

Improving Convergence of Congestion Control Algorithm

Abhineet Anand

Center of Information Technology
College of Engineering Studies
Bidholi, Dehradun - 248007

Abhijeet

Senior Research Fellow
National Research Center on Litchi
Muzaffarpur - 842001

D P Das

PG Dept of Physics
BRBA Bihar University
Muzaffarpur - 842001

ABSTRACT

To transmit the transport control system from the initial state to the steady state many convergence of congestion control algorithm has been proposed and studied. Efficiency and fairness of convergence are two aspects major issue which has been targeted by the researcher. The available bandwidth of the link is grabbed by the newly-starting flow, when it joins the network. The various algorithm has been discussed to improve the convergence efficiently. To achieve the convergence up to level of fairness the network where the existing flow have taken the whole bandwidth, it is assured that this new flow should achieve fair bandwidth allocation as soon as possible.

General Terms:

Congestion Control, Wavelength

Keywords:

Computer Network, Wavelength, Congestion Control

1. INTRODUCTION

It is matter of fact that the traditional TCP congestion control is not fully efficient for high speed networks and it is a challenge to design a high speed replacement for TCP. By comparing some existing high speed protocols, it has been seen that these high speed protocols have limitations in convergence and stability. To address these problems, a population ecology model is applied 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.

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.

2. STUDY OF EXISTING ALGORITHM

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 , it is known 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 High speed Transmission Control Protocol (HSTCP) [4], Sensor Transmission Control Protocol (STCP) [12], eXplicit Control Protocol (XCP) [11], Exponential Max-min Kelly Control (EMKC) [24], Variable-structure Congestion control Protocol (VCP) [23], Explicit Virtual Load Feedback - Transmission Control Protocol (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 re-

search, 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 criterion for different transport protocols :

- Fluid-based analysis of a network of AQM routers supporting TCP flows with an application to RED. - By V. Misra, W.-B. Gong, and D. F. Towsley.[18]
- A control theoretic analysis of RED. - By C. V. Hollot, V. Misra, D. F. Towsley, and W. Gong. [7]
- A state feedback control approach to stabilizing queues for ecn-enabled tcp connections - By Y Gao and J. Hou. [6]
- Global stability of congestion controllers for the internet. - By S. Deb and R. Srikant. [2]
- On the stability of end-to-end congestion control for the internet. - By G. Vinnicombe [20]
- Stability of distributed congestion control with heterogeneous feedback delays. - By L. Massoulie. [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 of network parameters to the stability of the transport protocol as much as possible.

This chapter 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] has been discussed. 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.

3. DESIGN BASIS

3.1 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 Additive increase/multiplicative decrease(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. below, 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:

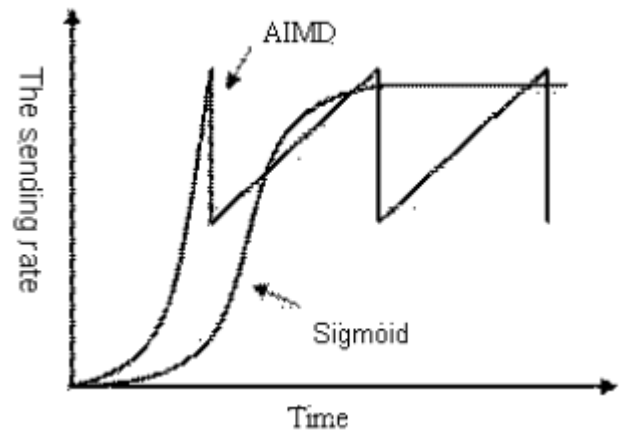


Fig. 1. Comparison of the sigmoid of the control curve with the AIMD Control Curve

- (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.

3.2 Logistic Model

Firstly, the foundation of the Logistic Model is introduced. 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:

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

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

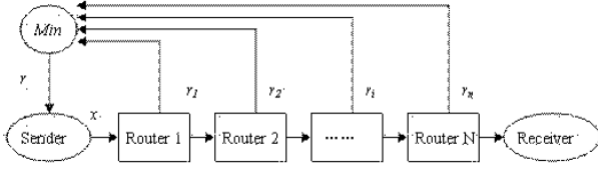


Fig. 2. Principal of bandwidth pre-assignment mechanism

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:

- The intrinsic rate of increase r , and
- 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.

3.3 Algorithm Design

Similar to eXplicit Control Protocol (XCP), our novel mechanism is an explicit bandwidth pre-assignment mechanism. Its principle is shown in Fig.18. 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 MaxNet: A Congestion Control Architecture for Scalable Networks. by Wydrowski, L. L. Andrew, and M. Zukerman. [22]

3.4 Link Algorithm

Consider the system of differential equations:

$$r_l(t) = \beta r_l(t) \left(1 - \frac{\sum_{p \in l} x_p(t) + (q_l(t) - q_0)/T}{C_l} \right) \quad (2)$$

where β is a constant parameter, q_0 is the expected queue length in steady state, T is the time constant. The term $\sum_{p \in l} x_p(t)$ 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 queueing traffic in the router buffer.

The term $(1 - \frac{\sum_{p \in l} x_p(t) + (q_l(t) - q_0)/T}{C_l})$ 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 $\sum_{p \in l} x_p(t)$ 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.

3.5 Source Algorithm

Consider the system of differential equations

$$x_p(t) = \alpha x_p(t) (\ln \gamma_p(t) - \ln x_p(t)) \quad (3)$$

where

$$\gamma_p(t) = \min\{\gamma_l(t) | l \in p\} \quad (4)$$

γ is a constant parameter. In the real network, the path p often contains multiple links. Because all link l maintains preassignment rate factor r_l , in order to obtain the most congested node in the network, it 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. It use the logarithm function in the source algorithm to keep the time for convergence to fairness as $O(\ln \ln P)$.

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, the whole control model consists of (2) (3) and (4) are named as the Coupling Logistic Model, and the corresponding transport protocol are as 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. This mechanism bandwidth are named **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 queueing model [18]. For the bottleneck link, given the aggregate arrival rate and link bandwidth, The instantaneous queue length can be calculated $q_l(t)$ from:

$$q_l(t) = \sum_{p \in l} x_p(t) - C_l \quad (if q_l(t) > 0) \quad (5)$$

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

4. PERFORMANCE ANALYSIS

The global asymptotical stability of the system can be described by differential equations and determine the time to convergence. The impact of control parameters on CLTCP's performance can be analysed, and provide a guideline to determine the appropriate parameters of CLTCP.

4.1 Stability

The system described by differential equations of whole control model of the network is globally asymptotically stable independent of the bottleneck capacity, the number of flows and the round trip time.

Suppose $u_p(t) = \ln x_p(t)$, $v_i(t) = \ln r_i(t)$, and then the differential equations can be rewritten as:

$$\begin{aligned} u_p(t) &= \alpha(u_p(t - \tau_p) - u_p(t)) \\ v_i(t) &= \beta(1 - \frac{\sum_p \exp(v_p(t) + q(t) - q_0)/\tau)}{C_l}) \\ q_i(t) &= \sum_{p \in l} \exp(u_p(t)) - C_l \quad (if q_i(t) > 0) \\ v_p(t) &= \min(v_i(t) | l \in p) \end{aligned} \quad (6)$$

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.

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

$$u_i(t) = \beta f(u_p(t)) \quad (7)$$

Furthermore,

$$\begin{aligned} \frac{d}{dt} u &= \sum_p \epsilon_p \frac{\delta}{\delta u_p(t)} u \cdot \frac{d}{dt} u_p(t) + \sum_{l \in L} \frac{\delta}{\delta u_l(t)} u \cdot \frac{d}{dt} u_l(t) \\ &= \sum_p \epsilon_p [-\alpha(v_p(t - \tau_p) - v_p(t)) - \beta f(v_p(t))] \cdot [\alpha(v(t - \tau_p) - v_p(t))] \\ &\quad + \sum_p \epsilon_p \alpha \beta (v_p(t)) \cdot f(v_p(t)) \\ &= -[\alpha(v_p) - v_p(t)]^2 (8) \end{aligned}$$

Clearly, the function v is negative by definition. Thus function v is a Lyapunov function for the system of differential equations (6). According to Lyapunov stability theory, the system is globally asymptotically stable.

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.

4.2 Convergence

CLTCP converges to efficiency and fairness exponentially. Since it is hard to solve the differential equations in complex topology networks, the influence of delay is ignored and only consider the network in which there are N long-flows and a single bottleneck link.

4.2.1 Convergence to efficiency. To better understand the time CLTCP requires to reach a certain level of efficiency.

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

$$f(t) = \frac{\sum_{i=1}^N x_i(t)}{C} \geq \theta \quad (9)$$

Thus, the time for convergence to efficiency is the interval that the link utilization increases from the minimal utilization to 0 first, i.e. $f(t_\theta) = \theta$. The time for CLTCP convergence to efficiency is $O(\ln \frac{C}{N})$ and it is exponentially fast.

4.2.2 Convergence to fairness. To study the time CLTCP requires to reach a certain level of fairness, it is defined as:

For a given positive constant $\epsilon (0 < \epsilon \leq 1)$, a resource allocation (x_1, x_2, \dots, x_N) exhibits ϵ fairness, if:

$$g(t) = \frac{\min_{i=1}^N x_i(t)}{\max_{j=1}^N x_j(t)} \geq \epsilon \quad (10)$$

Thus the time for convergence to fairness is the interval between when g increases from the maximally unfair state to fairness, i.e. $g(t_\epsilon) = \epsilon$. The time for CLTCP convergence to fairness is $O(\ln \ln \frac{C}{N})$ and it is also exponentially fast.

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

5. IMPLEMENTATION OF CLTCP

Because computer control is just one kind of sampling control, the differential equations cannot be used directly. It is necessary to derive the discrete time equations. It is shown in the implementation 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.

5.1 Packet Header

It is easier to implement the CLTCP algorithm with the extension CLTCP header. As shown in figure 19, 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.

5.2 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.

5.3 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 $\sum x(k)$.

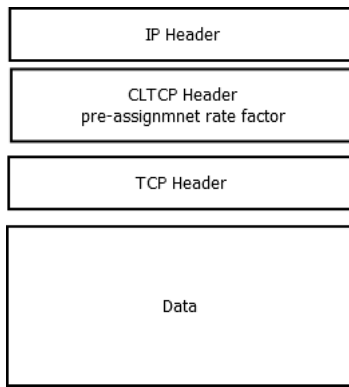


Fig. 3. CLTCP Packet Header

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.

5.4 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.

6. SIMULATION RESULT OF CLTCP IMPLEMENTATION

Simulation Result of the performance of CLTCP is tested in context of XCP. ns2 are used to test results. A tail-drop discipline at the router buffer, and the buffer size is set to 10 Mbytes. In all experiments the data packet size is 1000 bytes, while the ACK packet size is 40 bytes.

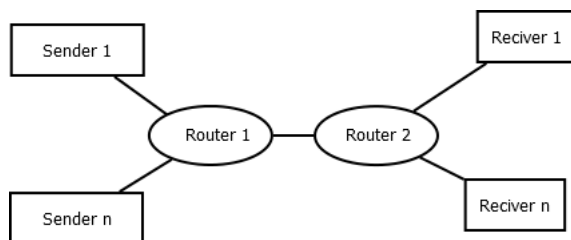


Fig. 4. Dumbbell topology

6.1 Convergence

Here, the evaluation of CLTCP and XCP for the simple case of a single bottleneck link shared by multiple flows. The dumbbell topology used here, which consist of source/destination hosts, two routers, the links between the hosts and routers. 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.

7. CONCLUSION

In this chapter the focus is to find a better 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, stability, fairness, queue length, link utilization, and packet loss ratio. It is to 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.

8. 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 connections. 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 Scalability Analysis. *IEEE/ACM Transactions on Networking*, 11(5):564-577, August 2003.
- [16] A. J. Lotka. *Elements of physical biology*. Baltimore: Williams and Wilkins Co., 1925.
- [17] L. Massoulié. Stability of distributed congestion control with heterogeneous feedback delays. Microsoft Research Technical Report. <http://citeseer.ist.psu.edu/massoulieOOstability.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 mathématiques sur la loi d'accroissement de la population*. Nouveaux meacute de l'Académie 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.