Bandwidth-based congestion control for TCP: measurement noise-aware parameter settings and self-induced oscillation

Mizuho Kodama
Graduate School of Information
Science and Technology, Osaka University
1-5 Yamadaoka, Suita,
Osaka, 565-0871, Japan
Email: m-kodama@ist.osaka-u.ac.jp

Go Hasegawa
Cybermedia Center,
Osaka University
1-32 Machikaneyama, Toyonaka,
Osaka, 560-0043, Japan
Email: hasegawa@cmc.osaka-u.ac.jp

Masayuki Murata
Graduate School of Information
Science and Technology, Osaka University
1-5 Yamadaoka, Suita,
Osaka, 565-0871, Japan
Email: murata@ist.osaka-u.ac.jp

***Abstract*—With increase in the heterogeneity and complexity of the Internet, many problems have emerged in the traditional TCP Reno. Against the problem, we have proposed a new *bandwidth-based* TCP congestion control mechanism, TCP Symbiosis. Unlike the recent works on delay-based and hybrid congestion control, TCP Symbiosis relies on the bandwidth measurement to control the congestion window size. Although we have confirmed that TCP Symbiosis has the effectiveness in terms of average throughput, stability, and scalability to the bandwidth-delay product, the throughput of TCP Symbiosis highly depends on the measurement results of available bandwidth. In this paper, we redesign TCP Symbiosis to deal with the measurement error and noise of the available bandwidth. First, we propose the dynamic parameter setting algorithm based on the variance of the measured available bandwidth. Second, for the purpose of absorbing the ill-effect of environmental change, we propose to add self-induced oscillation to the congestion window size of TCP connection. We confirm the effectiveness of the proposed methods through ns-2 simulation experiments.**

***Index Terms*—TCP, congestion control, network measurement, available bandwidth, measurement error, Lotka-Volterra competition model**

I. INTRODUCTION

Recent research results have revealed that there are many problems in TCP Reno's congestion control mechanism [1-5] with increase in the heterogeneity and complexity of the Internet. The main reasons for those problems are that the network congestion is indicated only by packet losses and that TCP Reno uses fixed Additive Increase and Multiplicative Decrease (AIMD) parameter values to increase and decrease the congestion window size, whereas those parameters should be changed according to the network environments.

Many solutions have been proposed for the above problems [5-11]. In general, they are classified into 3 types in terms of congestion indication: *loss-based* [5-7], *delay-based* [8, 9] and *hybrid* [10, 11] approaches. *Loss-based* approaches utilize packet losses for network congestion indication as TCP Reno, but employ the modified functions to regulate the congestion window size. *Delay-based* approaches utilize RTT values for the congestion indication to deal with the incipient network congestion. *Hybrid* approaches combine *loss-based* and *delay-based* approaches to alleviate their unfairness problems.

However, these studies have not focused on the bandwidth information. Since a window size indicates the maximum amount of packets that TCP can transmit for one Round Trip Time (RTT), an adequate window size for a TCP connection is equal to the product of the available bandwidth and the round-trip propagation delay of the network path between the sender and receiver hosts. TCP measures RTT by checking the departure times of data packets and the arrival times of the corresponding ACK packets. However, TCP Reno and its variants do not have an effective mechanism to recognize bandwidth-related information of the network path. We believe, therefore, that if a TCP sender recognizes the bandwidth information of the network path quickly and accurately, we can create a better mechanism for congestion control for TCP that can resolve various problems such as self-induced periodical packet losses.

We have proposed a new congestion control mechanism, TCP Symbiosis in [12]. TCP Symbiosis employs *bandwidth-based* congestion control mechanisms. That is, it controls the congestion window size based on the bandwidth information obtained from the measurement mechanism. Note that TCP Symbiosis can be utilized with any bandwidth measurement mechanisms. One possible solution is to combine TCP Symbiosis with ImTCP [13], which estimates the physical capacity and available bandwidth of the network path in an inline fashion, by using data and ACK packets transmitted by an active TCP connection. Since the ImTCP sender obtains bandwidth information every 3-4 RTTs, it is well able to follow the traffic fluctuation of the underlying IP network.

In TCP Symbiosis, we employ congestion control mechanisms based on the algorithms borrowed from biophysics. We utilize from the logistic growth model and the Lotka-Volterra competition model [14], both of which are used in biophysics to describe changes in the population of species. The biophysics models were chosen based on their intrinsic stability and robustness, which is achieved even when they behave without any interaction in an autonomous and distributed fashion. This is the case for the congestion control of TCP: each TCP connection behaves independently, but still we want to improve the bandwidth utilization and the throughput of the connection.

In the previous study, we have investigated the performance of TCP Symbiosis by ns-2 [15] simulation and implementation experiments under the public Internet environment [12, 16]. As a result, we have confirmed that TCP Symbiosis has the effectiveness in terms of average throughput, stability, and scalability to the bandwidth-delay product. However, we have also revealed that the throughput of TCP Symbiosis highly depends on the measurement results of available bandwidth by ImTCP. When the measurement result is larger than true value, overlarge window size causes the buffer overflow. On the other hand, when the measurement result is smaller than true value, the window size of TCP Symbiosis is not large enough to utilize the available bandwidth to the full. That is, the performance of TCP Symbiosis would degrade when the measurement results of the bandwidth information have some errors and noises.

In this paper, therefore, we redesign TCP Symbiosis to deal with the measurement error and noise of the available bandwidth. First, we give mathematical explanation of the effect of error and noise in the available bandwidth measurement, and reveal that they affect the parameter settings in Lotka-Volterra competition model for fair share of the bottleneck bandwidth among competing TCP connections. We then propose the dynamic parameter setting algorithm based on the variance of the measured available bandwidth. Second, we show that TCP Symbiosis degrades its throughput against the sudden changes in the network environment (e.g. the amount of the available bandwidth). Against this problem, we propose to add self-induced oscillation to the congestion window size of a TCP connection to absorb the effect of the environmental change. We confirm the effectiveness of the proposed methods through ns-2 simulation experiments. We believe that the proposed methods against measurement error and noise can be applied not only to bandwidth-based approaches like TCP Symbiosis but also to delay-based and hybrid approaches.

The rest of this paper is organized as follows. In Section II, we briefly introduce TCP Symbiosis. Then, we introduce two approaches and demonstrate them by the simulation results in Sections III and IV. We finally conclude this paper and offer future work in Section V.

II. TCP SYMBIOSIS

TCP Symbiosis utilizes the information of physical capacity and available bandwidth obtained from an inline measurement technique [13], and the window updating algorithm is borrowed from biophysics models; the logistic growth and Lotka-Volterra competition models. In many biological systems, the actions of the entity (e.g., living organism) are not determined based on the results of direct interactions among entities, but rather on information obtained through the environment, which is a fundamental necessary condition for the system to be robust. Furthermore, the species which have the same characteristic (e.g., the carrying capacity of the environment, the intrinsic growth and the ratio of the competition coefficient of other species) can keep fairness. Therefore, we expect TCP Symbiosis can achieve stability, robustness and fairness.

The logistic equation is a formula that represents the evolution of the population of a single species over time. Generally, the per capita birth rate of a species increases as the population of the species becomes larger. However, since there are various restrictions on living environments, the environment has a carrying capacity (upper limit of the total population size), which is usually determined by the available sustaining resources. The Lotka-Volterra competition model is a well known model for examining the population growth of two or more species that are engaged in inter-specific competition. It makes a change to the logistic growth model, as it includes the effects of both inter-specific competition and intra-specific competition.

The basic two species Lotka-Volterra competition model with both species N_1 and N_2 having logistic growth in the absence of the other is comprised of the following equations [14]:

$$\frac{d}{dt}N_1 = \epsilon_1\left(1 - \frac{N_1 + \gamma_{12}N_2}{K_1}\right)N_1, \tag{1}$$

$$\frac{d}{dt}N_2 = \epsilon_2\left(1 - \frac{N_2 + \gamma_{21}N_1}{K_2}\right)N_2 \tag{2}$$

where N_i, K_i, and ϵ_i are the population of the species, the carrying capacity of the environment, and the intrinsic growth rate of the species i, respectively. γ_{ij} is the ratio of the competition coefficient of species i with respect of species j, and t is time. Furthermore, we can easily extend Equations (1) and (2) for n species as follows:

$$\frac{d}{dt}N_i = \epsilon\left(1 - \frac{N_i + \gamma\sum_{j=1,i\neq j}^{n} N_j}{K}\right)N_i. \tag{3}$$

To convert the changes in the population of species into the changes in congestion window sizes, the population of a species can be viewed as the window size of a TCP connection, the carrying capacity of the environment as the physical capacity of the network path, and inter-specific competition among species as bandwidth sharing among competing TCP connections. We estimate the sum of the window size of all of the other connections by using the physical capacity and available bandwidth of a network path. It gives the following equation:

$$\frac{d}{dt}w_i = \epsilon\left(1 - \frac{w_i + \gamma(K - A_i)\tau_i}{K\tau_i}\right)w_i \tag{4}$$

where w_i, A_i and τ_i are the congestion window size, the available bandwidth and the minimum value of the RTTs of TCP connection i, respectively. K is physical capacity. Finally, we integrate Equation (4) as follows:

$$w_i(t) = \frac{w_i(0)e^{\epsilon t\left\{1-\gamma\left(1-\frac{A_i}{K}\right)\right\}}\{K - \gamma(K - A_i)\}\tau_i}{w_i(0)\left(e^{\epsilon t\left\{1-\gamma\left(1-\frac{A_i}{K}\right)\right\}} - 1\right) + \{K - \gamma(K - A_i)\}\tau_i} \tag{5}$$

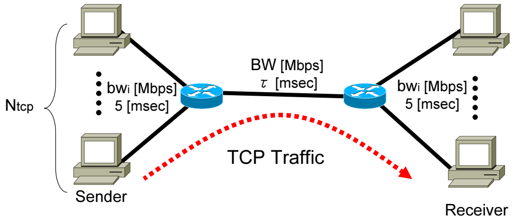


Fig. 1. Network topology in simulation experiments

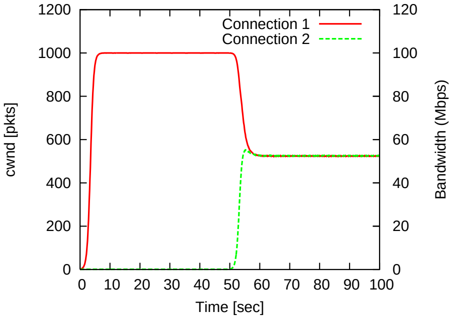


Fig. 2. Basic behavior of TCP Symbiosis

In Equation (5), when we set the initial value of the window size ($w_i(0)$) and the current time to 0 ($t = 0$), we can directly obtain window size $w_i(t)$ for any time t. We use the above equation for the control algorithm of the window size of TCP connections.

Since Equation (5) requires the measurements of the physical capacity and available bandwidth of an end-to-end network path, we use the same algorithm as TCP Reno for window updating algorithm until the measurement results are obtained through ImTCP. In case of detecting packet losses by receiving three duplicated ACKs, a window size is halved in an identical way to TCP Reno. Similarly, when a timeout occurs, the sender TCP discards all measurement results, the window size is reset to one packet, and the slow-start phase begins. With obtained bandwidth information of the network path, we can calculate the window size from Equation (5) by regarding the arrival interval between the latest two ACK packets as t.

We show the typical behavior of TCP Symbiosis by using the results of the simple simulation experiment. The simulation topology is a dumbbell topology as shown in Figure 1. The bottleneck link bandwidth is 100 Mbps and the minimum RTT is 120 msec. We run 2 TCP Symbiosis connections and the second connection joins the network at 50 sec. Figure 2 illustrates the change in the congestion window size of the two TCP Symbiosis connections. This figure shows that TCP Symbiosis quickly converges the window size without packet losses and the two connections fairly share the bandwidth. Refer to [12, 16] for more performance evaluation results.

III. MEASUREMENT ERROR-AWARE PARAMETER SETTINGS

A. *Mathematical explanation and simulation results*

In [16], we have revealed that the measurement error and noise of the available bandwidth degrade the throughput of TCP Symbiosis. In this section, we give the mathematical explanation of the effect of the measurement noise and propose the dynamic parameter setting algorithm to alleviate the problem.

In general, the measured value of the available bandwidth, which is denoted as $A'(t)$ where t is time, includes the measurement error. On the other hand, we assume that the physical capacity, K, remains constant and can be accurately measured, since the measurement error of the physical capacity is relatively small compared with that of the available bandwidth. Equation (4) can be written as follows:

$$\frac{d}{dt}w(t) = \epsilon\left(1 - \frac{w(t) + \gamma(K - A'(t))\tau}{K\tau}\right)w(t). \tag{6}$$

Given that the noise ratio of $(K - A'(t))$ is $n(t)$. Then,

$$K - A'(t) = (1 + n(t))(K - A(t)), \tag{7}$$

where $A(t)$ is the true value of the available bandwidth at time t. Hence Equation (6) can be written as follows:

$$\frac{d}{dt}w(t) = \epsilon\left(1 - \frac{w(t) + (1 + n(t))\gamma(K - A(t))\tau}{K\tau}\right)w(t).$$

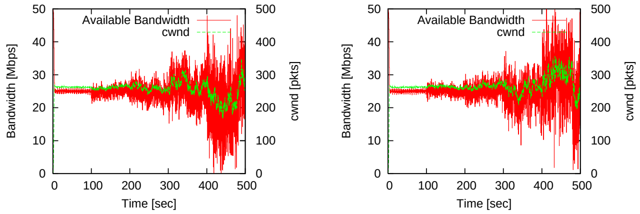
Now, Equation (7) can be written as follows:

$$\gamma'(t) = (1 + n(t))\gamma, \tag{8}$$

$$\frac{d}{dt}w(t) = \epsilon\left(1 - \frac{w(t) + \gamma'(t)(K - A(t))\tau}{K\tau}\right)w(t). \tag{9}$$

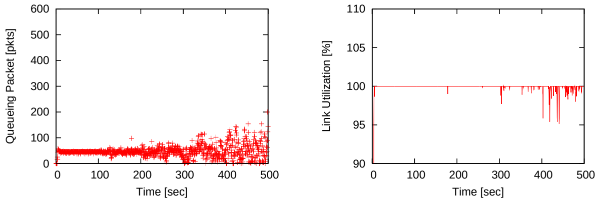
By comparing Equations (4) and (9), we can observe that the measurement noise in the available bandwidth can be viewed as the noise in the control parameter γ.

As shown in Section II, γ has to satisfy $0 < \gamma < 1$ for stable behavior in Lotka-Volterra competition model. When $\gamma > 1$, some species (corresponding to the competing TCP connections) become extinct, meaning that some TCP connections suffer from very low throughput. Therefore, the noise in γ degrades the performance of TCP Symbiosis because it sometimes makes γ be larger than 1. Setting γ to the safe side (far smaller than 1) enables to avoid the extinct situation. However, we can recognize from the Equation (4) that the small value of γ would increase the queue length at the bottleneck link in the convergence states. We exhibit the above property by simulation results. The simulation topology is the same as in Figure 2. The bottleneck link bandwidth is 50 Mbps and the minimum RTT is 120 msec. We ran 2 TCP Symbiosis flows at time 0. We intentionally add a white noise to the measurement results of the available bandwidth to investigate the effect of the measurement noise. We start the simulation with the small noise, and double the noise size every 100 sec. The results are summarized in Figure 3 ($\gamma = 0.95$) and Figure 4 ($\gamma = 0.5$). When γ closed to 1 ($\gamma = 0.95$), we can see that the congestion window size of the two connections becomes more



(a) Available bandwidth and cwnd of connection 1

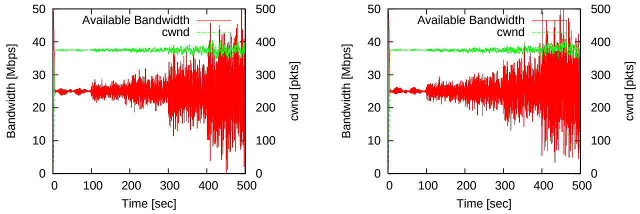
(b) Available bandwidth and cwnd of connection 2



(c) Queue length of bottleneck link

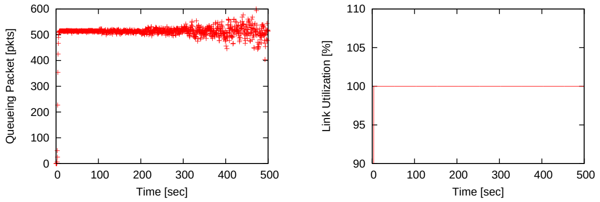
(d) Bottleneck link utilization

Fig. 3. Simulation results with $\gamma = 0.95$



(a) Available bandwidth and cwnd of connection 1

(b) Available bandwidth and cwnd of connection 2



(c) Queue length of bottleneck link

(d) Bottleneck link utilization

Fig. 4. Simulation results with $\gamma = 0.5$

unstable with larger noise. This is because the larger noise in the available bandwidth makes $\gamma'(t)$ in Equation (9) to be larger than 1 frequently. This results in unfairness between the two connections and unstable queue length (Figure 3(c)) and lower utilization of the bottleneck link (Figure 3(d)). On the other hand, when γ is far from 1 ($\gamma = 0.5$), the congestion window size of the two connections remain stable regardless of the noise ratio (Figures 4(a) and 4(b)), and the bottleneck link utilization remains 100% (Figure 4(d)). However, as shown in Figure 4(c), the queue length of the bottleneck link remains larger value as compared with Figure 3(c) with γ=0.95. This large queue length would increase the queueing delay at the bottleneck link buffer and the probability of buffer overflow also increases with sudden change of the background traffic.

From the above results, we conclude that the constant value of γ can not perform well in noise-prone environments.

B. *Dynamic parameter setting*

Against the above problem, we propose the dynamic setting algorithm of γ according to the measurement results of the available bandwidth. We determine γ based on the variance of the measured bandwidth values over time.

From Equation (8), we can dynamically determine γ as follows:

$$\gamma(t) = \frac{\gamma_t}{1 + lC_v}, \qquad (10)$$

where γ_t is the target value for γ, which is the ideal value. Since $n(t)$ in Equation (8) is the noise ratio of $(K - A'(t))$, we replace $n(t)$ with C_v, which is the coefficient of variation of the measurement values of $(K - A'(t))$. In addition, we introduce the parameter l to control the degree of noise awareness.

We obtain the deviation of the measurement results of the available bandwidth as RTO calculation in TCP [17]:

$$\begin{aligned} Err &= x - \mu, \\ \mu &\leftarrow \mu + gErr, \\ \sigma &\leftarrow \sigma + h(|Err| - \sigma), \end{aligned}$$

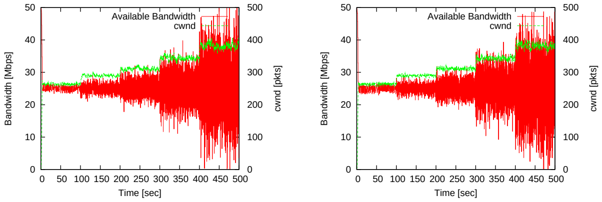
where x, μ and σ are the latest measured value, the moving average value and the average deviation. g and h are the weighting factor. Then we can calculate $C_v = \sigma/\mu$

In Figure 5, we show the simulation results with the same simulation setting as in Figures 3 and 4, where we set g=0.125, h=0.25 [17], l=4 and γ_t=0.95. Figure 5(e) exhibits the change in $\gamma(t)$ in the proposed mechanism. We can see from Figure 5 that the congestion window sizes of the two connections remain stable regardless of the degree of the noise ratio (Figures 5(a) and 5(b)) and that the utilization of the bottleneck link keeps 100% (Figure 5(d)). In addition, the queue length of the bottleneck link is regulated properly and is smaller than in Figure 4(c) especially when the degree of the noise ratio is small.

IV. SELF-INDUCED OSCILLATION FOR ABSORBING ENVIRONMENTAL CHANGE

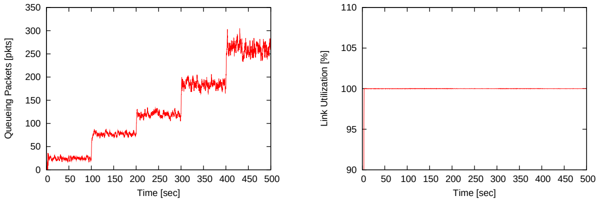
A. *Effect of environmental change*

TCP Symbiosis highly depends on the measurement results of available bandwidth of a network path. In Section III, we discussed the dynamic parameter setting according to the degree of the measurement noise, while the true value of the available bandwidth remains unchanged. Here, we focus on the behavior of TCP Symbiosis when the true value of the available bandwidth suddenly changes due to some environmental change.



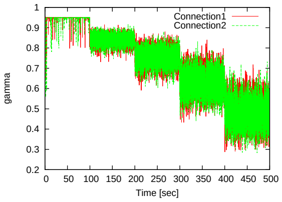
(a) Available bandwidth and cwnd of connection 1

(b) Available bandwidth and cwnd of connection 2



(c) Queue length of bottleneck link

(d) Bottleneck link utilization



(e) Change in the value of γ

Fig. 5. Simulation results with proposed method

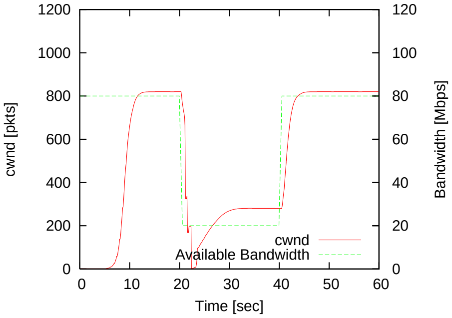


Fig. 6. Change in the congestion window size without self-induced oscillation

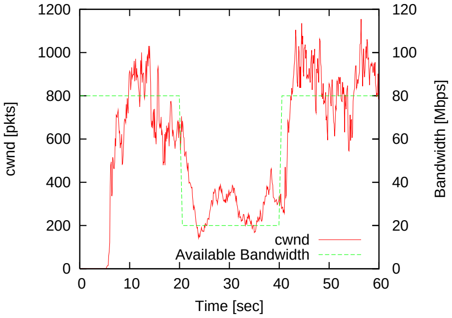


Fig. 7. Change in the congestion window size with self-induced oscillation

Generally, the end-to-end available bandwidth is measured by repeating the exchange of probe packets between endhosts. For example, in ImTCP, the sender TCP transmits TCP data packets with pre-defined transmission intervals, and calculates the available bandwidth from arrival intervals of the corresponding ACK packets. ImTCP gives estimation results every 3-4 RTTs. This means that when the true value of the available bandwidth changes, it takes several RTTs for the sender-side TCP to recognize the change. Furthermore, TCP symbiosis requires additional RTTs to converge its congestion window size according to the new value of the available bandwidth. These delayed behaviors of TCP Symbiosis affect the performance of the congestion control against the environmental change.

We show an example of the performance degradation by simulation experiments. Figure 6 shows the change in the congestion window size when the available bandwidth changes as 80 Mbps in 0-20 sec, 20 Mbps in 20-40 sec, and 80 Mbps in 40-60 sec. Note that the buffer size of the bottleneck link is set to 1000 packets, the background traffic is composed by an UDP flow, and there is one TCP Symbiosis connection. The bottleneck link bandwidth is 100 Mbps and the minimum RTT is 120 msec. When the available bandwidth suddenly decreases at 20sec, TCP's retransmission timeout occurs and the congestion window size is reset to one packet, which significantly degrades the throughput. This is caused by the packet losses at the buffer of the bottleneck link. On the other hand, when the available bandwidth suddenly increases at 40sec, TCP Symbiosis requires some delay to fill-up the increased bandwidth.

This delayed behavior of TCP Symbiosis is caused since it relies on the measurement results of the available bandwidth. When the network environment is stable, this mechanism gives good results in terms of stable and high throughput. However, when facing the environmental change, it degrades the performance due to its delayed behavior.

B. *Self-induced oscillation*

To solve the above problem, we propose to add self-induced oscillation to the congestion window size. This means that the congestion window size remains fluctuating even when the available bandwidth remains unchanged. Note that the average throughput does not change since the oscillation is equally injected in both of increase and decrease directions. By this mechanism, we expect that the effect of the environmental change can be absorbed probabilistically.

In [18], the authors define the two-species Lotka-Volterra model in the presence of a noise as follows:

$$\begin{aligned} \frac{dx}{dt} &= \mu_1 x(\alpha_1 - x - \beta_1(t)y) + x\xi_x(t), \\ \frac{dy}{dt} &= \mu_2 y(\alpha_2 - y - \beta_2(t)x) + y\xi_y(t), \end{aligned} \tag{11}$$

where x and y are the population of the species, α_1 and α_2 are the intrinsic growth rate, $\beta_1(t)$ and $\beta_2(t)$ are the coefficient of

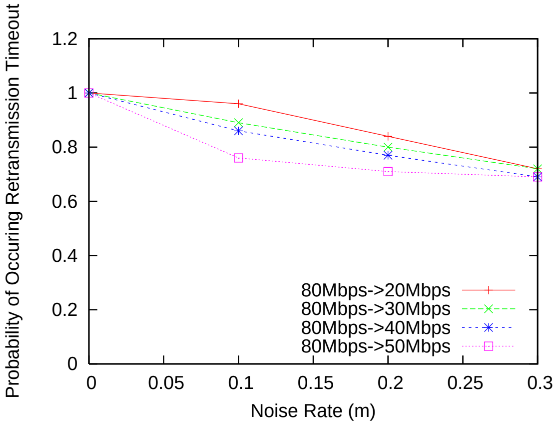


Fig. 8. Probability of retransmission timeout

intra-specific competition. $\xi_x(t)$ and $\xi_y(t)$ denotes Gaussian white noise. These equations represent the change in the population of two species with the presence of the noise and some other deterministic periodical drive force present in the ecosystems such as the temperature. We refer to these equations for adding self-induced oscillation to TCP Symbiosis mechanisms, by converting the noise in Equation (11) into the self-induced oscillation. We then obtain the following equation to control the congestion window size as follows:

$$\frac{d}{dt}w(t) = \epsilon\left(1 - \frac{w(t) + \gamma(K - A(t))\tau}{K\tau}\right)w(t) + mw(t)\xi(t), \tag{12}$$

where $\xi(t)$ denotes Gaussian white noise and m is a parameter to determine the degree of the self-induced oscillation.

Figure 7 shows the simulation results of the proposed method with $m = 0.3$ in the same simulation settings as in Figure 6. We can observe that the congestion window size continues fluctuating even when the available bandwidth remains stable. Note that at 20 sec, the proposed method experiences no packet loss and retransmission timeout. After the sudden increase of the available bandwidth at 40 sec, the congestion window size increases in shorter time than in Figure 6.

In Figure 8, we show the probability of occurring retransmission timeout when the available bandwidth suddenly decreases, as a function of m. Note that m=0 corresponds to TCP Symbiosis without self-induced oscillation. In the graph, we show the results of the three cases of the decrease degree of the available bandwidth. We can observe from this figure that by adding the oscillation to the congestion window size, we can avoid the retransmission timeout with significant probability, regardless of the degree of the bandwidth decrease.

V. CONCLUSION

In this paper, we have proposed two methods to improve the performance of TCP Symbiosis, a *bandwidth-based* congestion control mechanism. The first one is a dynamic setting algorithm of the control parameter of Lotka-Volterra competition model, which TCP Symbiosis utilizes to regulate the congestion window size. Another one is to add the self-induced oscillation to the congestion window size to absorb the effect of the sudden environmental change. By applying these two methods, we have provided a new advantage to TCP Symbiosis: robustness to the measurement errors and environmental changes, which is important to deploy TCP Symbiosis to the actual Internet environment.

To the best of our knowledge, TCP Symbiosis is the first proposal of bandwidth-based congestion control mechanisms for TCP. However, we believe that the proposed methods in this paper can be applied to any other bandwidth-based congestion control mechanisms. In addition, we can also say that the proposed methods are effective to delay-based and hybrid approaches.

REFERENCES

[1] S. Shenker, L. Zhang, and D. D. Clark, “Some observations on the dynamics of a congestion control algorithm,” *ACM Computer Communication Review*, vol. 20, pp. 30–39, Oct. 1990.
[2] J. C. Hoe, “Improving the start-up behavior of a congestion control scheme of TCP,” *ACM SIGCOMM Computer Communication Review*, vol. 26, pp. 270–280, Oct. 1996.
[3] L. Guo and I. Matta, “The war between mice and elephants,” Tech. Rep. BU-CS-2001-005, Boston University, Computer Science Department, May 2001.
[4] Z. Fu, P. Zerfos, H. Luo, S. Lu, L. Zhang, and M. Gerla, “The impact of multihop wireless channel on TCP throughput and loss,” in *Proceedings of IEEE INFOCOM 2003*, Apr. 2003.
[5] S. Floyd, “HighSpeed TCP for large congestion windows,” *Request for Comments 3649*, Dec. 2003.
[6] T. Kelly, “Scalable TCP: Improving performance in highspeed wide area networks,” in *Proceedings of PFLDnet 2003*, Feb. 2003.
[7] I. Rhee and L. Xu, “CUBIC: A new TCP-friendly high-speed TCP variant,” in *Proceedings of PFLDnet 2005*, Feb. 2005.
[8] C. Jin, D. X. Wei, and S. H. Low, “FAST TCP: motivation, architecture, algorithms, performance,” in *Proceedings of IEEE INFOCOM 2004*, Mar. 2004.
[9] L. S. Brakmo and L. L. Peterson, “TCP Vegas: End to end congestion avoidance on a global Internet,” *IEEE Journal on Selected Areas in Communications*, vol. 13, pp. 1465–1480, Mar. 1995.
[10] K. T. J. Song, Q. Zhang, and M. Sridharan, “Compound TCP: A scalable and TCP-friendly congestion control for high-speed networks,” in *Proceedings of PFLDnet 2006*, Feb. 2006.
[11] H. Shimonishi, T. Hama, and T. Murase, “TCP-Adaptive Reno: Improving efficiency-friendliness tradeoffs of TCP congestion control algorithm,” in *Proceedings of PFLDnet 2006*, Feb. 2006.
[12] G. Hasegawa and M. Murata, “TCP Symbiosis: Congestion control mechanisms of TCP based on Lotka-Volterra competition model,” in *Proceedings of Inter-Perf 2006*, Oct. 2006.
[13] C. L. T. Man, G. Hasegawa, and M. Murata, “A simultaneous inline measurement mechanism for capacity and available bandwidth of end-to-end network path,” *IEICE Transactions on Communications*, vol. E89-B, pp. 2469–2479, Sept. 2006.
[14] J. D. Murray, *Mathematical Biology I: An Introduction*. Springer Verlag Published, 2002.
[15] The Network Simulation - NS2. available at http://www.isi.edu/nsnam/ns/.
[16] M. Kodama, G. Hasegawa, and M. Murata, “Implementation experiments of TCP Symbiosis: bio-inspired mechanisms for Internet congestion control,” in *Proceedings of CQR 2008*, Apr. 2008.
[17] V. Jacobson, “Modified TCP Congestion Avoidance Algorithm,” *end2end-interest mailing list*, Apr. 1990.
[18] A. F. ”B. Spagnolo, D.Valenti, “Noise in ecosystems: A short review,” *Mathematical Biosciences and Engineering*, vol. 1, pp. 185–211, June 2004.