# Steady state and transient state analyses of TCP and TCP-friendly rate control mechanism using a control theoretic approach

Hiroyuki Hisamatsu, Hiroyuki Ohsaki and Masayuki Murata

Graduate School of Information Science and Technology, Osaka University
1-5 Yamadaoka, Suita, Osaka, Japan

## ABSTRACT

In recent years, various real-time applications in the Internet have been emerging with rapid increase of the network bandwidth. A real-time application traditionally uses either UDP (User Datagram Protocol) or TCP (Transmission Control Protocol) as its transport layer protocol. However, using either UDP or TCP is insufficient for most real-time applications because of lacking a smooth rate control mechanism or suffering a significant transfer delay. In the literature, several transport-layer communication protocols for real-time applications have been proposed. In this paper, among these transport-layer communication protocols, we focus on TFRC (TCP-Friendly Rate Control). Steady state performances of TCP and TFRC connections such as throughput and fairness have been throughly investigated by many researchers using simulation experiments. However, transient state properties of TCP and TFRC connections such as stability and responsiveness have not been investigated. In this paper, we therefore analyze both steady state and transient state performances of TCP and TFRC connections using a control theoretic approach. We frist model TFRC and TCP connections with different propagation delays and the active queue management mechanism of RED (Random Early Detection) router as independent discrete-time systems. By combining these discrete-time systems, we analyze steady state performance of TCP and TFRC connections such as throughput, transfer delay, and packet loss probability. We also analyze transient state performance of TCP and TFRC connections using linearization of discrete-time systems around their equilibrium points.

**Keywords:** TFRC (TCP-Friendly Rate Control), TCP (Transmission Control Protocol), Steady state analysis, Transient state analysis, Control theory

## 1. INTRODUCTION

In recent years, real-time applications such as video streaming applications have been widely deployed with increase in the transmission speed of the Internet. A real-time application uses either UDP (User Datagram Protocol) or TCP (Transmission Control Protocol) as its transport layer protocol. The Internet is a best-effort network where multiple users share the network bandwidth. Hence, all network applications should have a mechanism for adapting to the congestion status of a network. However, UDP is a simple protocol for datagram transfer, and does not have a congestion control mechanism. When a real-time application uses UDP as its

---

Further author information: (Send correspondence to Hisamatsu)
H.H.: E-mail: hisamatu@ist.osaka-u.ac.jp, Telephone: +81 6 6879 4542
O.H.: E-mail: oosaki@ist.osaka-u.ac.jp, Telephone: +81 6 6879 4551
M.M.: E-mail: murata@ist.osaka-u.ac.jp, Telephone: +81 6 6879 4540

transport layer protocol, to prevent congestion collapse of a network, it is necessary to implement some congestion control mechanism at on application layer.

On the contrary, TCP has a mechanism for adjusting its packet transmission rate according to the measured available bandwidth of a network by performing congestion control between source and destination hosts. However, TCP is a transport-layer communication protocol originally designed for data transfer applications that can tolerate a certain amount of delay. Since the congestion control mechanism of TCP is an AIMD (Additive Increase Multiplicative Decrease) window flow control, the packet transmission rate from a source host fluctuates at a time scale of a round-trip time. Although this is not a problem when using TCP with non-real-time applications such as data transfer applications, it becomes a serious problem for real-time applications such as video streaming applications.

Therefore, transport-layer communication protocols for real-time applications such as TFRC[1] (TCP-Friendly Rate Control), RAP[2] (Rate Adaptation Protocol), and TEAR[3] (TCP Emulate At Receivers) have been proposed in the literature. These transport-layer communication protocols commonly have the following features: (1) a congestion control mechanism for dynamically adjusting the packet transmission rate from a source host according to the congestion status of a network, (2) realizing fairness with competing TCP connections (TCP-friendliness) using the TCP throughput formula,[4] and (3) changing the packet transmission rate from a source host smoothly at a larger time scale than a round-trip time. In this paper, among these transport-layer communication protocols for real-time applications, we focus on TFRC, which has been standardized as RFC3448.[1]

TFRC or TCP-friendly rate control mechanism have been studied variously. For example, the fairness between TFRCP and TCP in steady state has been evaluated from simulation experiments and traffic measurements in the Internet.[5] These studies have shown the validity of the rate control mechanism based on the TCP throughput formula. Moreover, the fairness between TFRC and TCP in steady state has been evaluated from simulation experiments,[6] and has been shown that TFRC and TCP can share a bandwidth fairly. On the other hand, the transient state behavior of a TCP-friendly rate control mechanism has been evaluated from simulation experiments where TCP-friendly connections and TCP connections coexist.[7, 8] These studies have shown the smoothness of the throughput variations and the responding speed to the changes of the network congestion status have been shown quantitatively.

Most of these researches are based only on simulation experiments. Therefore, the steady state and transient state behavior of TFRC have not been analyzed when TFRC and TCP connections coexist. Therefore, in this paper, we analyze the steady state and transient state behavior of TFRC and TCP in a network where TFRC and TCP connections share the single bottleneck link. First, we model TFRC connection, TCP connection, and RED router as discrete time systems, respectively. We then derive TCP throughput, TFRC throughput, average queue length of the RED router, and packet loss probability at the RED router in steady state by utilizing these discrete time systems. Furthermore, by linearizing discrete time systems around their equilibrium points, we analyze the transient state behavior of TCP and TFRC in a networks where the TFRC connections and TCP connections coexist.

The rest of this paper is organized as follows. In Section 2, we model TCP connection, TFRC connection, and RED router as independent discrete-time systems. In Section 3, we derive TCP throughput, TFRC throughput, average queue length of the RED router, and packet loss probability at the RED router in steady state. Furthermore, in Section 4, by linearizing discrete time systems around their equilibrium points, we analyze the transient state behavior of TCP and TFRC. In Section 5, we present several numerical examples, and validate

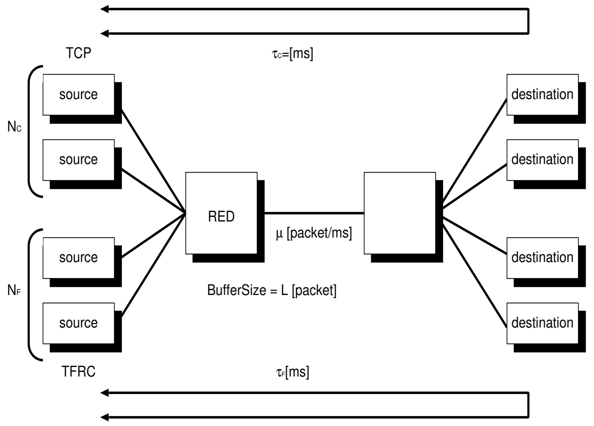


Figure 1: Analytic model

our approximate analysis by comparing analytic results with simulation ones. Finally, in Section 6, we conclude the current paper and discuss future works.

## 2. ANALYTIC MODEL

Figure 1 shows our analytic model used in this paper where $N_F$ TFRC connections and $N_C$ TCP connections share the single bottleneck link. For simplicity, we assume that all TFRC connections and TCP connections operate synchronously, respectively. We assume that two-way propagation delays of TFRC and TCP connections are equal. Those are denoted by $\tau_F$ and $\tau_C$, respectively. In this paper, we assume that access links between source host and router are sufficiently faster than the link between routers. Therefore, in this paper, we assume that buffering delays at routers connected to non-bottleneck links are negligible.

The bottleneck link capacity is denoted by $\mu$. We assume that all routers in the network are RED routers, and four control parameters of RED routers are denoted by $max_p$ (maximum packet drop probability), $max_{th}$ (maximum threshold), $min_{th}$ (minimum threshold), and $w_q$ (weight of the low pass filter).[9] The buffer size of a RED router is denoted by $L$. In this paper, TFRC connection, TCP connection, and RED router are modeled as discrete-time systems with a time slot of $\Delta$. Table 1 shows the definition of symbols used throughout in this paper.

First, we model change of the TCP window size. In this paper, we assume: (1) our focusing on steady state, all TCP connections operate in their congestion avoidance phase, (2) the maximum window size of TCP is sufficiently larger than the bandwidth–delay product of a network, and (3) RED routers operate appropriately; i.e. the average queue length of RED router $\overline{q}$ is kept between $min_{th}$ and $max_{th}$. By letting the packet loss probability in the network and the TCP window size be $p$ and $w$, change of TCP window size is given by[10]

$$w \leftarrow w + (1-p)\frac{1}{w} - p(1-p_{TO}(w,p))\frac{1}{2}\frac{4\,w}{3} - p\,p_{TO}(w,p)(\frac{4\,w}{3} - 1)$$

where, $p_{TO}(w,p)$ is the probability that a source host detects a packet loss by the timeout mechanism. $p_{TO}(w,p)$ is given by[4]

$$p_{TO}(w,p) \;\; = \;\; \frac{(1-(1-p)^3)\,(1+(1-p)^3\,(1-(1-p)^{w-3}))}{(1-(1-p)^w)}$$

Let $p(k)$ be the packet drop probability of the RED router at slot $k$, $w(k)$ be the TCP window size, and $R_F(k)$ and $R_C(k)$ be the round-trip times of TFRC and TCP connections, respectively. At slot $k$, the average arrival rate of ACK packets to a source host is approximately given by $w(k-\frac{R_C(k)}{\Delta})/R_C(k)$, and the packet loss probability is by $p(k-\frac{R_C(k)}{\Delta})$. Hence, TCP window size $w(k+1)$ at slot $k+1$ is approximately given by

$$\begin{aligned} w(k+1) \quad \simeq \quad & w(k) + \frac{w(k-\frac{R_C(k)}{\Delta})}{R_C(k)}\,\Delta \left\{ (1-p(k-\frac{R_C(k)}{\Delta}))\frac{1}{w(k)} \right. \\ & -p(k-\frac{R_C(k)}{\Delta})(1-p_{TO}(w(k-\frac{R_C(k)}{\Delta}),p(k-\frac{R_C(k)}{\Delta})))\frac{2\,w(k)}{3} \\ & \left. -p(k-\frac{R_C(k)}{\Delta})\,p_{TO}(w(k-\frac{R_C(k)}{\Delta}),p(k-\frac{R_C(k)}{\Delta}))(\frac{4\,w(k)}{3}-1) \right\} \end{aligned} \tag{1}$$

Next, we model change of the transmission rate from a TFRC connection. A destination host of TFRC measures a packet loss event rate, and sends it to a source host as feedback information. Let $p_e(k)$ be the packet loss event rate, and $T(k)$ be the transmission rate from TFRC connection at slot $k$. If the source host of TFRC receives feedback information at slot $k$, it updates its transmission rate $T(k+1)$ as

$$T(k+1) \quad = \quad \min\left(X(k,s), 2\,T(k-\frac{R_F(k)}{\Delta})\right) \tag{2}$$

where $X(k,s)$ is defined as

$$X(k,s) \quad = \quad \frac{s}{R_F(k)\sqrt{\frac{2p_e(k)}{3}} + t_{RTO}\left(3\sqrt{\frac{3p_e(k)}{8}}p_e(k)(1+32p_e(k)^2)\right)}$$

**Table 1.** Definition of symbols

| | |
|---|---|
| $N_F$ | the number of TFRC connections |
| $\tau_F$ | two-way propagation delay of TFRC connection |
| $R_F(k)$ | round-trip time of TFRC connection |
| $p_e(k)$ | packet loss event rate of TFRC connection |
| $T(k)$ | transmission rate of TFRC connection |
| $t_{RTO}$ | time out value of TFRC connection |
| $N_C$ | number of TCP connections |
| $\tau_C$ | two-way propagation delay of TCP connection |
| $R_C(k)$ | round-trip time of TCP connection |
| $w(k)$ | window size of TCP connection |
| $\mu$ | bottleneck link capacity |
| $L$ | buffer size of RED router |
| $\tau_Q(k)$ | queuing delay of RED router |
| $max_p$ | maximum packet drop probability |
| $min_{th}$ | minimum threshold for RED router |
| $max_{th}$ | maximum threshold for RED router |
| $w_q$ | weight of the low pass filter |
| $q(k)$ | current queue length of RED |
| $\overline{q}(k)$ | average queue length of RED |
| $p(k)$ | packet drop probability of RED router |
| $\Delta$ | time slot length |

where $s$ is the packet length of TFRC and $t_{RTO}$ is a TCP retransmission timer value which is, for simplicity, assumed to be $4R_F(k)$.[1]

The packet loss event rate $p_e(k)$ at slot $k$ and the packet loss probability $p(k)$ at slot $k$ satisfy the following equation.

$$\frac{1}{p(k)} = 1 \times \sum_{i=1}^{N} \left( (1-p_e(k))^{i-1} \, p_e(k) \right) + \sum_{i=N+1}^{\infty} \left( i \, (1-p_e(k))^{i-1} \, p_e(k) \right) \tag{3}$$

where $N$ is the number of packets that a TFRC connection transmits in a round-trip time, and it is given by

$$N = R_F \, T(k)$$

Furthermore, we model change of the current and average queue lengths of the RED router. Let $q(k)$ and $\overline{q}(k)$ be the current queue length and the average queue length of the RED router at slot $k$. At slot $k$, packet arrival rates at the RED router from each TFRC and TCP connection are given by $T(k)$ and $w(k)/R_C(k)$. So, letting the buffer size of the RED router be $L$, the current queue length at slot $k+1$ is given by

$$q(k+1) = \min \left[ \max \left\{ q(k) + \left( N_C \, \frac{w(k)}{R_C(k)} + N_F \, T(k) - \mu \right) \Delta, 0 \right\}, L \right] \tag{4}$$

Since a RED router updates an average queue length $\overline{q}$ as

$$\overline{q} \leftarrow (1 - w_q) \, \overline{q} + w_q \, q$$

The average queue length at slot $k+1$ is approximately given by

$$\overline{q}(k+1) \simeq \overline{q}(k) + \left( N_F \, T(k) + N_C \, \frac{w(k)}{R_C(k)} \right) \Delta \, w_q (q(k) - \overline{q}(k)) \tag{5}$$

The buffering delay $\tau_Q(k)$ of the RED router at slot $k$ is given by

$$\tau_Q(k) = \frac{q(k)}{\mu}$$

and round-trip times of TFRC and TCP connections at slot $k$ are given by

$$\begin{aligned} R_F(k) &= \tau_Q(k) + \tau_F \\ R_C(k) &= \tau_Q(k) + \tau_C \end{aligned}$$

Since a RED router randomly drops arriving packets with a probability determined by its average queue length $\overline{q}(k)$, the packet drop probability at the RED router is given by[9]

$$p(k) = \left( \frac{max_{th} - min_{th}}{2 \, max_p \, (\overline{q}(k) - min_{th})} + \frac{1}{2} \right)^{-1} \tag{6}$$

## 3. STEADY STATE ANALYSIS

We analyze the steady state behavior of TCP and TFRC connections utilizing the analytic model derived in Section 2. Specifically, by utilizing Eqs. (1)–(6), we derive TFRC throughput, TCP throughput, average queue length, and packet drop probability of the RED router in steady state.

Since the congestion control mechanism of TCP is an AIMD (Additive Increase Multiplicative Decrease) based feedback control, when the feedback delay is not zero, the window size of TCP oscillates and never converges to a constant value. Note that the TCP window size $w(k)$ in this paper represents not an instantaneous value of the oscillating TCP window size but the expected value of TCP window size

TCP window size, TFRC transmission rate, average queue length, and packet drop probability of the RED router in steady state are denoted by $w^*$, $T^*$, $\overline{q}^*$, and $p^*$, respectively. We can numerically obtain $w^*$, $T^*$, $\overline{q}^*$, and $p^*$ by equating $w(k) = w^*$, $T(k) = T^*$, $\overline{q}(k) = \overline{q}^*$, and $p(k) = p^*$ in both sides of Eqs. (1)–(6), and solving them for $w^*$, $T^*$, $\overline{q}^*$, and $p^*$. TFRC and TCP goodputs in steady state are given by $T^*(1 - p^*)$ and $w^*(1 - p^*)/R_C$, respectively.

## 4. TRANSIENT STATE ANALYSIS

We then analyze the transient state behavior of TCP and TFRC connections by linearizing discrete time systems around their equilibrium points. The network state including all TFRC and TCP connections, and the RED router is uniquely determined by *state variables*: $T(k), \cdots, T(k - \frac{R_F(k)}{\Delta})$, $w(k), \cdots, w(k - \frac{R_C(k)}{\Delta})$, $q(k), \cdots, (k - \max(\frac{R_C(k)}{\Delta}, \frac{R_F(k)}{\Delta}))$, and $\overline{q}(k), \cdots, \overline{q}(k - \max(\frac{R_C(k)}{\Delta}, \frac{R_F(k)}{\Delta}))$. We introduce a state vector $\mathbf{x}(k)$ which is defined as differences between each state variable at slot $k$ and its equilibrium value.

$$\mathbf{x}(k) \equiv \left[ \begin{array}{rcl} w(k) & - & w^* \\ & \vdots & \\ w(k - \frac{R_C(k)}{\Delta}) & - & w^* \\ T(k) & - & T^* \\ & \vdots & \\ T(k - \frac{R_F(k)}{\Delta}) & - & T^* \\ q(k) & - & q^* \\ & \vdots & \\ q(k \max(\frac{R_C(k)}{\Delta}, \frac{R_F(k)}{\Delta})) & - & q^* \\ \overline{q}(k) & - & \overline{q}^* \\ & \vdots & \\ \overline{q}(k - \max(\frac{R_C(k)}{\Delta}, \frac{R_F(k)}{\Delta})) & - & \overline{q}^* \end{array} \right]$$

We assume that the destination host of TFRC notifies its source host of feedback information every $M$ slots. We focus on the state transition from slot $k$ to slot $k+M$. Although all discrete time systems (Eqs. (1) –(6)) are nonlinear, they can be written in the following matrix form by linearizing them around their equilibrium points.

$$\mathbf{x}(k+M) = \mathbf{A}\,\mathbf{B}^{M-1}\mathbf{x}(k) \tag{7}$$

Matrix $\mathbf{A}$ represents the state transition from $\mathbf{x}(k+M-1)$ to $\mathbf{x}(k+M)$, when the source host of TFRC receives feedback information from its destination host (Eq. (2)). Moreover, matrix $\mathbf{B}$ represents the state transition from $\mathbf{x}(k)$ to $\mathbf{x}(k+1)$ when the source host of TFRC does not receive feedback information at slot $k$ (i.e., $T(k+1) = T(k)$). The eigen values of the state transition matrix $\mathbf{A}\,\mathbf{B}^{M-1}$, representing transition of state variables from slot $k$ to slot $k+M$, determines the transient state behavior around equilibrium points. Let $\lambda_i (1 \leq i \leq \frac{R_C(k)}{\Delta} + \frac{R_F(k)}{\Delta} + 2\max(\frac{R_C(k)}{\Delta}, \frac{R_F(k)}{\Delta}))$ be the eigen values of the state transition matrix $\mathbf{A}\,\mathbf{B}^{M-1}$. The maximum absolute value of eigen values, *maximum modulus*, determines the stability and the transient behavior of the feedback system around their equilibrium points.[11] It is known that the smaller the maximum modulus is, the better the transient behavior becomes. It is also known that the system is stable if the maximum modulus is less than 1.0.

## 5. NUMERICAL EXAMPLES

We present several numerical examples on our steady state analysis as well as several simulation results. Using ns-2 simulator,[12] we run simulation experiments for the same network model with Fig. 1. Each simulation experiment is run for 300 seconds, and a result of the last 200 seconds is used for calculating TFRC and TCP goodputs, average queue length, and packet loss event rate of TFRC.

In obtaining analytic and simulation results, we use the following parameters. TFRC and TCP packet size are fixed at 1000 [byte], and the number of TFRC and TCP connections are $N_F = 10$ and $N_C = 10$, respectively. Two-way propagation delays of TFRC and TCP are set to the same value: $\tau = \tau_C = \tau_F = 50, 100, or 200$ [ms]. According to the two-way propagation delay $\tau$ and the bottleneck link capacity $\mu$, we configure control parameters ($min_{th}$ and $max_{th}$) and the buffer size of RED routers $L$; i.e., $min_{th} = 0.25\mu\tau$, $max_{th} = 1.25\mu\tau$, $L = 2.5\mu\tau$ [packet]. Other control parameters of the RED router are fixed at $max_p = 0.1$ and $w_q = 0.002$.

Figure 2 shows TFRC goodput, TCP goodput, average queue length, packet loss event rate of TFRC for different bottleneck link capacities. In the figure, both analytic and simulation results are shown for different two-way propagation delays of TFRC and TCP with $\tau = 50$ and $\tau = 100$ This figure indicates that both analytic and simulation results show a good agreement, showing the validiy of our approximate analysis.

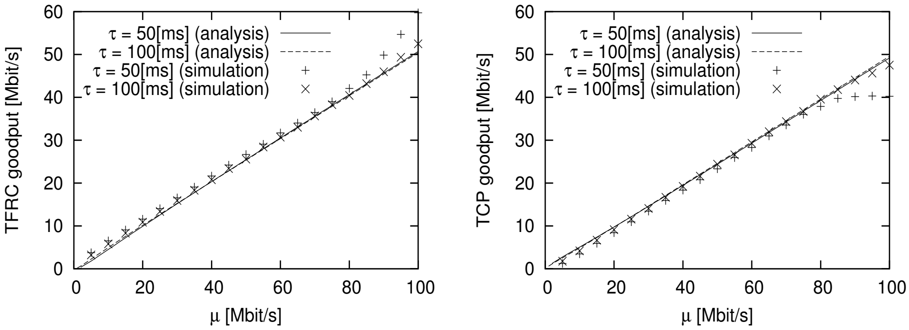
As can be found by comparing Fig. 2(a) with Fig. 2(b), the TFRC goodput is slightly higher than that of TCP although TFRC is designed for realizing fairness with TCP connections (TCP-friendliness). This phenomenon is in agreement with other simulation studies.[6]

## 6. CONCLUSION

In this paper, we have analyzed the steady state behavior and the transient state behavior of TFRC and TCP in a network where TFRC and TCP connections commonly share the single bottleneck link. First, we have modeled TFRC connection, TCP connection, and RED router as discrete-time systems, respectively. We have then derived TFRC goodput, TCP goodput, average queue length of the RED router, and packet loss probability at the RED router in steady state by utilizing these discrete-time systems. By linearizing discrete-time systems around their equilibrium points, we have also analyzed the transient state behavior of TCP and TFRC.

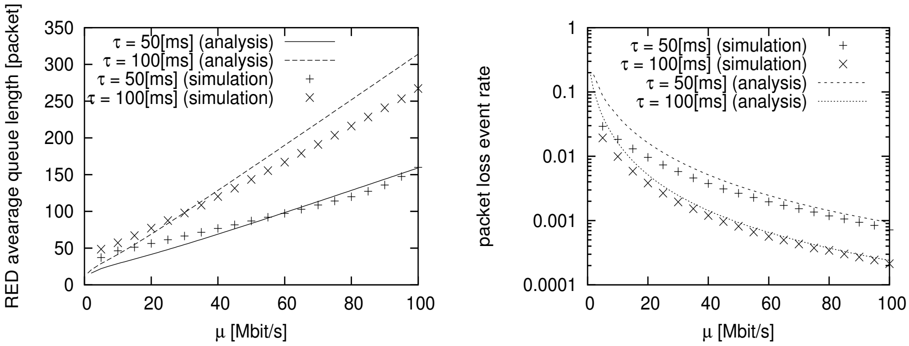
## REFERENCES

1. M. Handly, S. Floyd, J. Padhye, and J. Widmer, “TCP friendly rate control (TFRC): protocol specification,” *Request for Comments (RFC) 3448* , 2003.
2. R. Rejaie, M. Handley, and D. Estrin, “RAP: An end-to-end rate-based congestion control mechanism for realtime streams in the internet,” in *Proceedings of INFOCOM 1999*, pp. 1337–1345, Mar. 1999.
3. I. Rhee, V. Ozdemir, and Y. Yi, “TEAR: TCP emulation at receivers – flow control for multimedia streaming,” *NCSU Technical Report* , Apr. 2000.
4. 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.
5. J. Padhye, J. Kurose, D. Towsley, and R. Koodli, “A model based TCP-friendly rate control protocol,” *UMass-CMPSCI Technical Report TR 98-04* , 1998.



(a) TFRC goodput

(b) TCP goodput



(c) RED average queue length

(d) TFRC packet loss event rate

Figure 2: **Steady state behavior for different bottleneck link capacities**, $NF = 10$, $NC = 10$, $maxp = 0.1$, $w_q = 0.002$, $min_{th} = 0.25 \times \mu \times \tau$, $max_{th} = 1.25 \times \mu \times \tau$, $L = 2.5 \times \mu \times \tau$

6. S. Floyd, M. Handley, J. Padhye, and J. Widmer, "Equation-based congestion control for unicast applications," in *Proceedings of ACM SIGCOMM*, pp. 43–56, (Stockholm, Sweden), Aug. 2000.
7. Y. R. Yang, M. S. Kim, and S. S. Lam, "Transient behaviors of TCP-friendly congestion control protocols," in *Proceedings of INFOCOM*, pp. 1716–1725, Apr. 2001.
8. D. Bansal, H. Balakrishnan, S. Floyd, and S. Shenker, "Dynamic behavior of slowly-responsive congestion control algorithms," in *Proceedings of ACM SIGCOMM*, pp. 263–274, Aug. 2001.
9. S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Transactions on Networking* **1**, pp. 397–413, Aug. 1993.
10. S. H. Low, F. Paganini, and J. C. Doyle, "Internet congestion control," *IEEE Control Systems Magazine* **22**, pp. 28–43, Feb. 2002.
11. 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.
12. "The network simulator – ns2." available at `http://www.isi.edu/nsnam/ns/`.