. Convergence

In this section, we show that CLTCP converges to efficiency and fairness exponentially. Since it is hard to solve the differential equations (6) in complex topology networks, we ignore the influence of delay and only consider the network in which there are N long-flows and a single bottleneck link.

1) Convergence to efficiency: To better understand the time CLTCP requires to reach a certain level of efficiency, we define:

Definition 1: For a given positive constant $\theta(0 < \theta \leq 1)$ and bottleneck link with finite capacity C, a resource allocation $(x_1, x_2, ..., x_N)$ is θ *efficiency*, if:

$$f(t) = \frac{\sum_{i=1}^{N} x_i(t)}{C} \geq \theta \quad (17)$$

Thus the time for convergence to efficiency is the interval that the link utilization increases from the minimal utilization to θ first, i.e. $f(t_\theta) = \theta$.

Based on this definition, we derive the following theorem:

Theorem 2: Consider N synchronous CLTCP flows starting to compete for the bottleneck bandwidth C with the initial throughput of each flow as $x_0(x_0 << C/N)$. Then the time for convergence to efficiency satisfies the following equations:

$$max(\frac{\ln(\frac{C}{Nx_0} \cdot \frac{\theta}{1-\theta})}{\beta}, \frac{\ln(\frac{\ln \frac{Nx_0}{C}}{\ln \theta})}{\alpha}) < t_\theta < \frac{\ln(\frac{C}{Nx_0} \cdot \frac{\theta}{1-\theta})}{\beta} + \frac{\ln(\frac{\ln \frac{Nx_0}{C}}{\ln \theta})}{\alpha} \quad (18)$$

Proof: Since there is no persistent packet queuing before the utilization reaches θ, $q(t) = 0$. Considering all flows as synchronous, the system can be simplified as:

$$\begin{cases} \dot{x}_i(t) = \alpha x_i(t) \cdot (\ln r(t) - \ln x_i(t)) \\ \dot{r}(t) = \beta r_l(t) \cdot (1 - \frac{Nx_i(t) - q_0/T}{C}) \end{cases} \quad (19)$$

Consider two extreme cases: (1) Let $x_i(t)$ equal $r(t)$ directly in the end system and $r(t)$ be adjusted based on the link algorithm, and suppose the time for convergence to efficiency is t_1 in this case; (2) Keeping $r(t)$ equal to C/N, and $x_i(t)$ adjusted based on the source algorithm, suppose the time for convergence to efficiency is t_2 in this case. Certainly we have $max(t_1, t_2) < t_\theta < t_1 + t_2$.

In the first case, the control law can be expressed as:

$$\begin{cases} x_i(t) = r(t) \\ \dot{r}(t) = \beta r_l(t) \cdot (1 - \frac{Nx_i(t) - q_0/T}{C}) \end{cases} \quad (20)$$

Furthermore,

$$\dot{x}_i(t) = \beta x_i(t) \cdot (1 - \frac{Nx_i(t) - q_0/T}{C}) \quad (21)$$

Solve the above equation and we have,

$$x(t) = \frac{\frac{C(1-q_0/CT)}{N}}{1 + (\frac{C(1-q_0/CT)}{Nx_0} - 1) \exp\{-b(1 - q_0/CT)t\}} \quad (22)$$

In high speed networks, there is $C >> q_0/T$, $C >> Nx_0$ thus,

$$x(t) \approx \frac{\frac{C}{N}}{1 + \frac{C}{Nx_0} \exp\{-bt\}} \quad (23)$$

Based on Definition 1, $f(t_1) = Nx_i(t_1)/C = \theta$. Therefore we can solve t_1 as follows:

$$\begin{aligned} t_1 &= \frac{\ln[(\frac{C}{Nx_0} - 1) \cdot (\frac{\theta}{1-\theta})]}{\beta} \\ &\approx \frac{\ln(\frac{C}{Nx_0} \cdot \frac{\theta}{1-\theta})}{\beta} \end{aligned} \quad (24)$$

In the second case, the control law can be expressed as:

$$\dot{x}_i(t) = \alpha x_i(t)(\ln \frac{C}{N} - \ln x_i(t)) \quad (25)$$

Since $x(t_2) = \theta C/N$, we can solve t_2 as follows:

$$t_2 = \frac{\ln(\frac{\ln \frac{Nx_0}{C}}{\ln \theta})}{\alpha} \quad (26)$$

Finally,

$$max(\frac{\ln(\frac{C}{Nx_0} \cdot \frac{\theta}{1-\theta})}{\beta}, \frac{\ln(\frac{\ln \frac{Nx_0}{C}}{\ln \theta})}{\alpha}) < t_\theta < \frac{\ln(\frac{C}{Nx_0} \cdot \frac{\theta}{1-\theta})}{\beta} + \frac{\ln(\frac{\ln \frac{Nx_0}{C}}{\ln \theta})}{\alpha} \quad (27)$$

■

Clearly, the time for CLTCP convergence to efficiency is $O(\ln \frac{C}{N})$ and it is exponentially fast.

2) Convergence to fairness: To study the time CLTCP requires to reach a certain level of fairness, we define:

Definition 2: For a given positive constant $\varepsilon(0 < \varepsilon \leq 1)$, a resource allocation $(x_1, x_2, ..., x_N)$ exhibits ε *fairness*, if:

$$g(t) = \frac{min_{i=1}^{N} x_i(t)}{max_{j=1}^{N} x_j(t)} \geq \varepsilon \quad (28)$$

Thus the time for convergence to fairness is the interval between when g increases from the maximally unfair state to ε *fairness*. i.e. $g(t_\varepsilon) = \varepsilon$.

Based on this definition, we derive the following theorem:

Theorem 3: Consider N synchronous CLTCP flows have completely shared the bottleneck capacity C at steady state, and suppose a new flow enters into the network with initial throughput $x_0(x_0 << C/N)$, then after

$$t_\varepsilon = \frac{\ln \ln \frac{C\varepsilon}{Nx_0}}{\alpha} \tag{29}$$

the network achieves ε *fairness*.

Proof: Suppose $x_{N+1}(t)$ denotes the new joining flow, and $x_j(t)(j = 1, 2, ..., N)$ denotes one flow of the existing N synchronous flows. And then the system can be described by:

$$\begin{cases} \dot{x}_{N+1}(t) = \alpha x_{N+1}(t) \cdot (\ln r(t) - \ln x_{N+1}(t)) \\ \dot{x}_j(t) = \alpha x_j(t) \cdot (\ln r(t) - \ln x_j(t)) \end{cases} \tag{30}$$

Based on Definition 2, $g(t) = x_j(t)/x_{N+1}(t)$. And then we derive:

$$\dot{g(t)} = -\alpha g(t) \ln g(t) \tag{31}$$

thus, we have

$$\ln \dot{\ln g}(t) = -\alpha \tag{32}$$

Consider the initial condition $g(0) = \frac{C/N}{x_0} = \frac{C}{Nx_0}$, $g(t_\varepsilon) = \frac{1}{\varepsilon}$. We have:

$$t_\varepsilon = \frac{\ln \ln \frac{C\varepsilon}{Nx_0}}{\alpha} \tag{33}$$

■

Clearly, the time for CLTCP convergence to fairness is $O(\ln \ln \frac{C}{N})$ and it is also exponentially fast.

C. Setting the parameters

In this section, we discuss the choice of parameters used by CLTCP and we implement the coupling logistic model using the linearizing method of control theory [14]. In order to build the control system model of the CLTCP transport system, we only consider the network in which there are N synchronous long-flows and a single bottleneck link and omit the influence of delay.

Let the rate of each flow be $x(t)$, the pre-assignment rate factor be $r(t)$, and the queue length be $q(t)$. For simplicity, we use x, r, q to denote $x(t)$, $r(t)$ and $q(t)$. Suppose

$$\begin{cases} F(x, r, q) = \alpha x \cdot (\ln r - \ln x) \\ G(x, r, q) = \beta r \cdot (1 - \frac{Nx + (q - q_0)/T}{C}) \\ H(x, r, q) = Nx - C \end{cases} \tag{34}$$

Suppose the equilibrium point is M(x^*, r^*, q^*),where $x^* = r^* = C/N$, $q^* = q_0$, thus we linearize the equations (34) at point M, and we have

$$\begin{aligned} \frac{\partial F}{\partial x}|_{x=x^*} &= -\alpha, \ \frac{\partial F}{\partial r}|_{r=r^*} = \alpha, \frac{\partial F}{\partial q}|_{q=q^*} = 0 \\ \frac{\partial G}{\partial x}|_{x=x^*} &= -\beta, \ \frac{\partial G}{\partial r}|_{r=r^*} = 0, \frac{\partial G}{\partial q}|_{q=q^*} = -\frac{\beta}{NT} \\ \frac{\partial H}{\partial x}|_{x=x^*} &= N, \ \frac{\partial H}{\partial r}|_{r=r^*} = 0, \frac{\partial H}{\partial q}|_{q=q^*} = 0 \end{aligned} \tag{35}$$

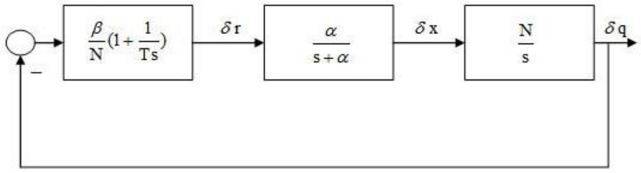


Fig. 3. Block of close-loop control system

Suppose $\delta x = x - x^*, \delta r = r - r^*, \delta q = q - q^*$. Then we have:

$$\begin{cases} \delta \dot{x} = \alpha \delta r - \alpha \delta x \\ \delta \dot{r} = -\beta \delta x - \frac{\beta}{NT} \delta q \\ \delta \dot{q} = N \delta x \end{cases} \tag{36}$$

Furthermore,

$$\begin{cases} \delta \dot{x} = \alpha \delta r - \alpha \delta x \\ \delta \dot{r} = -\frac{\beta}{N} \delta \dot{q} - \frac{\beta}{NT} \delta q \\ \delta \dot{q} = N \delta x \end{cases} \tag{37}$$

The block of the closed-loop control system is shown in Fig.3. From Fig.3, we see that the link algorithm in fact is a PI controller component in which the proportional constant is β/N, and the integral constant is T, the source algorithm in fact is a inertia controller in which the time constant is $1/\alpha$, and the queue model is a integral controller. Therefore we can represent the following open-loop transfer function of the CLTCP transport system as:

$$\begin{aligned} L(s) &= \frac{\alpha}{s + \alpha} \cdot \frac{\beta}{N}(1 + \frac{1}{Ts}) \cdot \frac{N}{s} \\ &= \frac{\alpha\beta(s + \frac{1}{T})}{s^2(s + \alpha)} \end{aligned} \tag{38}$$

In control theory, if the transfer function of an unadjusted system is

$$L_0(s) = \frac{K_0}{s(1 + T_0 s)} \tag{39}$$

then it can be adjusted by a PI controller whose transfer function is

$$L_c(s) = \frac{1 + \tau s}{T_i s} \tag{40}$$

to improve the performance of the system. Thus the final transfer function of the system is

$$L(s) = L_0(s) L_c(s) = \frac{K_0}{T_i} \frac{1 + \tau s}{s^2(1 + T_0 s)} \tag{41}$$

In order to obtain the maximum stability margin and fast convergence speed as much as possible, the parameters in the PI controller takes

$$\tau = 4T_0, \quad T_i = 8K_0T_0^2 \tag{42}$$

and is better in general [8].

Comparing equation (38) to equation (41) and (42), we have

$$\beta = \alpha/2, \quad T = \frac{4}{\alpha} \tag{43}$$

From the analysis of convergence, we can see that the larger the α is, the faster the convergence to efficiency and fairness. However we observed from simulations that the traffic also becomes more volatile when α becomes large. To balance the convergence and oscillation, we set $\alpha = 2sec^{-1}$. Then we have $\beta = 1sec^{-1}$, $T = 2sec$.

V. Implementation

Because computer control is just one kind of sampling control, the differential equations (6) cannot be used directly. It is necessary to derive the discrete time equations. Suppose T_1 and T_2 are the sampling time in the end system and router respectively, and we use the discrete time, kT_1 and kT_2, to represent time t. Then we can make the following approximate transformation:

$$\begin{cases} \frac{d\ln(x(t))}{dt} \approx \frac{\ln x((k+1)T_1) - \ln x(kT_1)}{T_1} = \frac{\ln x(k+1) - \ln x(k)}{T_1} \\ \frac{d\ln(r(t))}{dt} \approx \frac{\ln r((k+1)T_2) - \ln r(kT_2)}{T_1} = \frac{\ln r(k+1) - \ln r(k)}{T_2} \end{cases} \tag{44}$$

where we use $x(k)$ and $r(k)$ to denote $x(kT_1)$ and $r(kT_2)$ respectively and omit the suffixes of the variables. Then we substitute the above equation into equation (6) and obtain:

$$\begin{cases} \frac{\ln x(k+1) - \ln x(k)}{T_1} = \alpha \cdot (\ln r(k - \frac{\tau}{T_1}) - \ln x(k)) \\ \frac{\ln r(k+1) - \ln r(k)}{T_2} = \beta \cdot (1 - \frac{\Sigma x(k) + (q(k) - q_0)/T}{C}) \end{cases} \tag{45}$$

Further we have,

$$\begin{cases} x(k+1) = x(k)^{1-\alpha T_1} \cdot r(k - \frac{\tau}{T_1})^{\alpha T_1} \\ r(k+1) = r(k) \cdot \exp\{\beta T_2 \cdot (1 - \frac{\Sigma x(k) + (q(k) - q_0)/T}{C})\} \end{cases} \tag{46}$$

In the implementation, we finally select $T_1 = \tau$ and $T_2 = 0.1s$ to guarantee the sampling precision. At the same time we let q_0 be 100 packets which is similar to some AQM algorithms [13] to provide small queuing delay in high speed networks. Substituting the value of all parameters into the equation (49), we have the final congestion controller:

$$\begin{cases} x(k+1) = x(k)^{1-2\tau} \cdot r(k-1)^{2\tau} \\ r(k+1) = r(k) \cdot \exp\{0.1(1 - \frac{\Sigma x(k) + (q(k) - 100)/2}{C})\} \end{cases} \tag{47}$$

From the above equation, we see that the complex logarithm computation is excluded from the practical congestion control algorithm so that the computational overhead is reduced. Meanwhile, the small sampling time in the router, 0.1s, also reduces the router computational overhead compared with XCP, which is based on per-packet computation.

A. Packet Header

It is easier to implement the CLTCP algorithm with the extension CLTCP header. As shown in Fig.4, the CLTCP header including a pre-assignment rate factor field is inserted into the position between the IP header and the TCP header. The routers along the route modify the pre-assignment rate factor field to directly control the sending rate of the sender.

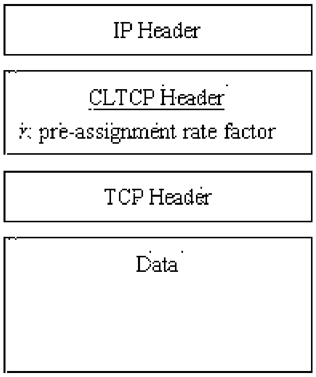


Fig. 4. CLTCP packet header

B. Sender

On packet departures, the CLTCP sender initializes the pre-assignment rate factor to -1. Whenever a new ACK packet arrives, the sender reads the pre-assignment rate factor $r(k-1)$ from the ACK packet, and tracks the RTT τ. As a result the sender adjusts the sending rate as follows:

$$x(k+1) = x(k)^{1-2\tau} \cdot r(k-1)^{2\tau} \tag{48}$$

C. Router

The main task of the router is to generate its pre-assignment rate factor and insert it into the headers of all passing packets. During the sampling interval, the router tracks the total amount of data that has arrived into the queue. At each sampling point, the router tracks the instantaneous queue length and computes the average incoming traffic rate $\Sigma x(k)$. Based on this information, the router computes an estimate of the pre-assignment rate factor as follows:

$$r(k+1) = r(k) \cdot \exp\{0.1(1 - \frac{\Sigma x(k) + (q(k) - 100)/2}{C})\} \tag{49}$$

In addition, the router examines whether its locally recorded pre-assignment rate factor is smaller than the one carried in the packet. If so, the router replaces the corresponding field in the packet. In this manner, after traversing the whole path, each packet obtains the pre-assignment rate factor from the most congested link.

D. Receiver

A CLTCP receiver is similar to a TCP receiver except that when acknowledging a packet, it copies the extension CLTCP header from the data packet to its acknowledgment packet.

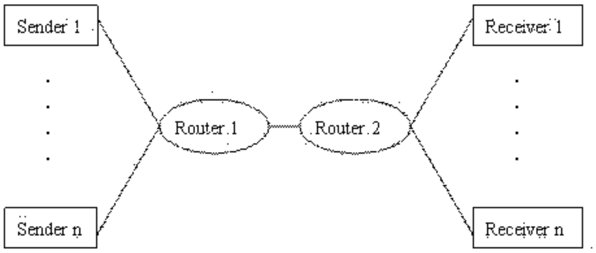
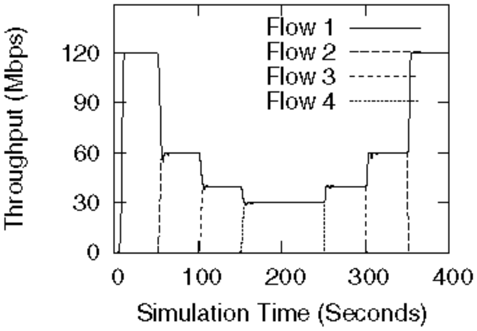
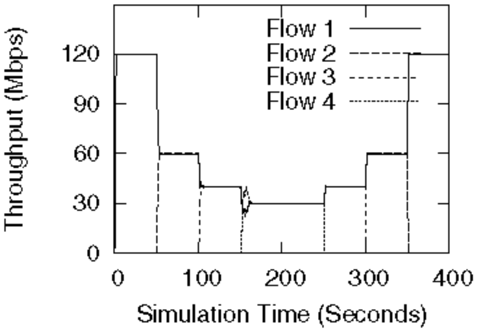


Fig. 5. Dumbbell topology



(a) CLTCP.



(b) XCP.

Fig. 6. The rate dynamics of four flows with CLTCP and XCP .

VI. SIMULATION RESULTS

In this section, we present some simulation results of the performance of CLTCP. As a comparison we choose to test XCP and CLTCP and We use ns2 for the simulation experiments. We deploy a tail-drop discipline at the router buffer, and the buffer size is set to 10Mbytes. In all experiments the data packet size is 1000 bytes, while the ACK packet size is 40 bytes. For all the graphs, rate, utilization, packet loss rate and queue length are sampled over 1s intervals.

A. Convergence

In this experiment, we evaluate the performance of CLTCP and XCP for the simple case of a single bottleneck link shared by multiple flows. The dumbbell topology used here is depicted in Fig.5. It consists of source/destination hosts, two routers, and links between the hosts and routers. We run four flows with different RTT of 50ms, 100ms, 200ms and 400ms, while the bottleneck bandwidth is 120Mbps. These flows start at 0s, 50s, 100s and 150s and stop at 400s, 350s, 300s, 250s respectively. The rate curves are drawn in Fig.6.

As shown in Fig.6, the convergence speed of CLTCP is as fast as XCP whenever converging to efficiency or converging to fairness. At the same time, CLTCP and XCP both achieve full utilization and a zero packet loss rate. In addition, CLTCP and XCP both achieve max-min fairness independent of RTT.

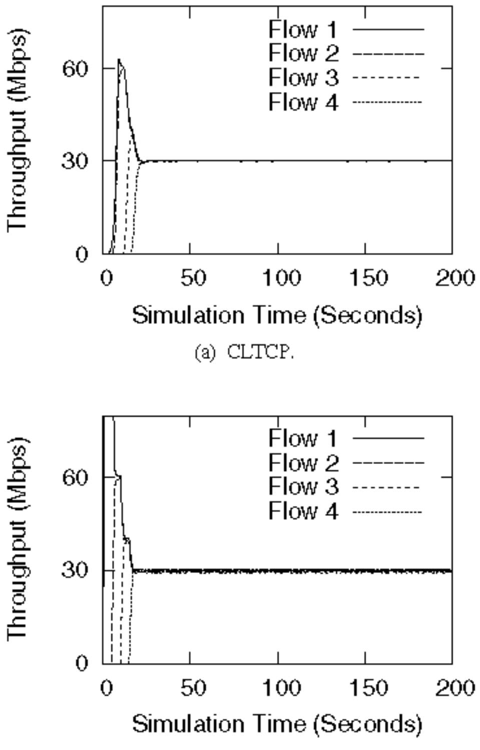
B. Stability

This experiment shows the stability of CLTCP in the presence of web traffic, burst traffic and reverse traffic. The dumbbell topology is used here, where the bottleneck bandwidth is 120Mbps and round trip propagation delay is 50ms. For comparison purposes, two simulations are conducted.

In the first simulation, there are only four high speed flows on the forward path without disturbing traffic. In the second simulation, in addition to the four high speed flows as above, there are another four high speed flows on the backward path, and the average web traffic of 20Mbps generated by 100 random on-off sources is always on. These web flows arrive according to the Poisson process. Moreover, burst CBR traffic of 40Mbps generated by 10 UDP sources is injected into the network at 100s, and then all UDP sources drop out at 150s.

The rate dynamic curves of the first simulation and the second simulation are drawn in Fig.7 and Fig.8 respectively. The average utilization, average packet loss rate and average queue length of the bottleneck link of the first simulation and the second simulation are listed in table I.

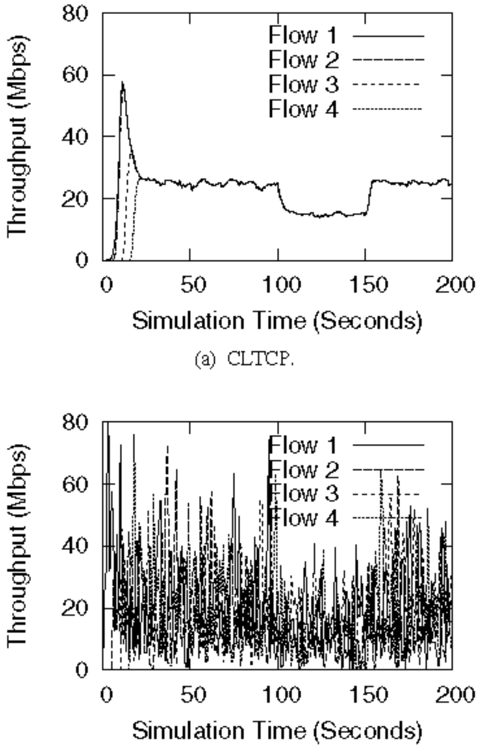
As shown in Fig.7 and the S1 columns in table I, CLTCP and XCP converge to an equal rate in steady state, and the average utilizations of the bottleneck link are very high in the absence of disturbing traffic. In addition, the average queue length of CLTCP approaches the expected queue length q_0. However, Fig.8 and the S2 columns in table I show that XCP oscillates acutely in the presence of disturbing traffic. Moreover, the average utilizations of XCP degrade badly, and the average loss rates increase obviously. Fig.8 shows that the CLTCP flows can converge to an equilibrium rate of 25Mbps even with web traffic. When burst traffic appears at 100s, the CLTCP flows give up the bandwidth rapidly. At 105s, the CLTCP flows converge to a new equilibrium rate(15Mbps). After the burst traffic leaves at 150s, the CLTCP flows catch the available bandwidth and converge to the previous equilibrium rate quickly. Table I also shows that CLTCP achieves a higher link utilization and lower packet loss rate than XCP even in the presence of disturbing traffic.



(a) CLTCP.

(b) XCP.

Fig. 7. The rate dynamics of ten flows using CLTCP and XCP in the absence of web traffic, burst traffic and reverse traffic.



(a) CLTCP.

(b) XCP.

Fig. 8. The rate dynamics of ten flows using CLTCP and XCP in the presence of web traffic, burst traffic and reverse traffic.

TABLE I

THE AVERAGE UTILIZATION, AVERAGE PACKET LOSS RATE AND AVERAGE QUEUE LENGTH

Protocol	utilization(%)		packet loss ratio		queue length	
	S1	S2	S1	S2	S1	S2
CLTCP	100	98.63	0	7.59×10^{-3}	108	393
XCP	99.94	86.28	0	3.36×10^{-2}	0.8	485

S1: The first simulation in the absence of disturbing traffic
S2: The second simulation in the presence of disturbing traffic

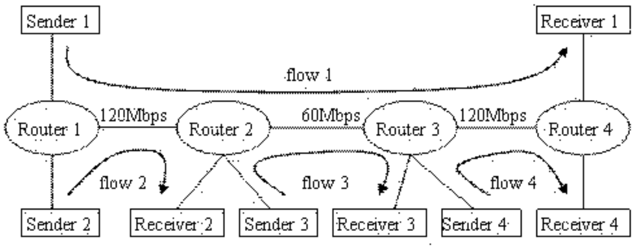


Fig. 9. Parking-lot topology

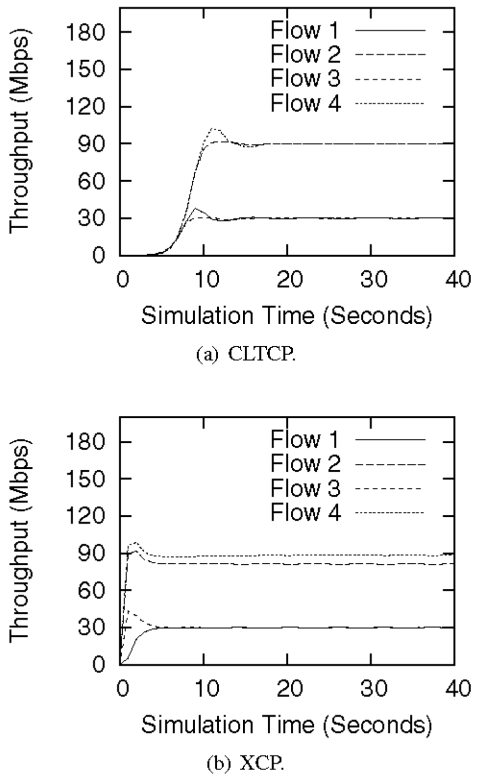
C. Multiple Bottlenecks

Next, we study the performance of CLTCP with a more complex topology of multiple bottlenecks. For this purpose, we use a typical parking-lot topology with three links depicted in Fig.9. All the links have a 20ms one-way propagation delay. There is one high speed flow(flow 1) traversing all the links in the forward direction. In addition, each individual link has one crossing high speed flow(flow 2, flow 3, and flow 4) traversing in the forward direction. The middle link has the smallest bandwidth of only 60Mbps, and the other links have the same bandwidth of 120Mbps. All flows start at time zero.

As shown in Fig.10, in CLTCP, the throughput of flow 1 and flow 3 both achieve 30Mbps and the throughput of flow 2 and flow 4 both achieve 90Mbps in steady state. In XCP, the throughput of flow 1 and flow 3 both achieve 30Mbps , but the throughput of flow 2 only achieves 81Mbps and the throughput of flow 4 only achieves 87Mbps in steady state. Namely, CLTCP completely achieves max-min fairness while XCP approximately achieves.

VII. CONCLUSION

In this paper we developed a novel congestion control model consisting of the link algorithm and the source algorithm for high speed networks based on the logistic model. The key mechanism is based on bandwidth pre-assignment which is similar to XCP and MaxNet. The pre-assignment rate factor is computed based on the information of the router capacity, the aggregate incoming traffic and the queue length, and then senders adjust the sending rate based on the pre-assignment rate factor to strengthen the convergence and stability of transport protocol. We also discuss the convergence and stability through theoretical analysis. The performance of this algorithm is shown via simulation in terms of convergence,



(a) CLTCP.

(b) XCP.

Fig. 10. The rate dynamics of four flows using CLTCP and XCP in a multiple bottleneck topology.

stability, fairness, queue length, link utilization, and packet loss ratio. We show that CLTCP can provide fast convergence and strong stability, as well as high utilization and fair bandwidth allocation, all of which are desirable for high speed networks. In particular, it can reduce the computational overhead in routers compared to XCP and yet achieve better performance.

VIII. Acknowledgments

This work is supported in part by a grant from the National Grand Fundamental Research 973 Program of China (No. 2003CB314804, 2004CB318000, 2003CB317007, 2006CB303000), and the National Natural Science Foundation of China (No. 60573122, 90412012,60773138).

References

[1] M. Allman and W. Stevens. TCP congestion control. *RFC 2581*.

[2] S. Deb and R. Srikant. Global stability of congestion controllers for the internet. In *2002 IEEE Conference on Decision and Control*, 2002.

[3] W. Feng, K. Shin, D. Kandlur, and D. Saha. The blue active queue management algorithms. *IEEE/ACM Transactions on Networking*, 10(4):513–528, 2002.

[4] S. Floyd. High-speed TCP for Large Congestion Windows. *RFC 3649*.

[5] S. Floyd and V. Jacobson. Random early detection gateways for congestion avoidance. *IEEE Transactions on Networking*, 1(4):397–413, 1993.

[6] Y. Gao and J. Hou. A state feedback control approach to stabilizing queues for ecn-enabled tcp connecitons. In *INFOCOM'03*, 2003.

[7] C. V. Hollot, V. Misra, D. F. Towsley, and W. Gong. A control theoretic analysis of RED. In *INFOCOM'01*, pages 1510–1519, 2001.

[8] S. Hu. *Principle of automatic control(In Chinese). 4th edition.* Science Publisher, 2001.

[9] X. Huang, C. Lin, and F. Ren. A novel high speed transport protocol based on explicit virtual load feedback. *Computer Networks*, 51(7):1800–1814, 2007.

[10] V. Jacobson. Congestion Avoidance and Control. *Computer Communication Review*, 18(4):314–329, August 1988.

[11] D. Katabi, M. Handley, and C. Rohrs. Congestion Control for High Bandwidth Delay Product Networks. In *ACM SIGCOMM'02*, August 2002.

[12] T. Kelly. Scalable TCP: Improving Performance in High-speed Wide Area Networks. In *the 1st Workshop on Protocols for Fast Long-Distance Networks*, 2003.

[13] S. Kunniyur and R. Srikant. An adaptive virtual queue (avq) algorithm for active queue management. *IEEE/ACM Transactions on Networking*, 12(2):286–299, 2004.

[14] B. C. Kuo and F. Golnaraghi. *Automatic control systems. 8th edition.* Wiley, New York, 2003.

[15] D. Loguinov and H. Radha. End-to-End Rate-Based Congestion Control: Convergence Properties and Salability Analysis. *IEEE/ACM Transactions on Networking*, 11(5):564C577, August 2003.

[16] A. J. Lotka. *Elements of physical biology*. Baltimore: Williams and Wilkins Co., 1925.

[17] L. Massoulie. Stability of distributed congestion control with heterogeneous feedback delays. *Microsoft Research Technical Report. http://citeseer.ist.psu.edu/massoulie00stability.html*.

[18] V. Misra, W.-B. Gong, and D. F. Towsley. Fluid-based analysis of a network of AQM routers supporting TCP flows with an application to RED. In *ACM SIGCOMM'00*, pages 151–160, 2000.

[19] P. F. Verhulst. *Recherches math matiques sur la loi d'accroissement de la population.* Nouveaux meacute de l'Academie Royale des Sci. et Belles-Lettres de Bruxelles, 1845.

[20] G. Vinnicombe. On the stability of end-to-end congestion control for the internet. *http://www-control.eng.cam.ac.uk/gv/internet/TR398.pdf*.

[21] M. Welzl. *Scalable Performance Signalling and Congestion Avoidance*. Kluwer Academic Publishers, 2003.

[22] Wydrowski, L. L. Andrew, and M. Zukerman. MaxNet: A Congestion Control Architecture for Scalable Networks. *IEEE Communications Letters*, 7(10):511–513, 2003.

[23] Y. Xia, L. Subramanian, I. Stoica, and S. Kalyanaraman. One More Bit Is Enough. In *ACM SIGCOMM'05*, 2005.

[24] Y. Zhang, S. Kang, and D. Loguinov. Delayed Stability and Performance of Distributed Congestion Control. In *ACM SIGCOMM'04*, 2004.