INTERNATIONAL JOURNAL OF COMMUNICATION SYSTEMS
Int. J. Commun. Syst. 2005; **00**:1–18 *Prepared using* dacauth.cls *[Version: 2002/09/18 v1.01]*

Steady State and Transient State Behaviours Analyses of TCP Connections considering Interactions between TCP Connections and Network

Hiroyuki Hisamatsu[1], Hiroyuki Ohsaki[2] and Masayuki Murata[3]

[1] *Graduate School of Information Science and Technology, Osaka University 1-5, Yamadaoka, Suita, Osaka, Japan*
Phone: +81-6-6879-4542, Fax: +81-6-6879-4544 E-mail: hisamatu@ist.osaka-u.ac.jp
[2] *Graduate School of Information Science and Technology, Osaka University 1-5, Yamadaoka, Suita, Osaka, Japan*
Phone: +81-6-6879-4551, Fax: +81-6-6879-4554 E-mail: oosaki@ist.osaka-u.ac.jp
[3] *Graduate School of Information Science and Technology, Osaka University 1-5, Yamadaoka, Suita, Osaka, Japan*
Phone: +81-6-6879-4540, Fax: +81-6-6879-4544 E-mail: murata@ist.osaka-u.ac.jp

SUMMARY

The Internet uses a window-based congestion control mechanism in TCP (Transmission Control Protocol). In the literature, there have been a great number of analytical studies on TCP. Most of those studies have focused on the statistical behavior of TCP by assuming a constant packet loss probability in the network. However, the packet loss probability, in reality, changes according to the packet transmission rates from TCP connections. Conversely, the window size of a TCP connection is dependent on the packet loss probability in the network. In this paper, we explicitly model the interaction between the congestion control mechanism of TCP and the network as a feedback system. By using this model, we analyze the steady state and the transient state behaviors of TCP. We derive the throughput and the packet loss probability of TCP, and the number of packets queued in the bottleneck router. We then analyze the transient state behavior using a control theoretic approach, showing the influence of the number of TCP connections and the propagation delay on the transient state behavior of TCP. Copyright © 2005 John Wiley & Sons, Ltd.

KEY WORDS: *TCP (Transmission Control Protocol), congestion control, feedback system, fluid flow approximation, queueing theory, control theory, transient state behavior*

1. Introduction

A feedback-based congestion control mechanism is essential to realize an efficient data transfer services in a packed-switched network. In the current Internet, a sort of feedback-based congestion control mechanisms called TCP (Transmission Control Protocol) has been used. TCP has two mechanisms called *packet retransmission mechanism* and *congestion control mechanism*. The packet retransmission mechanism of TCP realizes reliable data transfer between source and destination hosts by keeping track of lost packets in the network. The congestion control mechanism of TCP, on the contrary, realizes efficient utilization of network resources by dynamically adjusting the number of in-flight packets.

The most-widely deployed implementation of TCP called *TCP Reno* uses a packet loss in the network as feedback information from the network since a packet loss implies congestion occurrence in the network [1]. The fundamental operation of TCP Reno is summarized as follows. Until a packet loss occurs in the network, TCP Reno gradually increases the window size of a source host. As soon as the window size exceeds the bandwidth-delay product (i.e., the available bandwidth × the round-trip delay), excess packets are queued at the buffer of an intermediate router. When the window size increases further, the buffer of the router overflows, resulting in a packet loss. At the source host, TCP Reno concludes there is packet loss after receiving more than three duplicate ACKs. TCP Reno then decreases the window size for resolving congestion. After the reduction of the window size, congestion in the network is relieved, and TCP Reno increases the window size of the source host again. By repeating this control indefinitely, TCP Reno tries to efficiently utilize network resources as well as to prevent congestion in the network.

In the literature, there have been a great number of analytical studies on TCP (e.g., [2-12]). Most of those studies assume a constant packet loss probability in the network, and derive the throughput of TCP connections [2, 8, 10, 12] or the distribution of window sizes of TCP connections [6, 7, 11]. However, the packet loss probability, in reality, changes according to packet transmission rates from TCP connections. Conversely, the window size of a TCP connection is dependent on the packet loss probability in the network. In this paper, we explicitly model the interaction between the congestion control mechanism of TCP and the network as a feedback system for investigating the steady state and the transient state behaviors of TCP. For modeling the congestion control mechanism of TCP, we use four different analytic models presented in [13-15]. As a network model, we use a $M/M/1/m$ queueing system, where the input traffic is mixture of TCP traffic and background traffic (i.e., non-TCP traffic).

In [12], the authors have analyzed the performance of TCP by modeling the network as a $M/D/1/m$ queuing system. However, the authors have focused only on the steady state behavior of TCP; that is, the transient state behavior of TCP has not been evaluated. In addition, their analytic model is not TCP Reno but TCP Tahoe, which does not have several important mechanisms found in TCP Reno. For instance, the effect of the fast retransmit mechanism in TCP Reno has not been investigated. In [4, 15], analytic models for TCP Reno and the RED (Random Early Detection) router have been presented, and the performance of TCP with the RED router has been analyzed. In [4], the primary focus of the analysis is on the steady state behavior of TCP. Only a qualitative discussion on the transient state behavior has been presented. In [15], a control theoretic approach has been taken to analyze the stability and the transient state behavior of TCP, where the RED router is modeled by a non-linear discrete-time system. On the other hand, the main objective of this paper is to analyze the transient state behavior of TCP with the Drop-Tail router, since most existing routers in the current Internet are Drop-Tail routers. We take a different approach of modeling the Drop-Tail router using a queuing theory.

In [3], the authors have derived the average file transfer time without assuming a constant packet loss probability in the network. However, the stability and the transient state behavior of TCP have not been analyzed. In [9], the authors have modeled the interactions of a set of TCP flows and AQM (Active Queue Management) gateways, and they have showed a transient state behavior of TCP. Since their methodology is based on the solution of the differential equations, they have not showed the transient state behavior of TCP, rigorously. In [5], the authors have analyzed a combined TCP and AQM (Active Queue Management) gateways model from a control theoretic standpoint. However, they have only focused on the stability of the queue length of RED gateways.

In our analytic model, both TCP traffic and background traffic are taken account of. We model the interaction between the congestion control mechanism of TCP and the network as a feedback system; that is, both the congestion control mechanism of TCP running on a source host and the network seen by TCP are modeled by dynamic systems (Fig. 1).

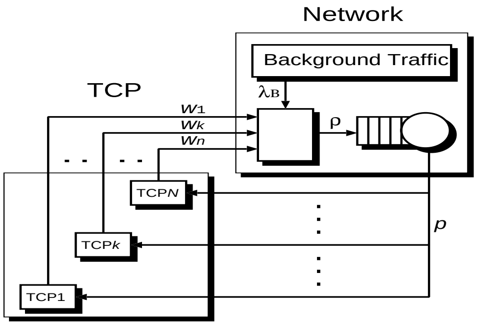


Figure 1: Analytic model as a feedback system consisting of TCP connections and network.

The congestion control mechanism of TCP is a window-based flow control mechanism, and it dynamically changes the window size according to occurrence of packet losses in the network. Hence, there exists a tendency that when the packet loss probability is small, the window size becomes large. On the contrary, when the packet loss probability is large, the window size tends to become small. We model the congestion control mechanism of TCP as a dynamic system, where the input to the system is the packet loss probability in the network and the output from the system is the window size of TCP.

On the other hand, the network seen by TCP behaves such that when the number of packets entering the network increases, some packets are waited at the buffer of the router destined for the bottleneck link. This sometimes causes buffer overflow, resulting in a packet loss. So the packet loss probability becomes large when the number of packets entering the network increases. Thus, the network seen by TCP can be modeled by a dynamic system, where the input to the system is the window size and the output from the system is the packet loss probability.

The main contribution of this paper is to allow steady state analysis of TCP, and more importantly, to analyze the transient state behavior of TCP using a rigorous manner based on control theory. We derive the throughput of each TCP connection, the packet loss probability at the bottleneck router, and the average queue length (i.e., the number of packets awaited in the buffer) at the bottleneck router. By utilizing the control theory, which has been developed in the control engineering, we analyze the transient state behavior of TCP. We show quantitatively how the stability and the transient state behavior of TCP are affected by several system parameters: the number of TCP connections, the propagation delay, the bottleneck link capacity, and the buffer size of the bottleneck router.

Organization of this paper is as follows. In Section 2, we describe how the interaction between TCP and the network can be modeled as a feedback system. We then present four analytic models of the congestion control mechanism of TCP. In Section 3, comparing these four analytic models with simulation results, we discuss which analytic model is most suitable for analyzing the transient state behavior of TCP. In Section 4, we derive the TCP throughput, the packet loss probability, and the queue length of the bottleneck router. We also validate our approximate analysis by comparing analytic results with simulation ones. In Section 5, we analyze the transient state behavior of TCP using a control theoretic approach. Finally, in Section 6, we conclude the current paper and discuss future works.

2. Analytic Model

In this section, we describe how the interaction between TCP and the network can be modeled as a feedback system. We first model the network using queueing theory, and then present four analytic models of the congestion control mechanism of TCP.

2.1. Modeling Network using Queuing Theory

We assume that there exists only a single bottleneck link in the network. In the followings, the router just before the bottleneck link is called *bottleneck router*. We also assume that the bottleneck router adopts a Drop-Tail discipline. Provided that the network is stationary, the bottleneck router can be modeled by a single queue. Thus, once the packet arrival rate and the capacity of the bottleneck router are known, the packet loss probability and the average waiting time can be obtained from the queuing theory. Since the packet departure process from a source host is oscillatory, in reality, the network is not stationary. However, as we will show in Section 3, the network seen by TCP can be well modeled by a queueing system at a relatively large time scale (e.g., the round-trip time). In the rest of this subsection, we formally describe how the network seen by TCP can be modeled by a queueing system.

Let N be the number of TCP connections, and w_i and r_i be the window size and the round-trip time of ith ($1 \leq i \leq N$) TCP connection. Assuming that each TCP connection continuously sends packets, the transmission rate from ith TCP connection can be approximated by w_i/r_i. The average packet arrival rate at the bottleneck router, λ, is therefore given by $\sum_{i=1}^{N} w_i/r_i + \lambda_B$, where λ_B is the average arrival rate of the background traffic at the bottleneck router. Let μ be the capacity of the bottleneck link, the offered traffic load at the bottleneck router ρ is given by $\rho = \lambda/\mu$. Depending on the packet arrival process, the distribution of the packet processing time, and the buffer capacity, there can be several queuing systems suitable for modeling the network seen by TCP. As a network model, we use a finite buffer queuing system, $M/M/1/m$, where m represents the buffer size of the bottleneck router. Namely, the packet loss probability at the bottleneck router is given by

$$p = \frac{(1-\rho)\,\rho^m}{1-\rho^{m+1}} \tag{1}$$

where ρ and r are given by

$$\begin{aligned}\rho &= \frac{1}{\mu}\left(\frac{\sum_{i=1}^{N} w_i}{r_i} + \lambda_B\right)\\ r_i &= 2\tau_i + \frac{\rho\,(1 - m\,\rho^m + m\,\rho^{m+1})}{\mu(1-\rho^{m+2})\,(1-\rho)}\end{aligned}$$

Use of the queuing model to analyze the steady state behavior of TCP is straightforward and promising. However, the application of the queueing model to analyze the transients state behavior requires the careful treatment since the queueing model was originally developed for analyzing not the dynamic behavior but the statistical behavior. However, we believe that the queueing model can give some insight on dynamic systems. The applicability of our approximate analysis will be validated in the latter sections by comparing analytic results with simulations ones.

2.2. Modeling TCP using Different Approaches

The congestion control mechanism of TCP is quite complicated since it performs several control mechanisms such as detecting packet losses in the network and retransmitting lost packets. It is therefore impossible to build an exact analytic model of TCP. In this paper, we model only the main part of the congestion control mechanism of TCP, and ignore the rest; that is, we model the essential behavior of TCP (i.e., the window-based flow control mechanism and the loss recovery mechanism including the fast retransmit mechanism of TCP Reno) in its congestion avoidance phase. In our analytic model, several TCP mechanisms, such as the slow-start phase, the Nagle algorithm, and the delayed ACK, are not modeled. Since our main focus is in long-lived TCP connections, we assume that the TCP traffic is persistent. Under this assumption of persistent traffic, effect of these unmodeled factors could be negligible.

In [14-16], several analytic models for the congestion avoidance phase of TCP have been presented, describing the relation between the packet loss probability in the network and the resulting window size of TCP. In what follows, we introduce four analytic models called A, A', B, and C, which are derived from different modeling approaches. In Section 3, we will discuss which model is suitable for analyzing the transient state behavior of TCP.

- **Model A**

In [14], by assuming a constant packet loss probability in the network (denoted by p), the authors have presented an analytic model describing the window size of a TCP connection in steady state. The authors have derived the average throughput of a TCP connection. In this model, the authors assume that the initial window size at the beginning of a congestion avoidance phase is equal to that at the beginning of the next congestion avoidance phase, and that TCP sends the number $1/p$ of packets in each congestion avoidance phase. In summary, the average throughput of a TCP connection, λ_T, is derived as

$$\lambda_T = \frac{\frac{1-p}{p} + E[W] + \hat{Q}(E[W])\frac{1}{1-p}}{r\left(\frac{b}{2}E[W]+1\right) + \hat{Q}(E[W])T_o\frac{f(p)}{1-p}}$$

where

$$\begin{aligned} E[W] &= \frac{2+b}{3b} + \sqrt{\frac{8(1-p)}{3bp} + \left(\frac{2+b}{3b}\right)^2} \\ \hat{Q}(w) &= \frac{(1-(1-p)^3)(1+(1-p)^3(1-(1-p)^{w-3}))}{(1-(1-p)^w)} \\ f(p) &= 1+p+2p^2+4p^3+8p^4+16p^5+32p^6 \end{aligned}$$

and r is the average round-trip time of the TCP connection, and b is a parameter of delayed ACKs (i.e., a destination host returns an ACK packet for every b data packets). T_O is the length of TCP's retransmission timer. $\hat{Q}(w)$ is a probability that when the window size is w, the source host fails to detect a packet loss from duplicate ACKs. From these equations, the window size of TCP in steady state, w_A, is given by

$$w = \lambda_T\, r \tag{2}$$

- **Model A$'$**

When the packet loss probability is very small ($p << 1$), Eq. (2) is approximated as [14]

$$w \simeq \sqrt{\frac{3}{2bp}} \tag{3}$$

• **Model B** In [16], the authors have analyzed a congestion control mechanism using ECN (Explicit Congestion Notification). ECN is a mechanism to explicitly notify source hosts of congestion occurrence in the network. When a router experiences congestion, by setting the CE bit of arriving packets, it informs destination hosts of the congestion occurrence. Then the destination host informs the source host of the congestion occurrence by setting the ECN echo bit of ACK packets In [16], the authors assume that the ECN echo bit of an ACK packet is set with a probability of p_E, and have derived a state transition equation for the window size. Let $w(k)$ be the window size at slot k (i.e., the time when kth ACK packet is received). Their analytic model is different from TCP; that is, when the ECN echo bit is set, the source host linearly increases the window size by $I(w(k))$. Otherwise, it multiplicatively decreases the window size by $D(w(k))$. By calculating the expected value of the window size at each receipt of an ACK packet, the evolution of the window size is given by

$$w(k) = w(k-1) + (1-p_E)\, I(w(k-1)) - p_E\, D(w(k-1)) \tag{4}$$

The analytic model presented in [16] is not for TCP, but can be easily applied. Namely, an ACK packet with the ECN echo bit not set corresponds to a non-duplicate ACK in TCP (i.e., indication of no congestion). Similarly, an ACK packet with the ECN bit set corresponds to duplicate ACKs (i.e., indication of congestion). Thus, when the packet loss probability is p, the state transition equation for the window size, w_B, is given by

$$\begin{aligned} w(k) &= w(k-1) + (1-p)\,\frac{1}{w(k-1)} \\ &\quad - p\,(1-\hat{Q}(w(k-1)))\,\frac{w(k-1)}{2} - p\,\hat{Q}(w(k-1))\,(w(k-1)-1) \end{aligned} \tag{5}$$

Note that we modify and extend Eq. (4) to include the timeout mechanism of TCP.

• **Model C**

In [15], the authors have derived the state transition equation for the window size in the congestion avoidance phase of TCP. This analytic approach uses a discrete-time model, where a time slot corresponds to the duration between two succeeding packet losses. However, their analytic model is not for the Drop-Tail router but for the RED router, where the router randomly discards arriving packets. In what follows, we describe a modification to the analytic model presented in [15] for analyzing TCP with the Drop-Tail router.

In [15], the authors have derived $\overline{X}(k)$, the expected number of packets passing through the RED router at slot k as

$$\overline{X}(k) = \frac{1/p_b(k)+1}{2}$$

where $p_b(k)$ is the packet dropping probability of the RED router at slot k. Let p be the packet loss probability of the Drop-Tail router, $\overline{X}(k)$ is changed to

$$\overline{X}(k) = \sum_{n=1}^{\infty} n(1-p)^{n-1}p = \frac{1}{p}$$

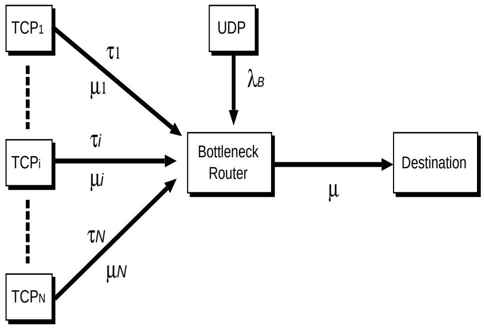


Figure 2: Simulation model of 10 TCP connections and a single bottleneck link.

Thus, when the packet loss probability is p, the window size w at the beginning of slot k is obtained as [15]

$$w(k) \quad = \quad \frac{1}{4}\left\{-1+\sqrt{(1-2w(k-1))^2+\frac{8}{p}}\right\} \tag{6}$$

Note that Eq. (6) is derived by assuming that a packet loss probability is constant in a slot. Since the packet loss probability is, in reality, increased as the window size increases, this analytic model might overestimate the window size.

We note that models A and A' are built based on the window size in steady state. It is therefore expected that these models are not suitable for analyzing the transient state behavior of TCP. On the contrary, models B and C describe the dynamic behavior of the window size in the congestion avoidance phase. Thus, it is expected that models B and C are suitable for analyzing the transient state behavior of TCP. In the next section, we compare these four analytic models using numerical and simulation results.

3. Model Validation with Simulation

3.1. Simulation Model

The simulation model is shown in Fig. 2. In this model, 10 TCP connections share the bottleneck link. The propagation delay of ith TCP connection is $5 + i$ [ms], and the link capacity from the ith source host to the router is $5 + 0.5i$ [packet/ms]. We model the background traffic as UDP packets, where the packet arrival of UDP packets is modeled by a Poisson process with the average arrival rate of $\lambda_B = 2$ [packet/ms]. Unless explicitly noted, we use the following parameters in all simulations: both TCP and UDP packet sizes are fixed at 1000 [byte], the capacity of the bottleneck link μ is 5 [packet/ms], and the propagation delay of the bottleneck link τ is 5 [ms]. Note that with these simulation parameters, 1 [packet/ms] corresponds to about 8 [Mbit/s]. Also note that performance evaluation for fixed-sized TCP packets would be sufficient since we assume persistent TCP traffic in our analysis. We run every simulation for 30 seconds using ns-2 [17].

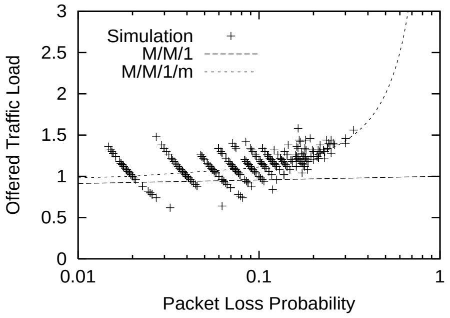


Figure 3: Comparison of $M/M/1/m$ queuing system with simulation result.

3.2. Network Models

Figure 3 shows the relation between the offered traffic load and the packet loss probability. These values are measured at the bottleneck router for every 10 [ms]. Namely, these values are rough estimation of the *instantaneous* offered traffic load and the *instantaneous* packet loss probability. In the figure, the packet loss probabilities obtained from well-known results of $M/M/1/m$ and $M/M/1$ are also plotted. This figure shows that the dynamics of the network at a relatively small time scale can be well modeled by the $M/M/1/m$ model. Note that the queuing theory is for analyzing the statistical behavior, not the dynamical behavior. Note also that UDP and TCP packet sizes are fixed at 1000 [byte]. This figure indicates that $M/M/1/m$ could be usable for analyzing the transient state behavior of TCP. However, simulation results are scattered around the result of $M/M/1/m$. This means that the packet loss probability has a variability even when the offered traffic load at the bottleneck router is fixed.

3.3. TCP Models

By comparing with simulation results, we discuss how accurately four analytic models of TCP capture the relation between the window size and the packet loss probability. Figure 4 shows the relation between the packet loss probability and the window size obtained using models A, A', B and C, respectively. In this figure, the window size for a given packet loss probability is obtained using Eqs. (2), (3), (5), and (6). Note that in the model A, the analytic result is calculated by assuming no timeout (i.e., $\hat{Q}(w) = 0$). Also note that in the model C, Eq. (6) gives the window size at the beginning of a slot, and not the average window size. For comparison purposes, the average window size is calculated and plotted in the figure. Refer to [15] for more detail. We also plot simulation results; that is, points corresponding to the average window size and the packet loss probability. As with Fig. 3, these values are *instantaneous values* of the average window size and the packet loss probability, which are measured at the bottleneck router for every 1 [s]. This figure shows that when the packet loss probability is less than 0.02, analytic results obtained from models A, A', and B show good agreement with simulation results. On the other hand, when the packet loss probability is more than 0.03, analytic results obtained from models B and C show good agreement.

From these observations and discussions, we choose the model B for analyzing the steady state and the transient state behaviors of TCP. Regarding the steady state behavior, the model B shows good agreement with simulation results in a certain range of packet loss probabilities. Although models

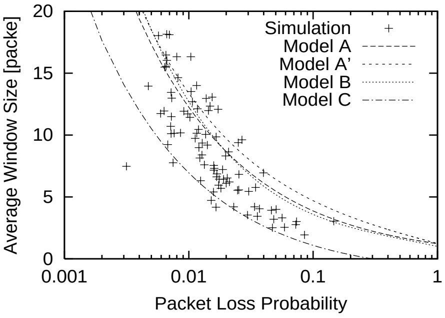


Figure 4: Comparison of four TCP models with simulation results.

A and A' give close analytic results with those with the model B, models A and A' should not be appropriate for analyzing the transient state behavior of TCP since these models are based on the steady state analysis of TCP. In the following sections, with using the model B, we will derive the TCP throughput, the packet loss probability, and the queue length of the bottleneck router. We will also analyze the transient state behavior of TCP using a control theoretic approach.

4. Steady State Analysis

In what follows, we present steady state analysis for the combination of the model B for TCP and a $M/M/1/m$ queuing system for the network. As have been explained in Section 2.2, the model B describes the change of the window size every receipt of an ACK packet. Hence, in the following analysis, the duration between two succeeding ACK packets corresponds to a unit time. In this section, we derive the TCP throughput, the packet loss probability, and the average queue length in steady state. We then validate our approximate analysis by comparing analytic results with simulation ones.

The congestion control mechanism of TCP is an AIMD (Additive Increase and Multiplicative Decrease) based feedback control. When the propagation delay is non-negligible, the window size oscillates and never converges to a constant value. Note that the symbol $w(k)$ represents not the instant value of the oscillating TCP window size but the expected value of the TCP window size after a long period.

Let equilibrium values of the TCP window size and the packet loss probability in the network be w^* and p^*, respectively; i.e.,

$$w^* \equiv \lim_{k\to\infty} w(k) \tag{7}$$

$$p^* \equiv \lim_{k\to\infty} p(k) \tag{8}$$

These values can be numerically obtained by solving Eqs. (1) and (5) with equating $w(k+1) \equiv w(k)$ and $p(k+1) \equiv p(k)$. Using these equilibrium values, the TCP throughput T and the average queue

Table I. Parameter values

$N = 10$	$\mu = 2$ [packet/ms]
$\tau = 30$ [ms]	$\lambda_B = 0.2$ [packet/ms]
$m = 50$ [packet]	packet size = 1000 [byte]

length of the bottleneck router L are given by

$$T = \frac{w^*}{r^*} \tag{9}$$

$$\begin{aligned} L &= \rho^* \mu (r^* - 2\tau) \\ &= \frac{\rho^{*2} \left(1 - m\,\rho^{*m} + m\,\rho^{*(m+1)}\right)}{\left(1 - \rho^{*(m+2)}\right)\left(1 - \rho^*\right)} \end{aligned} \tag{10}$$

where r^* and ρ^* are the equilibrium values of $r(k)$ and $\rho(k)$, respectively and decided by w^* and p^*. In the above equations, the TCP throughput T is approximated by the number of packet per a unit time emitted by source host, and the average queue length L is obtained from the number of customers waiting to be served of the $M/M/1/m$ queue.

We next compare analytic results with simulation ones for validating our approximate analysis. In the following analytic results, we calculate the TCP throughput T, the packet loss probability p^*, and the average queue length L using Eqs. (9), (8), and (10), respectively. Using ns-2 simulator, we run several simulation experiments at a packet level for the same network model with Fig. 1. Each simulation experiment is continued for 24 seconds, and the last 20 seconds are used for calculating simulation results — the TCP throughput, the packet loss probability, and the average queue length. Each simulation experiment is repeated 50 times, and 95 % confidence intervals of all performance measures are calculated. Note that our analytic model uses fluidflow approximation whereas simulation results are obtained using a packet-level network simulation.

In obtaining the analytic and simulation results, we use the following parameters: the number of TCP connections $N = 10$, the bottleneck link capacity $\mu = 2$ [packet/ms], the propagation delay $\tau = 30$ [ms], the average arrival rate of the background traffic $\lambda_B = 0.2$ [packet/ms], and the buffer size of the bottleneck router $m = 50$ [packet]. In simulation experiments, we model the background traffic by UDP traffic. The packet size of TCP and UDP packets is fixed at 1000 [byte]. The maximum window size of all TCP connections is fixed at a sufficiently large value, 10,000 [packets]. We use TCP version Reno on all source hosts. Table I summarizes parameters used in obtaining the analytic and simulation results.

Figure 5 shows the TCP throughput, the packet loss probability, and the average queue length for the different bottleneck link capacities. For comparison purposes, another analytic result of the TCP throughput from [14] is shown in Fig. 5(a). In [14], the TCP throughput is derived as a function of the round-trip time and the packet loss probability for a TCP connection. More specifically, the TCP throughput T' derived in [14] is given by

$$T' = \frac{\frac{1-p}{p} + E[W] + \hat{Q}(E[W])\frac{1}{1-p}}{r\left(\frac{b}{2}E[W] + 1\right) + \hat{Q}(E[W])T_o\frac{f(p)}{1-p}}$$

where

$$\begin{aligned} E[W] &= \frac{2+b}{3b} + \sqrt{\frac{8(1-p)}{3bp} + \left(\frac{2+b}{3b}\right)^2} \\ \hat{Q}(w) &= \frac{(1-(1-p)^3)(1+(1-p)^3(1-(1-p)^{w-3}))}{(1-(1-p)^w)} \\ f(p) &= 1+p+2p^2+4p^3+8p^4+16p^5+32p^6 \end{aligned}$$

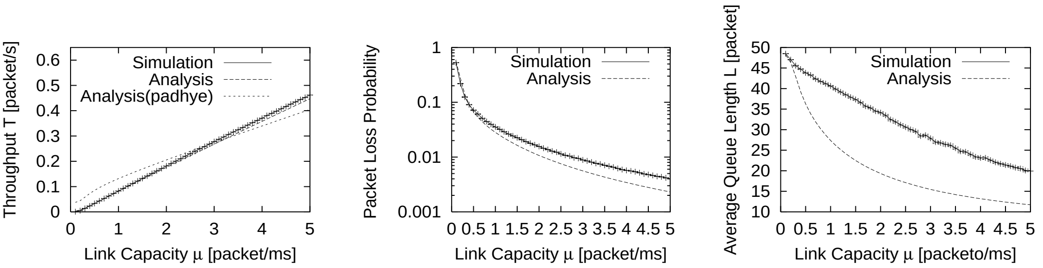
In this paper, we calculate the TCP throughput from the above equation using the packet loss probability and the round-trip time obtained from the simulation. It can be found, in terms of the TCP throughput and the packet loss probability, both analytic and simulation results show a good agreement. In particular, in respect to the TCP throughput, it can be found that our analytic results show better agreement with the simulation results than the value obtained from the expression in [14]. However, in terms of the average queue length, it can be found that our analytic results are much smaller than simulation results. Such a disagreement between analytic and simulation results is probably caused by our assumption that the packet arrival at the bottleneck router follows a Poisson process. In running the simulation, the average arrival rate of the background traffic is fixed at $\lambda_B = 0.2$ [packet/ms]. Hence, the amount of the TCP traffic becomes relatively larger than the amounts of the background traffic as the bottleneck link capacity becomes large. As a result, the packet arrival process at the bottleneck router cannot be modeled by a Poisson process.

In Fig. 6, both analytic and simulation results are shown for different propagation delays. Similarly to the previous case, it can be found that both analytic and simulation results show a good agreement in terms of the TCP throughput. However, as the propagation delay increases, the packet loss probability obtained from our analysis deviates from the corresponding simulation result. It can also be found that, in respect to the average queue length of the bottleneck router, our analytic results are much smaller than simulation ones. Such a disagreement between analytic and simulation results is probably caused by our assumption that the packet arrival at the bottleneck router follows a Poisson process. Since TCP uses a window-based flow control mechanism, the packet emission process from the source host becomes more bursty as the propagation delay becomes large. Hence, as the propagation delay becomes large, the Poisson process becomes insufficient for modeling the arrival process of the background traffic at the bottleneck router.

5. Transient State Behavior Analysis

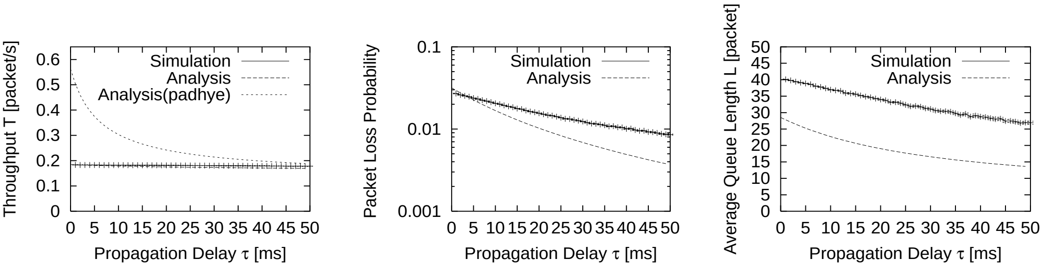
Using the analytic model presented in Section 2, we analyze the transient state behavior of TCP in the congestion avoidance phase. By the word *transient state behavior*, we mean the dynamics of the window size from its initial value to its equilibrium value. TCP changes the window size according to the occurrence of a packet loss in the network. Since a packet loss occurs probabilistically, the window size can be thought of as a random variable. By focusing on the *average behavior* of TCP, we analyze the transient state behavior of TCP. More specifically, we analyze the transient state behavior of TCP by investigating how the expected value of the window size changes.

The state of the network at slot k is then fully described by the window size $w(k)$ and the packet loss probability $p(k)$. For given initial values of the window size and the packet loss probability, the evolution of the window size and the packet loss probability can be numerically obtained. Recall that $w(k)$ is not the instant value of the window size, but the average value of the window size. Using these



(a) TCP throughput (b) Packet loss probability (c) Average queue length

Figure 5: Analytic and simulation results for different bottleneck link capacities.



(a) TCP throughput (b) Packet loss probability (c) Average queue length

Figure 6: Analytic and simulation results for different propagation delays.

equations and calculating the evolutions of $w(k)$ and $p(k)$, the transient state behavior of TCP can be analyzed. We next present several numerical examples, showing how the amount of background traffic λ_B and the propagation delay τ of the bottleneck link affect the transient state behavior of TCP. In the following numerical examples, unless explicitly noted, the initial window size is 1 [packet], the initial packet loss probability is 0, the number of TCP connections N is 10, the capacity of the bottleneck link μ is 5 [packet/ms], the propagation delay τ is 15 [ms], and the buffer size of the bottleneck router m is 50 [packet].

Figure 7 shows the evolution of the window size in the congestion avoidance phase for the amount of background traffic λ_B of 0, 2.0, and 4.5 [packet/ms]. From this figure, one can find that the window size in steady state becomes small as the amount of background traffic increases, indicating that TCP suffers less throughput. One can also find that the convergence speed (i.e., in this case, the increase rate of the window size) of the window size is independent of the amount of background traffic. This is because, in the congestion avoidance phase, TCP increases the window size by one packet per a round-trip time, which is essentially irrelevant to the TCP throughput.

Figure 8 shows the evolution of the window size in the congestion avoidance phase for the propagation delay τ of 10, 30, and 50 [ms]. One can find that the window size becomes large as the

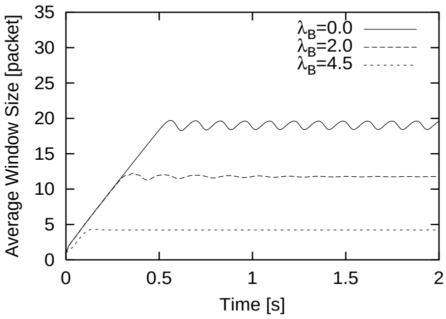


Figure 7: Transient state behavior of TCP for different amount of background traffic.

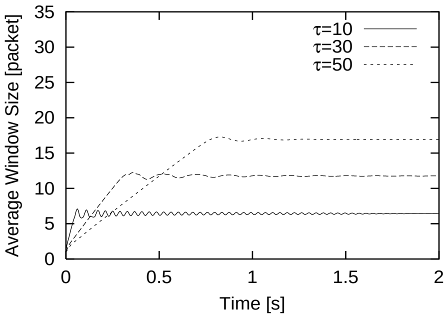


Figure 8: Transient state behavior of TCP for different propagation delay of bottleneck link.

propagation delay increases. This can be intuitively understood from the increased bandwidth-delay product. In addition, as the propagation delay becomes large, one can find that the convergence speed of the window size becomes slow, and that the ramp-up time of the window size becomes short. In general, as the feedback delay becomes large, the transient state behavior is degraded and the system becomes less stable. However, the contrary occurs when the propagation delay τ is small (e.g.,10 [ms]), the window size oscillates for long (e.g., more than 1.5 [s]). This is because, from a control theoretical viewpoint, the feedback gain in the congestion avoidance phase of TCP is changed according to the round-trip time. Namely, in the congestion avoidance phase of TCP, the window size is incremented by one packet for every round-trip time. Thus, increasing the propagation delay implies decreasing the feedback gain.

We then analyze the TCP behavior in transient state using state transition equations. Specifically, by applying the control theory, we show how the TCP window size and the packet loss probability converge to their equilibrium points.

Let $\mathbf{x}(k)$ be the difference between $(w(k), p(k))$ and (w^*, p^*).

$$\mathbf{x}(k) \equiv \left[\begin{array}{ccc} w(k) & - & w^* \\ p(k) & - & p^* \end{array} \right]$$

Since Eqs. (1) and (5) have non-linearity, we linearize them around their equilibrium points and write them in a matrix form

$$\mathbf{x}(k+1) = \mathbf{A}\,\mathbf{x}(k) \tag{11}$$

where **A** is a state transition matrix. Eigenvalues of the state transition matrix determine the stability and the transient state behavior of the feedback system around the equilibrium point [18]. It is known that the system is stable if the maximum modulus is less than one. It is also known that the smaller the maximum modulus is, the better the transient state behavior becomes. In the followings, we show several numerical examples to reveal how the stability and the TCP transient state behavior are affected by several system parameters — the number of TCP connections, the propagation delay, the bottleneck link capacity, and the buffer size of the bottleneck router.

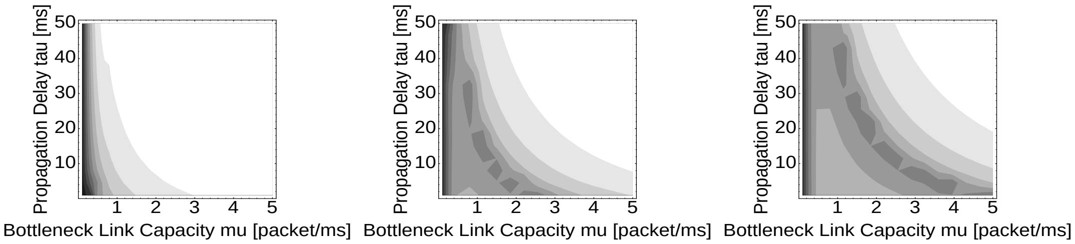
Figure 9 and 10 shows the contour plot of the maximum modulus of the eigenvalues. The figure means that the maximum modulus of the eigenvalues becomes large as the filled color becomes dark. A white area is an area where the maximum modulus of eigenvalues is more than 1.0 (i.e., unstable area). Figure 9 shows the maximum modulus of the eigenvalues for different numbers of TCP connections of N = 5, 10, and 15. In this figure, we plot the maximum modulus of eigenvalues of the state transient matrix **A** for different bottleneck link capacities of μ = 0–5 [packet/ms] and propagation delays of τ = 0–5 [ms]. The buffer size of the bottleneck router m is fixed at 50 [packet] and the average arrival rate of the background traffic λ_B is fixed at 0.2 [packet/ms].

From Fig. 9, one can find that the maximum modulus of the eigenvalues is mostly determined by $\mu \times \tau$. This indicates that the stability and the transient state behavior of TCP are determined by the bandwidth–delay product. This is because the congestion control mechanism of TCP is a window-based mechanism, and it changes the window size at every receipt of an ACK packet. Provided that the packet size is fixed, the number of ACK packets in the network during a round-trip time is proportional to the bandwidth–delay product. In the control engineer's view, the increase of the propagation delay means the decrease of the feedback gain or the feedback delay. Hence, the stability and the transient state behavior of TCP are determined by the bandwidth–delay product.

By comparing Figs. 9(a)–(c), one can find that as the number of TCP connections increases, the stability region becomes large. This is because the larger the number of TCP connections becomes, the smaller the bandwidth–delay product of each TCP connection becomes. The small bandwidth–delay product means that a source host receives a small number of ACK packets which carry feedback information. As a result, the increase of the number of TCP connections has the same effect with the decrease of the feedback delay and/or the feedback gain.

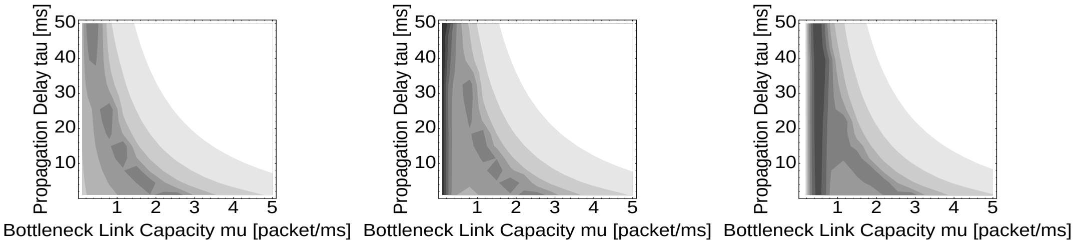
Figure 10 shows the maximum modulus of eigenvalues for the number of TCP connections $N = 10$ and different arrival rate of the background traffic, λ_B = 0, 0.2 and 0.5 [packet/ms]. By comparing Fig. 10(a)–(c), one can find that the stability region becomes slightly larger, as λ_B becomes large. This is because the increase of the background traffic corresponds to the decrease of the available bandwidth to TCP connections. Namely, the decrease of the available bandwidth to TCP connections results in the smaller bandwidth–delay product, which means a little feedback gain. Because the little feedback gain makes system sensitivity to the changes of the environment low, the reduction of the available bandwidth would bring the larger stability region.

To validate our transient state behavior analysis, we next show how the TCP transient state behavior changes for different maximum moduli using simulation experiments. Figure 11 shows the window size and the average queue length obtained from our simulation experiments for different bottleneck link capacities μ = 0.5, 2.0, and 5.0 [packet/ms]. Note that when the bottleneck link capacity μ is 0.5, 2.0, and 5.0, the maximum modulus of eigenvalues of the state transient matrix is 0.619, 0.780, and



(a) $N = 5$ (b) $N = 10$ (c) $N = 15$

Figure 9: Maximum modulus of eigenvalues for different number of TCP connections.



(a) $\lambda_B = 0$ [packet/ms] (b) $\lambda_B = 0.2$ [packet/ms] (c) $\lambda_B = 0.5$ [packet/ms]

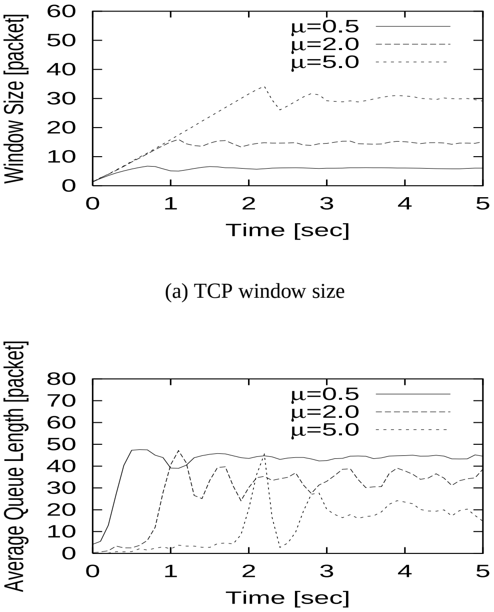
Figure 10: Maximum modulus of eigenvalues for different arrival rate of background traffic.

0.923, respectively. We use the same values with Tab. I for all parameters except the bottleneck link capacity.

Using the ns-2 simulator, we run simulations 50 times at a packet level for the same network model shown in Fig. 1, and investigate the evolution of the average TCP window size and the average queue length. More specifically, we calculate the average TCP window size and the average queue length every 100 ms. From this figure, one can find that the smaller the maximum modulus is (the smaller the bottleneck link capacity is), the better the transient state behavior becomes. From these observations, we conclude that our transient state behavior analysis using the control theory accurately captures the dynamics of TCP.

6. Conclusion and Future Work

In this paper, we have modeled both the congestion control mechanism of TCP and the network as a feedback system, and have analyzed the steady state and the transient state behaviors of TCP. We have derived the throughput of each TCP connection, the packet loss probability, and the average queue



(a) TCP window size

(b) Average queue length

Figure 11: Simulation results for different maximum moduli.

length at the bottleneck router. We have also analyzed the TCP transient state behavior by using the control theory. As a result, we have found that the bandwidth–delay product mostly determines the stability and the transient state behavior of TCP. We have also found that the network becomes stable as the number of TCP connections or the amounts of the background traffic increases. We have shown that the transient state behavior is heavily dependent on the propagation delay of the bottleneck link, but is almost independent of the amount of background traffic. As future work, it would be interesting to apply our approach to the more general heterogeneous network where several bottleneck routers exist, and improve the accuracy of the approximate analysis by using the rigorous analytic model of a $M/D/1/m$ queue.

REFERENCES

1. V. Jacobson and M. J. Karels, “Congestion avoidance and control,” in *Proceedings of ACM SIGCOMM ’88*, pp. 314–329, Nov. 1988.
2. B.Sikdar, S.Kalyanaraman, and K.S.Vastola, “Analytic models and comparative study of latency and steady-state throughput of TCP Tahoe, Reno nad SACK,” *IEEE/ACM Transactions on Networking*, vol. 11, pp. 959–971, Dec. 2003.
3. E.Alessio, M.Garetto, R.Cigno, M.Meo, and M.Marsan, “Analytical estimation of completion times of mixed NewReno and Tahoe TCP connections over single and multiple bottleneck networks,” in *Proceedings of IEEE GLOBECOM2001*, Nov. 2001.

4. V. Firoiu and M. Borden, “A study of active queue management for congestion control,” in *Proceedings of IEEE INFOCOM 2000*, pp. 1435–1444, Mar. 2000.
5. C. V. Hollot, V. Misra, D. F. Towsley, and W. Gong, “A control theoretic analysis of RED,” in *Proceedings of IEEE INFOCOM2000*, pp. 1510–1519, Mar. 2000.
6. A. Kumar, “Comparative performance analysis of versions of TCP in a local network with a lossy link,” *IEEE/ACM Transactions on Networking*, vol. 6, no. 4, pp. 485–498, 1998.
7. M. Mathis, J. Semke, and J. Mahdavi, “The macroscopic behavior of the TCP congestion avoidance algorithm,” *ACM SIGCOMM Communication Review*, vol. 27, July 1997.
8. A. Misra and T. J. Ott, “The window distribution of idealized TCP congestion avoidance with variable packet loss,” in *Proceedings of IEEE INFOCOM ’99*, pp. 1564–1572, Mar. 1999.
9. 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 *Proceedings of ACM SIGCOMM*, pp. 151–160, Aug. 2000.
10. T. J. Ott, J. Kemperman, and M. Mathis, “The stationary behavior of ideal TCP congestion avoidance,” 1996.
11. J. Padhye, V. Firoiu, and D. Towsley, “A stochastic model of TCP Reno congestion avoidance and control,” tech. rep., CMPSCI Technical Report 99-02, 1999.
12. C. Casetti and M. Meo, “A new approach to model the stationary behavior TCP connections,” in *Proceedings of IEEE INFOCOM 2000*, pp. 367–375, Mar. 2000.
13. J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, “Modeling TCP Reno performance: a simple model and its empirical validation,” *IEEE/ACM Transactions on Networking*, vol. 8, pp. 133–145, Apr. 2000.
14. J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, “Modeling TCP throughput: a simple model and its empirical validation,” in *Proceedings of ACM SIGCOMM ’98*, pp. 303–314, Sept. 1998.
15. H. Ohsaki and M. Murata, “Steady state analysis of the RED gateway: stability, transient behavior, and parameter setting,” *IEICE Transactions on Communications*, vol. E85-B, pp. 107–115, Jan. 2002.
16. T. J. Ott, “ECN protocols and the TCP paradigm,” June 1999. available at `http://web.njit.edu/˜ott/Papers/ECN/ECN.pdf`.
17. “The network simulator – ns2.” available at `http://www.isi.edu/nsnam/ns/`.
18. H. Ohsaki, M. Murata, T. Ushio, and H. Miyahara, “Stability analysis of window-based flow control mechanism in TCP/IP networks,” *1999 IEEE International Conference on Control Applications*, pp. 1603–1606, Aug. 1999.

AUTHORS’ BIOGRAPHIES

Hiroyuki Hisamatsu received the M. E. degree in the Information and Computer Sciences from Osaka University, Osaka, Japan, in 2003. He is currently a graduate student at the Graduate School of Information Science and Technology, Osaka University. His research work is in the area of performance evaluation of TCP/IP networks. He is a student member of IEEE and IEICE.

Hiroyuki Ohsaki received the M. E. degree in the Information and Computer Sciences from Osaka University, Osaka, Japan, in 1995. He also received the Ph. D. degree from Osaka University, Osaka, Japan, in 1997. He is currently an associate professor at Department of Information Networking, Graduate School of Information Science and Technology, Osaka University, Japan. His research work is in the area of traffic management in high-speed networks. He is a member of IEEE and Institute of Electronics, Information, and Computer Engineers of Japan (IEICE). His e-mail address is oosaki@ist.osaka-u.ac.jp

Masayuki Murata received the M.E. and D.E. degrees from Osaka University, Japan, in 1984 and 1988, respectively. In 1984, he joined Tokyo Research Laboratory, IBM Japan, as a Researcher. He was an Assistant Professor of Computation Center, Osaka University from 1987 to 1989, an Assistant Professor of Faculty of Engineering Science from 1989 to 1992, an Associate Professor of Graduate School of Engineering Science from 1992 to 1999. From 1999, he has been a Professor of Osaka University. He moved to Graduate School of Information Science and Technology in 2004. He has more than three hundred papers of international and domestic journals and conferences. His research interests include computer communication networks, performance modeling and evaluation. He is a member of IEEE, ACM, The Internet Society, IEICE and IPSJ.