End-to-end Congestion Control for High Speed Networks Based on Population Ecology Models

Xiaomeng Huang, Fengyuan Ren, Guangwen Yang, Yongwei Wu, Weiming Zhen, Chuang Lin
Department of Computer Science and Technology
Tsinghua University, Beijing, 100084, China
{huangxiaomeng,renfy,ygw,wuyw, zwm-dcs,chlin}@tsinghua.edu.cn

Abstract

Since TCP congestion control is ill-suited for high speed networks, designing a replacement for TCP has become a challenge. To address this problem, we extend the population ecology theory to design a novel congestion control algorithm. We treat the network flows as the species in nature, the throughput of the flows as the population number, and the bottleneck bandwidth as the food resources. Then we use the key idea of constructing population ecology models to develop a novel congestion control model, and implement the corresponding end-to-end transport protocol through measurement, which called Population Ecology TCP (PE-TCP). The theoretical analysis and simulation results validate that PE-TCP achieves high utilization, fast convergence, fair bandwidth allocation, and near-zero packet drops. These qualities are desirable for high speed networks.

1 Introduction

With the rapid advances in the deployment of ultra high speed links in the Internet, many recent studies reported that the classical Additive Increase Multiplicative Decrease(AIMD) mechanism[4] used in TCP does not perform well in high speed networks.

This problem motivates the proposal of several new high speed transport protocols, each with their own strengths and limitations. It is possible to classify these protocols into two types: implicit congestion feedback and explicit congestion feedback[15]. Implicit congestion feedback schemes, such as HSTCP[5], STCP[10], BICTCP[18], CUBIC[14], Layered TCP[3], FAST[8] and many more, keep the end-to-end principle, and use the loss or delay as the congestion signal which is incidental to the delivery of the packets. Explicit congestion feedback schemes, such as XCP[9], CADPC[16], EMKC[19] and VCP[17], use direct communication from routers to tell the end hosts the state of the network, accomplished by sending special packets or by changing some fields in packets as they traverse the routers. The use of explicit congestion feedback is likely to result in superior congestion control protocols that converge faster and keep lower packet loss rate than protocols using implicit congestion feedback. However, given a lack of deployment of router features for explicit congestion feedback, a transport protocol relying on implicit congestion feedback is easier to deploy now.

In this paper, we propose a pure end-to-end and measurement based solution, which combines some advantages of implicit congestion feedback and explicit congestion feedback, i.e., high utilization, fast convergence, fair bandwidth allocation, and near-zero packet drops. Motivated by explicit congestion feedback, we believe that if end nodes could know the network state, the control decision based on them should be more effective. Indeed, some basic information of the network, such as the bandwidth, the aggregate throughput and the instantaneous queue length in the bottleneck link, is not hard to be measured. For example, we can measure the bottleneck bandwidth via packet pair technique[11], which has been also used in TCP Westwood[6] and some active probe tools. In additional, the aggregate throughput and the instantaneous queue length in the bottleneck link can be observed from the variation in RTT.

Our contribution in this paper is that we extend the population ecology theory to design a novel congestion control algorithm for high speed networks based on the network state information. Population ecology[2], which is the branch of ecology that studies the dynamics of populations, has something in common with congestion control. In essence, the major problem in both cases is resource allocation. For instance, in population ecology, different species in nature compete for food resources, and in congestion control, network flows have to compete for bandwidth resources. In order to study the rules of population evolution as time goes on, biologists have proposed many classical

1063-6927/08 $25.00 © 2008 IEEE
DOI 10.1109/ICDCS.2008.26

and practical models. If we treat the network flows as the species in nature, the throughput of the flows as the population number, and the bottleneck bandwidth as the food resources, these mature population ecology models would be helpful to design a novel congestion control algorithm.

Therefore, we use the key idea of constructing population ecology models to develop a totally different congestion control model, and implement the corresponding protocol, which is called Population Ecology TCP(PE-TCP). To gain insight into the behavior of PE-TCP, we prove the local asymptotic stability and analyze the efficiency and fairness convergence speed of PE-TCP, and then determine the preset parameters of PE-TCP. Finally, we evaluate the performance of PE-TCP using ns2 simulations.

The remainder of this paper is organized as follows. In section 2 we introduce the fundamentals of population ecology theory. Then we discuss the basic relationship between the population ecology models and the congestion control models. Section 3 presents the model and the implementation of PE-TCP. Section 4 evaluates its performance through theoretical analysis. Simulation results are given in Section 5 and related works are discussed in Section 6. Finally, Section 7 concludes the paper and describes the future work.

2 Fundamentals

In this section, we introduce the fundamentals of population ecology models, and explain how to relate congestion control with population ecology.

2.1 Population Ecology

Population ecology is the branch of ecology that studies the dynamics of populations. It is the most formalized area in biology. In this section, we introduce two basic population ecology models to explain the mathematical foundation of population ecology.

2.1.1 Logistic Model

The Logistic Model was developed by P. Verhulst who suggested that the ratio of population increase may be limited by the limited resources.

$$\frac{dx(t)}{dt} = rx(t)(1 - \frac{x(t)}{K}) \tag{1}$$

Parameter $x(t)$ is the population number. Parameter r is the intrinsic rate of increase, which can be interpreted as a difference between the birth rate and the death rate of 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 $\frac{dx(t)}{xdt}$ declines with the population number $x(t)$ and reaches 0 when $x(t) = K$. If population number exceeds K, then the population growth ratio becomes negative and the population number declines.

2.1.2 Lotka-Volterra Model

The Logistic Model present the population growth of a single species. In an attempt to understand large ecosystems, the Lotka-Volterra model involving the interaction of more than one species was proposed. This model is a development of the logistic equation of population growth. To apply the Logistic Model to two competing species, it is necessary to determine the competitive impact of one species on another, i.e., how many individuals of one species are equivalent to one individual of another species in terms of their use of the resource. This is the competitive coefficient, and is denoted as w. For simplicity, we assume that two species compete for K resources, and we have two sets of these variables: x_1, x_2, r_1, r_2. There are two competitive coefficients: the perceived competitive equivalence of species 2 on a member of species 1(w_1), and the perceived competitive equivalence of species 1 on a member of species 2(w_2). Therefore the Lotka-Volterra Model can be expressed as:

$$\begin{cases} \frac{dx_1}{dt} = r_1x_1(1 - \frac{x_1+w_1x_2}{K}) \\ \frac{dx_2}{dt} = r_2x_2(1 - \frac{x_2+w_2x_1}{K}) \end{cases} \tag{2}$$

Overall, population ecology models, especially the Lotka-Volterra Model, provide a mature mathematical method to analyze how the multiple species share the limited resources. It is easy to see that the Lotka-Volterra Model consists of:(1)the intrinsic rate of increase, and (2)the density-dependent factor. When there is no resource limitation, the population number will exponentially increase with the 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 a equilibrium state.

The strong points of population ecology models, i.e., exponentially density-dependent growth and reasonable equilibrium state, match the requirements of the high speed protocols exactly. Therefore it is instinctive to migrate the models of population ecology to design a novel congestion control algorithm. Just as we mentioned above, we can treat the network flows as the species in nature, the throughput of the flows as the population number, and the bottleneck bandwidth as the food resources.

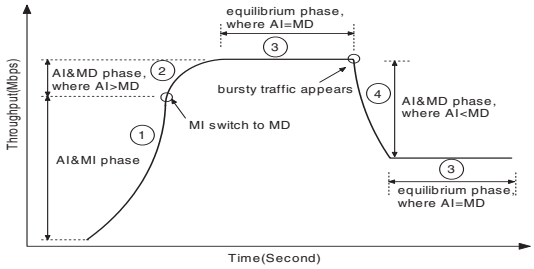


Figure 1. PE-TCP in working

3 Algorithm Design

3.1 Model

Consider that N long-lived flows share the bottleneck bandwidth B, the instantaneous throughput of the ith flow is denoted as $x_i(t)$, and the aggregate traffic in the bottleneck link is $X(t)$. Suppose the instantaneous queue length in the bottleneck is $q(t)$ and define a constant sampling interval T according to the queueing traffic. We propose the following congestion control model:

$$\frac{dx_i(t)}{dt} = bx_i(t) \cdot (1 - \frac{X(t) + q(t)/T}{B} + \frac{a}{bx_i(t)}) \quad (3)$$

where $X(t) = \sum_{i=1}^{N} x_i(t)$ and a, b and T are all positive constant.

Based on the foundation of population ecology, the control law of PE-TCP can be interpreted as the intrinsic rate of increase b, and the density-dependent factor is $(1 - \frac{X(t)+q(t)/T}{B} + \frac{a}{bx_i(t)})$. 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 equal to the bottleneck bandwidth minus the aggregate traffic and the queuing traffic in the router buffer. The term "$1 - \frac{X(t)+q(t)/T}{B}$" in the density-dependent factor just represents the normalized available bandwidth. To provide fast convergence to fairness, we hope weak flow with low throughput can increase more aggressively than strong flow with high throughput. Thus we add the term "$\frac{a}{b}\frac{1}{x_i(t)}$" into the density-dependent factor so that it is in inverse proportion to the current flow throughput.

Clearly, the equation (3) can be simplified as :

$$\frac{dx_i(t)}{dt} = a + bx_i(t) \cdot (1 - \frac{X(t) + q(t)/T}{B}) \quad (4)$$

From the equations (4), the growing process of PE-TCP throughput can be separated into four phases as shown in Figure 1.

1. When the network load is low, the quantity $(1 - \frac{X(t)+q(t)/T}{B})$ is close to 1, thus the throughput growth is essentially given by $(a + bx_i(t))$ which consists of a linear increase and an exponential increase. Comparing with AI mechanism in standard TCP, the PE-TCP throughput growth just like a mechanism combined by AI and MI in which a is the additive parameter and b is the multiplicative parameter. As the network load and flow throughput increase gradually, the quantity $(1 - \frac{X(t)+q(t)/T}{B})$ declines. Thus the dampening effect increases until the available bandwidth is consumed.

2. Although the available bandwidth is consumed, the AI mechanism causes the flow throughput to grow continuously. Thus the MI mechanism disappears and the MD mechanism appears at this time because the quantity $(1 - \frac{X(t)+q(t)/T}{B})$ become negative.

3. When the impact of the MD mechanism counteracts the impact of the AI mechanism, the growth rate becomes zero and the throughput of the flow reaches a steady state.

4. If the network load increases abruptly because of some burst traffic, the queue length will become large so that the impact of the MD mechanism overcomes the impact of the AI mechanism and the throughput growth rate becomes negative. Then the flow throughput will decline negative exponentially until it returns to another new steady state.

Clearly, due to the use of MI in the low load region, PE-TCP converges exponentially fast to high utilization. It just remedies the problem of standard TCP in high speed networks.

Another equation of the flow throughput and the queue length is shown by the fluid-flow queueing model [13]. For the bottleneck link, given the aggregate arriving rate and link bandwidth, we can calculate instantaneous queue length $q(t)$ from:

$$\frac{dq(t)}{dt} = X(t) - B \qquad (if \;\; q(t) > 0) \quad (5)$$

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

Consequently, the system model of the network can be described by the coupled, nonlinear differential equations (4) and (5).

3.2 Implementation

For each flow, the variables B, $X(t)$, and $q(t)$ are necessary information for executing the PE-TCP algorithm. These variables can be obtained by measurement. The

packet pair technique, which is also used in TCP Westwood, has been shown to be an effective method to measure the bottleneck bandwidth B. And from the equation (5), we see that if the varying rate of queue length $dq(t)/dt$ could be obtained, $X(t)$ would be calculated. The queueing delay, which is obtained by RTT measurement, can be used to calculate the instantaneous queue length and the varying rate of queue length. Next we will introduce how to measure B, $q(t)$ and $dq(t)/dt$ in detail.

We will first explain the variables that will be used in our model-based congestion control. For each flow, let

- $\tau(t)$: the current RTT when a ACK packet is received.
- τ_{min}: the minimal RTT.
- $\tau_q(t)$: the estimate of the current queueing delay, i.e. $\tau_q = \tau(t) - \tau_{min}$.
- $\tau_l(t)$: when the sender prepares to send a packet, $\tau(t)$ means the current RTT. When the corresponding ACK packet returns to the sender, the new $\tau(t)$ is recorded. Just before this time, we store the value of the old $\tau(t)$ into $\tau_l(t)$ which means the last RTT.
- τ_Δ: the minimal period between two consecutive ACK packets. Suppose the arrival time of the jth ACK packet and the $(j+1)$th ACK packet are t_j and t_{j+1} respectively, then we have $\tau_\Delta = \min[(t_2 - t_1), (t_3 - t_2), ..., (t_{j+1} - t_j), ...]$.

When the sender receives the ACK packets, it records the arrival time of every ACK packet. Then τ_Δ is easy to calculate. Based on packet pair technique[11], we have the following equation:

$$B = \frac{1}{\tau_\Delta} \tag{6}$$

where B is unit on packet per second.

In the PE-TCP algorithm, the sender records $\tau(t)$ and τ_{min} in real time. Using the queuing delay, the instantaneous queue length can be calculated from:

$$q(t) = \tau_q(t)B = \frac{\tau(t) - \tau_{min}}{\tau_\Delta} \tag{7}$$

Suppose the sample period is $\tau(t)$, the varying rate of queue length dq/dt can be approximated as:

$$\frac{dq(t)}{dt} \approx \frac{\Delta q}{\Delta t} = \frac{q(t) - q(t-\tau)}{\tau(t)} = \frac{\tau(t) - \tau_l(t)}{\tau(t) \cdot \tau_\Delta} \tag{8}$$

When the queue is empty, it means that there are enough bandwidth resources to allocate and there is no competition among the flows. Therefore for the ith flow, we assume it is the one and only flow in the network at this time, i.e., $X(t) = x_i(t)$. From the equation (5),we have

$$X(t) = \begin{cases} \frac{2\tau(t) - \tau_l(t)}{\tau(t) \cdot \tau_\Delta} & , (if \quad \tau(t) > \tau_{min}) \\ \\ x_i(t) & , (if \quad \tau(t) = \tau_{min}) \end{cases} \tag{9}$$

Suppose the sample period is $\tau(t)$, the varying rate of throughput $dx_i(t)/dt$ can be approximated as:

$$\frac{dx_i(t)}{dt} \approx \frac{x_i(t + \tau(t)) - x_i(t)}{\tau(t)} \tag{10}$$

After substituting equations (6), (7), (8) , (9) and (10) into (4), the final control law can be described with the following expression:

$$x_i(t + \tau(t)) = \begin{cases} x_i(t) + a\tau(t) + b\tau(t)x_i(t) \cdot (1 - \\ \frac{2\tau(t) - \tau_l(t)}{\tau(t)} - \frac{\tau(t) - \tau_{min}}{T}) \\ , (if \quad \tau(t) > \tau_{min}) \\ \\ x_i(t) + a\tau(t) + b\tau(t)x_i(t) \cdot (1 - \\ x_i(t)\tau_\Delta) \quad , (if \quad \tau(t) = \tau_{min}) \end{cases} \tag{11}$$

During transmission, the sender stores the current throughput $x_i(t)$ and $\tau(t)$ values into a hash table. After a round trip time, the corresponding ACK packet is received, the sender updates τ_{min} and τ_Δ and reads $x_i(t)$ and $\tau_l(t)$ from the hash table. Thus, the new admitted throughput $x_i(t+\tau)$ can be calculated from equation (11). The pseudo-code of the PE-TCP algorithm is listed in Table 1.

Table 1. Pseudo-code of PE-TCP Algorithm.

```
function Send(){
    Store x and τ into a hash table where
     the key value is the sequence number;
    Send pair packets;
    ............
}
function Receive(){
    Record the arrival time of ACK pakcet;
    Update τ_Δ;
    Compute τ;
    Update τ_min;
    Read x and τ_l from the hash table based
    on the sequence number in ack packet;
    Caculte the new x through equation (11);
    Reset the timer;
    Send();
    ............
}
```

4 Performance Analysis

In this section, through the evaluation of the stability, fairness and convergence of PE-TCP, we analyze the influence of parameters a, b, and T to PE-TCP's performance, and provide a guideline to determine the preset parameters of PE-TCP.

4.1 Stability and Fairness

When multi-flows compete for limited bandwidth resources, the possible results will be (1)multi-flows coexist or (2)some flows will persist, but other flows will be starved. To validate that PE-TCP can achieve a fair and stable coexisting state, we introduce the following theorem proved in our previous work [7]:

Theorem 1 *The N-dimensional system described by differential equations (4)(5) is locally asymptotically stable independent of the bottleneck bandwidth and the number of flows. Moreover, the equilibrium queue length is* $q^* = \frac{aTN}{b}$*, and all the flows have the same steady state throughput* $x_i^* = \frac{B}{N}$.

4.2 Convergence

In order to avoid that the congestion control algorithm is unable to fully utilize the bandwidth resource within a longer time, we expect the algorithm to have a fast convergence time. At the same time, when a new flow enters into the network in which the whole bandwidth resources have been completely shared by many existing flows, we hope the new flow can achieve the fairness quickly. Thereby we believe the convergence property of transport protocols should include two quantity metrics: convergence to efficiency and convergence to fairness[12]. Based on the definitions of θ *efficiency* and ε *fairness* in [7], we introduce the following two theorems proved in [7].

Theorem 2 *Consider N synchronous PE-TCP flows start to compete for the bottleneck bandwidth* B *with the initial throughput of each flow is* x_0*, where* N*,* B*,* x_0 *satisfy* $B >> Nx_0 >> 1$ *and* $N >> 1$*, then after*

$$t_\theta \approx \frac{\ln \frac{\theta}{1-\theta} + \ln \frac{B}{Nx_0}}{b} \tag{12}$$

, the network achieves θ *efficiency.*

The time of convergence to efficiency of PE-TCP is $O(\ln \frac{B}{N})$. For example, if we set $b = 1$, and $x_0 = 0.1$Mbps, while choosing the target θ as 99%, then one PE-TCP flow saturates a 100Mbps link in only 11.5 seconds.

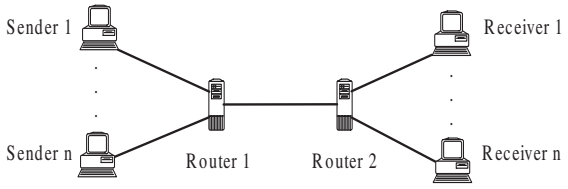


Figure 2. Dumbbell topology

Theorem 3 *Consider N synchronous PE-TCP flows have completely shared the bottleneck bandwidth* B *at steady state, and suppose a new flow enters into the network with initial throughput* x_0*, where* N*,* B*,* x_0 *satisfy* $B >> Nx_0 >> 1$ *and* $N >> 1$*, then after*

$$t_\varepsilon \approx \frac{B \ln \frac{1}{1-\varepsilon}}{aN} \tag{13}$$

, the network achieves ε *fairness.*

Clearly, the time of convergence to fairness is $O(\frac{B}{N})$. For example, if we set $a = 1$Mbps, and choose the target ε as 99%, then the network can achieve ε *fairness* state in 153 seconds with a 100Mbps bottleneck bandwidth shared by four PE-TCP flows.

4.3 Setting the parameters

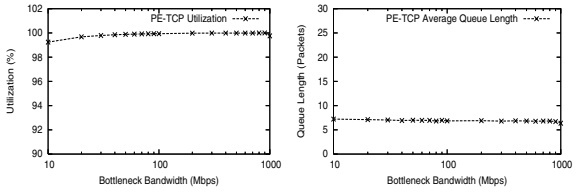
In this section, we provide a guideline to determine the preset parameters of PE-TCP: a, b, and T.

From theorem 2, we observe that increasing b shortens the time of convergence to efficiency. From theorem 3, we observe that increasing a shortens the time of convergence to fairness. From theorem 1, we observe that the equilibrium queue length is $q^* = \frac{aT}{b}N$. In order to avoid the number of flows making a negative impact on the queue length, we must choose small $\frac{aT}{b}$ whenever possible to keep a relative low queue length compared with the bandwidth-delay product.

Therefore, we set $a = 1$Mbps, $b = 1$, $T = 0.01$s and $x_0 = 0.1$Mbps in the final implementation of PE-TCP to balance the convergence to efficiency, convergence to fairness and queue length at steady state.

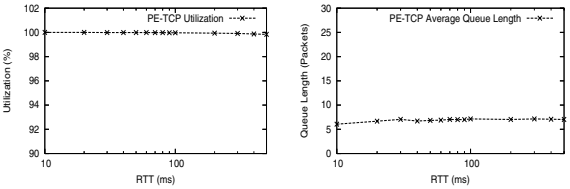
5 Simulation Results

In this section, we present some simulation results evaluating the performance of PE-TCP. For comparison purposes, we choose regular TCP-SACK and some other high-speed TCP variants, i.e., HSTCP, and BICTCP to test together. We used ns2 for the simulation experiments. The scenario used in all simulations is the dumbbell topology depicted in Figure 2. It consists of sender/receiver hosts,



(a) Bottleneck Average Utilization (b) Bottleneck Queue

Figure 3. PE-TCP with the bottleneck bandwidth ranging from 10Mbps to 1Gbps.



(a) Bottleneck Average Utilization (b) Bottleneck Queue

Figure 4. PE-TCP with the RTT ranging from 10ms to 500ms.

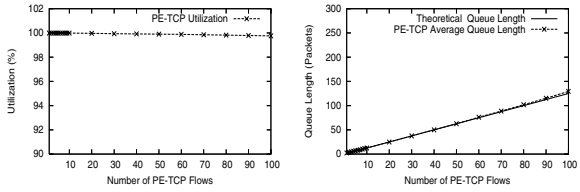
two routers, and links between the hosts and routers. We deployed a tail-drop discipline at the router buffer, and the buffer size is set to the 20% bandwidth-delay product of the bottleneck link between the two routers. The small router buffer size is in line with recent research results showing that the buffer size of a bottleneck link with a high degree of multiplexing of TCP connections can be much less than the bandwidth delay product[1]. In all experiments the data packet size is 1000 bytes, while the ACK packet is 40 bytes. For all the graphs, throughput, utilization, packet loss rate, and queue length are sampled over 1s intervals.

5.1 Experiment 1: Scalability

This experiment covers a wide range of network scenarios, such as link bandwidth in the range of [10Mbps, 1Gbps], round trip time in the range of [10ms, 500ms], a number of long-lived flows in the range of [1, 100]. In following simulations, all the simulations run for 400ms to ensure that the system has reached the steady state. And the average utilization statistics omit the first 200ms simulation time which is considered as a "warm up" phase.

5.1.1 Impact of Bottleneck Bandwidth

In this simulation, we vary the bottleneck bandwidth from 10Mbps to 1Gbps and fix the round trip time as 50ms. There are five PE-TCP flows on the forward path. Figure 3(a) demonstrates that as the bandwidth increases, it is surprising that the bottleneck average utilization is always close to 100%. From Figure 3(b), we see that the average queue length remains at 7 packets, which is close to the theoretical value of 6.25 packets. This simulation also shows that no packet loss occurs during the entire simulation time.



(a) Bottleneck Average Utilization (b) Bottleneck Queue

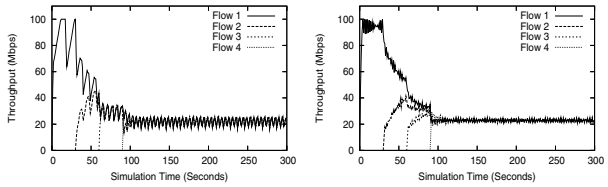
Figure 5. PE-TCP with the number of flows ranging from 1 to 100.

5.1.2 Impact of RTT

In this simulation, we vary the RTT from 10ms to 500ms, and fix the bottleneck bandwidth at 500Mbps. There are 5 PE-TCP flows are on the forward path. As shown in Figure 4(a), the bottleneck average utilization degrades slightly with the increasing of RTT. However we must note that even in the worst case i.e., 500ms RTT, the utilization is still higher than 99%. In real networks, we believe that flows with RTT larger than 400ms are very rare. From Figure 4(a), we can see that the bottleneck average utilization is close to 100% when RTT is 400ms. Figure 4(b) shows that the average queue length remains at 7 packets, which is close to the theoretical value 6.25 packets. This simulation also shows that no packet loss occurs during the entire simulation time.

5.1.3 Impact of Number of Flows

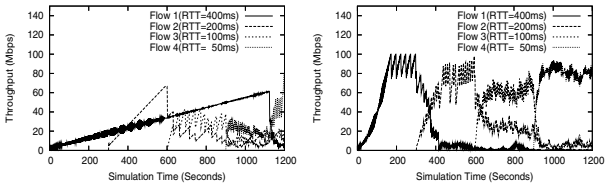
In this simulation, we fix the bottleneck bandwidth at 500Mbps and set the RTT of long-lived flows to 50ms. With an increase in the number of flows from 1 to 100, the bottleneck average utilization is mostly close to 100%, and the loss rate remains zero. Figure 5(a) shows the results. We see from Figure 5(b) that the bottleneck average queue length increases linearly with the number of flows and it is close to the theoretical value. When there are 100 PE-TCP flows in the network, the average queue length is 135 packets, which is 4.3% of the bandwidth-delay product(3125 packets). Compared with the other loss-based mechanism, PE-TCP maintains a lower queue length . However, we must note that although we have set a very small T value to shorten the equilibrium queue length since it is $O(N)$. When the number of flows is very large and the queue buffer



(a) TCP-SACK (b) HSTCP

(c) BICTCP (d) PE-TCP

Figure 6. The throughput dynamics of four flows of the same RTT with different protocols.



(a) TCP-SACK (b) HSTCP

(c) BICTCP (d) PE-TCP

Figure 7. The throughput dynamics of four flows of the different RTT with different protocols.

is very small, packets will be dropped. This drawback will be study further and improved in our further work.

Through a wide range of network configurations, the results validate the good scalability of PE-TCP since PE-TCP makes the bottleneck link always maintain high utilization with low queue length and zero packet drop independent of the bottleneck bandwidth, the RTT and the number of flows.

5.2 Experiment 2: RTT Fairness

In this experiment, we run four flows of a high speed protocol with the different RTT of 400ms, 200ms, 100ms and 50ms. We start the 400ms RTT flow, 200ms RTT flow, 100ms RTT flow and 50ms RTT flow start at 0s, 300s, 600s and 900s respectively. The bottleneck bandwidth is also set to 100Mbps. Figure 7 shows that PE-TCP has a better RTT fairness and faster convergence than the other protocols. This strong point remains even with large RTT differences. Clearly, HSTCP and BICTCP have a serious RTT unfairness problem since the short RTT flows starve off the long RTT flows.

5.3 Experiment 3: Robustness

This experiment validates the robustness of PE-TCP in the presence of web traffic and burst traffic. We start the simulation with four PE-TCP flows where the bottleneck bandwidth is 100Mbps and RTT is 50ms. During the 200s simulation time, the average web traffic of 20Mbps generated by 100 random on-off sources is always on. At 100s, burst CBR traffic of 40Mbps generated by 10 UDP sources is injected into the network. At 200s, all UDP sources drop

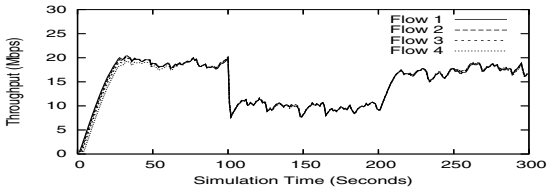


Figure 8. PE-TCP is robust against web traffic and burst traffic.

out. As shown in Figure 8, the PE-TCP flows converge to a equilibrium rate of 20Mbps at 10s. When the burst traffic appears at 100s, the PE-TCP flows give up the bandwidth rapidly. At 110s, the PE-TCP flows converge to a new equilibrium rate(10Mbps). After the burst traffic leaves at 200s, the PE-TCP flows catch the available bandwidth and converge to the previous equilibrium rate quickly.

6 Related Works

Our work builds upon some related work which have attempted to design transport protocols based on the population ecology models. For example, M. Welzl[16] used the Logistic Model to design CADPC. One limitation of CADPC is that it must be supported by router and end node, to determine the available bandwidth so that it is efficient only when all routers along the route support CADPC. Compared with CADPC, we do not use the population ecology models directly, but extract the key idea of constructing population ecology models and then create a new model to implement the congestion control algorithm. In addition,

we also consider the effect of queuing on the whole congestion control mechanism and provide a complete theoretical analysis based on the system model, which are not shown in the other two works. We think that the advantage of PE-TCP is that it is simpler and more practical in real networks.

7 Conclusions and Future Work

In this paper we developed a novel congestion control algorithm for high speed networks based on population ecology models. Through theoretical analysis and simulation validations, we show that PE-TCP can provide high utilization, fast convergence, fair bandwidth allocation, and near-zero packet drops, all of which are desirable for high speed networks. We believe that population ecology is useful for solving the resource allocation problem, and is easy to analyze and apply to the behavior of congestion control.

There are three questions remain unaddressed. First, the current measurement techniques according to minimal RTT and bottleneck bandwidth are not robust against possible routing changes during the lifetime of a connection. Whether we should remeasure this information periodically or mitigate the adverse effect by filtering out samples require further study. Second, we need to choose a very small T value to shorten the equilibrium queue length since it is $O(N)$. When the number of flows is very large and the queue buffer is small, packets will be dropped. Thus, we will try to reconstruct a more advanced model to make the equilibrium queue length independent of the number of flows. Finally, how to keep multiple transport protocols coexist reasonably in the same network is still an open issue.

8 Acknowledgments

This Work is supported by ChinaGrid project of Ministry of Education of China, Natural Science Foundation of China (90412006, 90412011, 60573110, 90612016, 60673152, 60573122, 60773138), National Grand Fundamental Research Program of China (2004CB318000, 2003CB317007, 2006CB303000, 2003CB314804), National High Technology Development Program of China (2006AA01A108, 2006AA01A111, 2006AA01A101, 2006AA01Z225), and IST programme of the European Commission (DG Information Society and Media, project n^o 034442).

References

[1] D. Barman, G. Smaragdakis, and I. Matta. The Effect of Router Buffer Size on HighSpeed TCP Performance. In *IEEE Globecom 2004*, Dallas, TX, December 2004.

[2] M. Begon, J. Harper, and C. Townsend. *Ecology-Individuals Populations and Communities*. Blackwell Science, Oxford, third edition, 1996.

[3] S. Bhandarkar, S. Jain, and A. L. N. Reddy. Improving TCP Performance in High Bandwidth High RTT Links Using Layered Congestion Control. In *Proc. 3rd Workshop on Protocols for Fast Long Distance Networks*, 2005.

[4] D. Chiu and R. Jain. Analysis of the Increase and Decrease Algorithms for Congestion Avoidance in Computer Networks. *Computer Networks and ISDN Systems*, 17:1–14, 1989.

[5] S. Floyd. High-speed TCP for Large Congestion Windows. *RFC 3649*, 2003.

[6] M. Gerla, B. K. F. Ng, M. Y. Sanadidi, M. Valla, and R. Wang. TCP Westwood with adaptive bandwidth estimation to improve efficiency/friendliness tradeoffs. *Journal of computer communications*, 27(1), January 2004.

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

[8] C. Jin, D. Wei, and S. Low. FAST TCP: Motivation, Architecture, Algorithms, Performance. In *IEEE INFOCOM*, March 2004.

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

[10] T. Kelly. Scalable TCP: Improving Performance in High-speed Wide Area Networks. In *First International Workshop on Protocols for Fast Long-Distance Networks*, February 2003.

[11] K. Lai and M. Baker. Measuring link bandwidths using a deterministic model of packet delay. In *ACM SIGCOMM*, pages 283–294, 2000.

[12] 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.

[13] 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*, pages 151–160, 2000.

[14] I. Rhee and L. Xu. CUBIC: A New TCP-Friendly High-Speed TCP Variant. In *Proc. 3rd Workshop on Protocols for Fast Long Distance Networks*, 2005.

[15] S. Shalunov, L. D. Dunn, Y. Gu, S. Low, I. Rhee, S. Senger, B. Wydrowski, and L. Xu. Design Space for a Bulk Transport Tool. In *Internet2 Bulk Transport Working Group Report*, 2005.

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

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

[18] L. Xu, K. Harfoush, and I. Rhee. Binary Increase Congestion Control for Fast Long Distance Networks. In *IEEE INFOCOM*, 2004.

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