Local and global stability of TCP-newReno/RED with many flows

Xinbing Wang\textsuperscript{a}, Do Young Eun\textsuperscript{b,∗}

\textsuperscript{a} Department of Electronic Engineering, Shanghai Jiaotong University, Shanghai, China
\textsuperscript{b} Department of Electrical and Computer Engineering, North Carolina State University, Raleigh, NC 27695, USA

Received 17 February 2006; received in revised form 14 November 2006; accepted 18 November 2006

Available online 15 December 2006

Abstract

Stability is one of the important issues for a TCP/AQM (Active Queue Management) system. In this paper, we study the local and global stability of TCP-newReno/RED under many flows regime. The existing results of the local stability are mostly for TCP-Reno, not for newReno. These results are obtained based on a small scale model with a few number of flows and thus cannot be blindly applied to a large system with many flows. Moreover, traditional approaches for the global stability based on Lyapunov functions is not suitable for a system with a large amount of flows due to its complexity. Motivated by this, we present a normalized discrete-time model to capture the essential dynamics of TCP-newReno/RED with many flows and obtain its local stability criterion. The normalized model allows us to proceed numerical iterations to analyze the global stability in an efficient manner. Our results show that by properly choosing some ‘free’ parameters, we can always ensure that a locally stable TCP-newReno/RED system is in fact globally stable. Our results become more accurate as the number of flows increases. Finally, we extend our normalized model to the case of heterogeneous RTTs.

© 2006 Elsevier B.V. All rights reserved.

Keywords: TCP; NewReno; Local stability; Global stability

1. Introduction

Stability has been one of the important issues in designing and analyzing TCP/AQM (Active Queue Management) system. According to [2], an unstable TCP/AQM system may increase delay and jitter, which can be detrimental to delay-jitter sensitive applications, and cause link under-utilization. The existing research on stability of TCP/AQM can be broadly classified into two groups: local stability and global stability.

The local stability is obtained by first linearizing the system around its equilibrium point. Then, by applying the Nyquist theorem to that linearized system, one can have a criterion for the local stability of the system [3,7,5,6,15]. However, due to the linearization involved, the local stability criterion is applicable only when the system operates near its equilibrium point. If the system operates on a region far away from its equilibrium point, then the nonlinear components may kick in and make such a linear approximation inaccurate. We define a \textit{stability region} of a system by a set of initial points, from which the system eventually converges to its equilibrium point. Then, for a locally stable system, this stability region is usually small and limited around the equilibrium point.

On the other hand, the global stability requires that starting from any initial points (not limited to the points near the equilibrium point), the system eventually converges to its equilibrium point. Obviously, the global stability implies the local stability. In general, the global stability is obtained by properly constructing a Lyapunov function \cite{16–22,10,23,24}. In [22], the authors present a criterion for the global stability of a congestion controller with a single link and a single flow. Further, they show that the global stability criterion for a system with many identical flows sharing a bottleneck link converges to that of a single flow \cite{11}. In other words, the condition for global stability with
a single flow can be applied to the case of multiple flows as well.

Although the existing results regarding stability reveal valuable insight into the system design, they have their own limitations. For local stability, it is mostly applied to a system with TCP-Reno [2,5,7] instead of TCP-newReno, which is an improved version of TCP-Reno and widely deployed in practice. The major difference between TCP-newReno and TCP-Reno is that for TCP-Reno, multiple packet losses or packet marks within one round trip time (RTT) results in multiple reductions of window sizes, while for TCP-newReno, only one reduction of window sizes is incurred (see [25] for more details). To the best of our knowledge, there is no result regarding the stability criterion of TCP-newReno under many flows.

Meanwhile, the existing local stability criterion is designed for a relatively small scale system (i.e., the number of flows in the system is on the order of tens), while we consider a large system where the number of flows is hundreds or thousands level. According to [26], the core router is shared by more than 10,000 flows, and we show that for such a large system, the local stability criterion is different from that of a relatively small system. For example, in a small system, the long RTT of a single flow will affect the stability criterion [2]. However, for a large system, we will show in this paper that the RTT for a single flow or a small number of flows will not make any impact on the stability of the system as a whole.

As to the global stability, the AQMs studied in the literature are mostly limited to rate-based marking schemes [11,22], and not applicable to any other popular queue-based marking schemes such as RED [27]. Still, to the best of our knowledge, there has been no result about the global stability of a TCP-newReno/RED system with many flows. This is mainly due to the following two reasons. First, as the network evolves, the core router is shared by increasingly large number of flows. Since each flow has its own window size dynamics, it becomes more difficult to find an appropriate Lyapunov function for such a system with many flows. Second, the packet marking function for RED is a nonlinear (usually piecewise-continuous) function, and this makes any theoretical analysis formidable.

In this paper, we consider a TCP-newReno/RED system with many flows and focus on its local and global stability. First, we propose a simple, yet accurate model to capture the key system dynamics of TCP-newReno/RED. Our model is motivated by the results in [11] and [28], in which the authors show that the system dynamics could be described by a normalized model with a single flow. Based on our model, we derive the local stability criterion for a TCP-newReno/RED system with many flows. After we choose the system parameters such that the system is locally stable, we point out that there are some ‘free’ parameters we can freely choose without affecting the local stability of the system. Then, using numerical analysis, we show that, by carefully choosing the free parameter of the RED, it is always possible to make the system globally stable. Our numerical approach shows that the stability region of a TCP/RED system is quite irregular and very sensitive to the RED configuration. This is in accordance with the results in [29], saying that the performance of TCP/RED is very sensitive to the parameter configuration. In other words, a relatively small variation of RED configuration may cause a dramatic change of the stability region. We note that our numerical approach is invariant with the number of flows, which greatly reduces the simulation complexity for a large number of flows. For example, it is infeasible for simulators such as ns-2 [30] to simulate a network with a large number of flows (say, 10,000 flows) mainly due to the state explosion. However, our approach can easily handle such case because our model directly captures the average behavior of the system via a single flow dynamics. Further, as we will see, our approach becomes more accurate as the number of flows in the system increases. Finally, we extend our model to the case of many flows with heterogeneous RTTs. We derive the corresponding stability criteria and illustrate their implications. Our results indicate that for a large system, the stability criterion is different from that of a small system and we cannot blindly apply the stability criterion for a small system to a large scale.

The remainder of this paper is organized as follows. In Section 2, we describe the TCP/RED system and propose a simple, yet accurate model for the system with many flows. In Section 3, we derive a criterion for the local stability of the system. In Section 4, by properly configuring system parameters, we show how to obtain the global stability of the system from the local stability criterion. We extend our results to heterogeneous RTTs in Section 5. Section 6 is devoted to some simulation results and Section 7 concludes this paper.

2. System model

In this section, we describe the problem and introduce a normalized model that essentially captures the average behavior of the queue-length and window sizes. Then, we justify our model by comparing it with the existing one in [28,31,32].

2.1. Problem description

We first consider a system with identical TCP flows with the same round-trip-time (RTT). Later in this paper, we will also consider the case of heterogeneous RTTs in Section 5. Suppose that there are $N$ long-lived TCP flows sharing one common bottleneck link with capacity $Nc$ packets/s, where $c$ is a given constant. Let the two-way propagation delay be $T$ s and $Q^N(t)$ be the queue-length at time $t$. We set the length of each time slot to be one RTT, which is equal to the two-way propagation delay plus the queuing delay, i.e., $RTT = T + Q^N(t)/Nc$. In this way, the target window size per flow becomes $C = c \times RTT = cT + Q^N(t)/N$. We plot the system model in Fig. 1(a).
RED is implemented in the bottleneck router. According to the standard of RED specified in [33], the making/dropping probability is determined only by the queue-length. If the queue-length is smaller than $N \times q_1$, then none of the incoming packets is marked. (See Fig. 1(b).) If the queue-length falls between $N \times q_1$ and $N \times q_2$, the marking probability increases slowly with slope $\rho_1$. Once the queue-length falls between $N \times q_2$ and $2N \times q_2$, the marking probability increases sharply with slope $\rho_2$ ($\rho_2 > \rho_1$). Finally, if the queue-length is larger than $2N \times q_2$, all the incoming packets are marked.

For the TCP/RED system with many flows, we want to find the RED parameters as in Fig. 1(b) such that the system is globally stable. We see that some parameters in our model are (i) the system behavior gets greatly simplified; (ii) by appealing to the law of large numbers, (4) becomes

$$Q(t+1) = [Q(t) + W(t) + C]_+, \quad (1)$$

where $[x]^+ = \max\{x, 0\}$. Note that (1) is equivalent to

$$Q_N(t+1) = \left[\begin{array}{c} Q_N(t) + \sum_{i=1}^{N} W_i(t) \end{array}\right]_+ - NC,$$

which is just the Lindley recursion for the actual queue-length $Q_N(t)$. **Base marking function:**

The base marking probability $f(x)$ is given by

$$f(x) = \begin{cases} 0 & \text{if } x \leq q_1, \\ \rho_1(x - q_1) & \text{if } q_1 < x \leq q_2, \\ P_{\text{max}} + \rho_2(x - q_2) & \text{if } q_2 < x \leq 2q_2, \\ 1 & \text{if } 2q_2 < x, \end{cases} \quad (2)$$

where $\rho_1 = \frac{P_{\text{max}}}{q_2 - q_1}$ and $\rho_2 = \frac{P_{\text{max}}}{2q_2 - q_1}$. Then, the probability that at least one among those $W(t)$ packets is marked becomes

$$p(t) = 1 - (1 - f(Q(t)))^{W(t)}. \quad (3)$$

**Average window dynamics:** According to TCP Additive-Increase-Multiple-Decrease (AIMD) algorithm, the window size increases by 1 with probability $1 - p(t)$ and halves with probability $p(t)$. Thus, on average, the average window dynamics $W(t)$ can be described as in [2]

$$W(t+1) = (W(t) + 1)(1 - p(t)) + \frac{W(t)}{2}p(t). \quad (4)$$

**Remark.** Our approach here is to use the normalized queue-length and the average window size to capture the system dynamics. The merits of using the normalized model are (i) the system behavior gets greatly simplified; (ii) by appealing to the law of large numbers, (4) becomes more accurate for larger number of flows. We next provide numerical results to verify that our model is indeed accurate as long as the number of flows is large enough.
2.3. Model comparison

We compare our normalized model in (1)–(4) with the original model. Table 1 lists parameters used in the numerical simulation for model comparison.

Fig. 2 shows the comparison of the original model and the normalized model, when the system is unstable. (In other words, the normalized queue-length and the average window size suffer wild fluctuations.) Fig. 2(a) displays the average window size dynamics of the original model and our normalized model. We observe that the normalized model yields almost the same window dynamics as the original model. Similarly, Fig. 2(b) shows the comparison of queue dynamics of two models. The normalized model can effectively capture the wild fluctuation of the unstable queue.

In Fig. 3, we compare two models when the system is stable. (In other words, the fluctuations in the normalized queue-length and in the average window size are small.) Here, we replace the parameters \( q_1 \) and \( q_2 \) in Table 1 with \( q_1 = 0.001 \) and \( q_2 = 1040 \) such that the system is stable. Figs. 3(a) and (b) show the average window size and the normalized queue-length, respectively, drawn on the same scale as in Fig. 2. We see that both the normalized queue-length and the average window size stay around some fixed values (equilibrium points) without much fluctuation.

Fig. 4 displays the variance of the average window size and the normalized queue-length in the original model.
when the number of flows $N$ increases. Clearly, the variance decreases sharply with the number of flows. In other words, when the number of flows is large enough, our normalized model is almost indistinguishable from the original model.

3. Local stability criterion

Based on the normalized model, we now study the local and the global stability of the system. In this section, we first configure the system to ensure the local stability. Then, we show in Section 4 that we can always enlarge the stability region to entire space (i.e., global stability) by configuring other “free” system parameters.

3.1. Linearizing system around equilibrium point

After the normalization, the system is still nonlinear and nondifferentiable. In order to analyze it, we need to make some simplification, while still keep the system key characteristics. Such methods are widely used in the literature [22,27].

Our approach is based on the following assumptions:

- (A1) $q_1 \leq Q(t) \leq q_2$.
- (A2) $C \gg 2$.

Since we are interested in the local stability of the system, it is reasonable to assume that the equilibrium queue-length is located between $q_1$ and $q_2$ [2,3,7]. Assumption (A2) states that the target window size for each long-lived TCP flow is far greater than 2, which is generally the case in practice. In fact, (A2) is just for simplicity and we can go without (A2) at the cost of more complicated computation.

To simplify the notation, we define

$$\hat{Q}(t) = Q(t) - q_1.$$  \hfill (5)

**Lemma 1.** If the system in (1)–(4) converges, then it must converge to the following unique equilibrium point:

$$\lim_{t \to \infty} W(t) = C,$$

$$\lim_{t \to \infty} \hat{Q}(t) = \frac{1}{\rho_1 C (1 + \frac{x}{2})}. \hfill (7)$$

**Proof.** As $t \to \infty$, $W(t + 1) = W(t) = W(\infty)$, and $Q(t + 1) = Q(t) = Q(\infty)$. By solving (1)–(4), we have $W(\infty) = C$. Then, using (4), we obtain the equilibrium marking probability $p(\infty) = 1/(1 + \frac{x}{2})$. From (3) and (5), we have

$$\hat{Q}(\infty) = \frac{1}{\rho_1} \left[ 1 - \left( 1 - \frac{1}{1 + \frac{x}{2}} \right) \right] \approx \left[ \rho_1 C \left( 1 + \frac{C}{2} \right) \right]^{-1},$$

where the approximation is due to $1 - (1 - x)^z \approx xz$ for small $x$ and $z$ and $C \gg 2$ from (A2). \hfill $\square$

Note that, from (2), the equilibrium marking probability $f^*$ becomes

$$f^* = 1 - \left( 1 - \frac{1}{1 + \frac{x}{2}} \right) \approx \left[ C \left( 1 + \frac{C}{2} \right) \right]^{-1}. \hfill (8)$$

Since the queue is always positive under (A1), we can simplify the queue dynamics in (1) as

$$\hat{Q}(t + 1) = \hat{Q}(t) + W(t + 1) - C.$$ 

Near the equilibrium point, we have $p(t) \approx f(\hat{Q}(t)) W(t)$ again from $1 - (1 - x)^z \approx xz$. We can thus simplify the marking probability in (3) as $p(t) = \rho_1 C \hat{Q}(t)$. Summarizing the above result, we obtain the following simplified model around the equilibrium point:

Fig. 4. Variance of the average window size and the normalized queue-length in the original model as the number of flows varies. (a) Variance of window size. (b) Variance of queue size.
\[ W(t+1) = W(t) \left( 1 - \frac{\rho_1 C Q(t)}{2} \right) + (1 - \rho_1 C) \tilde{Q}(t) \]
\[ = F_1(W(t), \tilde{Q}(t)), \tag{9} \]
\[ \tilde{Q}(t+1) = \tilde{Q}(t) \left( 1 - \rho_1 C - \frac{\rho_1 C W(t)}{2} \right) + W(t) + 1 - C \]
\[ = F_2(W(t), \tilde{Q}(t)), \tag{10} \]
where \( F_1, F_2 : \mathbb{R}^6 \rightarrow \mathbb{R} \) are appropriate functions.

3.2. Local stability criterion

Based on the above simplified model, we obtain the following criterion for the local stability.

**Theorem 1.** The system described by (9) and (10) converges to the equilibrium point, if both \(|\lambda_1| < 1\) and \(|\lambda_2| < 1\), where

\[ \lambda_{1,2} = 1 - \frac{1}{2} \left[ \rho_1 \left( \frac{C^2}{2} + C \right) + \frac{1}{2 + C} \right] \pm \sqrt{\left[ \rho_1 \left( \frac{C^2}{2} + C \right) + \frac{1}{2 + C} \right]^2 - 4 \rho_1 \left( \frac{C^2}{2} + C \right)}. \]  

(11)

**Proof.** From (9) and (10), we know that it is a discrete nonlinear system. Since we already know the equilibrium point of the system, we linearize the system near the equilibrium point to get

\[ \begin{pmatrix} \Delta W(t+1) \\ \Delta \tilde{Q}(t+1) \end{pmatrix} = A \begin{pmatrix} \Delta W(t) \\ \Delta \tilde{Q}(t) \end{pmatrix}, \]

where

\[ \Delta W(t) = W(t) - C, \quad \Delta W(t+1) = W(t+1) - C, \quad \Delta \tilde{Q}(t) = \tilde{Q}(t) - \frac{1}{\rho_1 C (1 + \frac{C}{2})}, \quad \Delta \tilde{Q}(t+1) = \tilde{Q}(t+1) - \frac{1}{\rho_1 C (1 + \frac{C}{2})}, \]

and matrix \( A \) is the Jacobian matrix given by

\[ A = \begin{pmatrix} \frac{\partial \tilde{Q}}{\partial W} & \frac{\partial \tilde{Q}}{\partial Q} \\ \frac{\partial W}{\partial W} & \frac{\partial W}{\partial Q} \end{pmatrix} \bigg|_{W=W(\infty), \tilde{Q}=Q(\infty)}. \]

Substituting \( W(t) \) and \( \tilde{Q}(t) \) by their equilibrium values, respectively, gives

\[ A = \begin{pmatrix} 1 - \frac{1}{2 + C} & -\rho_1 C (\frac{C}{2} + 1) \\ 1 - \frac{1}{2 + C} & 1 - \rho_1 C (\frac{C}{2} + 1) \end{pmatrix}. \]

The eigenvalue of matrix \( A \) is obtained by solving \( \det[\lambda I - A] = 0 \), i.e.,

\[ \lambda^2 - \left( 2 - \rho_1 \left( \frac{C^2}{2} + C \right) - \frac{1}{2 + C} \right) \lambda + 1 - \frac{1}{2 + C} = 0. \]

The roots of the above equation are obtained in (11). Then, we know that if \(|\lambda_1|\) and \(|\lambda_2|\) are smaller than 1, then the system in (9) and (10) is stable. Specifically,

\[ \begin{pmatrix} \Delta W(t) \\ \Delta \tilde{Q}(t) \end{pmatrix} = \Xi \begin{pmatrix} (\lambda_1)' & 0 \\ 0 & (\lambda_2)' \end{pmatrix} \Xi^{-1} \begin{pmatrix} \Delta W(0) \\ \Delta \tilde{Q}(0) \end{pmatrix}, \]

where \( \Xi \) is the transformation matrix of \( A \). Since \(|\lambda_1| < 1\) and \(|\lambda_2| < 1\), we have

\[ \lim_{t \to \infty} (\lambda_1)' = 0 \quad \text{and} \quad \lim_{t \to \infty} (\lambda_2)' = 0, \]

and

\[ \lim_{t \to \infty} \Delta W(t) = 0 \quad \text{and} \quad \lim_{t \to \infty} \Delta \tilde{Q}(t) = 0. \]

Therefore, we have

\[ \lim_{t \to \infty} W(t) - C = 0, \quad \text{and} \quad \lim_{t \to \infty} \tilde{Q}(t) - \frac{1}{\rho_1 C (1 + \frac{C}{2})} = 0. \]

In other words, the average window size converges to \( C \) and the normalized queue-length converges to \( \frac{1}{\rho_1 C (1 + \frac{C}{2})} \). This is accordance with the results in Lemma 1 and thus the system is locally stable. \( \square \)

Note that both \( \lambda_1 \) and \( \lambda_2 \) are smaller than 1 if

\[ 0 < \rho_1 (C/2 + 1) + 1/(2 + C) < 1. \]  

(12)

This suggests that we could always find \( \rho_1 \) (possibly very small) such that the system is locally stable. However, one needs to be careful here because \( C \) and \( \rho_1 \) are related with each other, which is ignored in the literature. Recall that \( C \) is the target window size of each flow, i.e., \( C = cT + \tilde{Q}(t)/Nc = cT + Q(t) \), where \( cT \) is constant and \( Q(t) \) is the normalized queue-length, which is a function of \( \rho_1 \). In general, if we decrease \( \rho_1 \), the equilibrium queue-length will be increased to get ‘right’ amount of marking to maintain equilibrium. This in turn makes \( C \) larger, which may compromise the effect of reduction in \( \rho_1 \) (see (12)). Therefore, we need to know the relationship between \( \rho_1 \) and \( C \) to precisely determine the local stability.

If the system is stable, it will converge to the equilibrium point eventually. Then at the equilibrium point, \( C = cT + Q(\infty) \), and by (7), \( Q(\infty) = [\rho_1 C (1 + \frac{C}{2})]^{-1} + q_1 \). So, we have \( C - cT = [\rho_1 C (1 + \frac{C}{2})]^{-1} + q_1 \). That is,

\[ \rho_1 C \left( \frac{C}{2} + 1 \right) (C - cT - q_1) = 1. \]

(13)

There is no technical difficulty to solve (13) to obtain the roots in terms of \( \rho_1 \). We can obtain three roots for (13), but only one root is meaningful and the other two are either complex numbers or negative numbers.
4. From local stability to global stability

To the best of our knowledge, there has been no result on how to extend the local stability to the global stability for the system under our consideration (cf. Fig. 1(a)).

Theorem 1 shows that a suitable choice of $\rho_1$ always ensures the local stability of the system. Recall that we have defined the stability region as the set of points, starting from which the system converges to its equilibrium. In this section, we will show that, by properly choosing other “free” parameters that have nothing to do with the local stability, we can always configure the system such that it converges to its unique equilibrium point starting any initial points outside of the local stability region, i.e., the system is globally stable. We first consider a box region and show that all the points outside the box as initial states will eventually move into the inside of the box. Then, we enumerate the points inside the box region as initial states to check whether or not the system converges to its equilibrium.

4.1. Convergence from outside a box

**Lemma 2.** Consider a box with vertices at $(0,0)$, $(0,W_{\text{max}})$, $(2q_2, W_{\text{max}})$, and $(2q_2, 0)$. If the system (defined by (1)–(4)) starts from any points outside the box, it will eventually fall into the inside of the box.

**Proof.** Without loss of generality, we can assume that there exists $n < \infty$ such that $2^{n-1}C \leq W_{\text{max}} < 2^n C$. Let us consider a box region displayed in Fig. 5.

- Case I: Consider a point $(W(t), Q(t))$ in area (I), i.e., $W(t) \leq C$ and $Q(t) \geq 2q_2$. Since $f(Q(t)) = p(t) = 1$, the window size will halve at $t + 1$, i.e., $W(t+1) = W(t)/2$. If $W(t+1) < C$, we have $W(t + 1) = W(t)/2$. If $W(t+1) < C$, we are done. When $W(t + 1) \geq C$, note that it takes at most $n$ iteration for $(W(t), Q(t))$ to fall back into the area (I) as $W(t + n) = W(t)/2^n < C$.

Thus, all the points in the area (I) and (II) will eventually move into the inside of the box and this completes the proof. □

4.2. Choosing free parameters to ensure global stability

In this section, we will show that $q_2$ is the “free” parameter we mentioned earlier. With properly chosen $q_2$, starting from any point within the box region defined in Lemma 2, the system converges to the equilibrium point.

Specifically, note that in Theorem 1 the local stability depends only on $\rho_1$ and $C$. So, we are free to choose any $q_2$ without affecting the local stability of the system. Let $q_2 = Q(\infty) + \Delta q$, where $Q(\infty)$ is the equilibrium queue-length as in (5) and (7), and $\Delta q$ is the deviation from the equilibrium point. From $\Delta q = 0$, we increase $\Delta q$ until the system converges starting from any points inside the box region. Then, by Lemma 2, the system will be globally stable.

Now, we explain how to proceed numerical analysis based on our normalized model. For every given $\Delta q > 0$, we consider points inside the box with vertices at $(0,0)$, $(0,W_{\text{max}})$, $(2q_2, W_{\text{max}})$, $(2q_2, 0)$, and set those points as initial point of $(Q(0), W(0))$. Then, by iterating (1)–(4), we can check whether or not the system starting from those points converges to the equilibrium point. If $\Delta q$ is too small, not all the points inside the box will converge. So we gradually increase $\Delta q$ until all the points inside the box as initial points will converge to the equilibrium point. In addition, Lemma 2 asserts that the system starting from outside of the box will fall back into inside of the box. In this way, we can configure the system to be globally stable as long as all the points inside the box converges.

**Example 1.** We set $C = 12$ and $\rho_1 = 0.01$. By (11), we have $\lambda_{1,2} = 0.5443 \pm 0.7952$. Since $\|\lambda_{1,2}\| = 0.9636 < 1$, we obtain $Q(\infty) = 1.2764$, $W(\infty) = 12$ by (7). Without loss of generality, we assume $W_{\text{max}} = 20$. We plot 3 figures with different small $\Delta q$ in Figs. 6(a)–(c). The points falling in the dark region will converge to the equilibrium point. On the contrary, the points within the blank region will not converge to the equilibrium point, which means the system will oscillate for ever. We can obtain Fig. 6(a) with $\Delta q = 0.3$, Fig. 6(b) with $\Delta q = 0.4$, and Fig. 6(c) with...
\( \Delta q = 0.6 \). When \( \Delta q = 0.6 \), the whole box area is covered by convergence points.

**Example 2.** We set parameters of (1) and (4) as follows. \( C = 6.70, \rho_1 = 0.01 \). By simple calculation of (11), we have \( \lambda_{1,2} = 0.7968 \pm i0.5002 \), and we can verify that \( |\lambda_{1,2}| < 1 \). Then the system is locally stable with the unique equilibrium point of \( (Q(\infty) = 3.8237, W(\infty) = 6.70) \) by (7). Without loss of generality, we assume \( W_{\text{max}} = 20 \). We plot 2 figures with different small \( \Delta q \) in Fig. 7. The points falling in the red region will converge to the equilibrium point. On the contrary, the points within the blank region will not converge to the equilibrium point, which means the system will oscillate for ever. Figs. 7(a) and (b) display the stable points with \( \Delta q = 0.05 \) and \( \Delta q = 0.07 \), respectively. When \( \Delta q = 0.1 \), all the points within the box areas: (0,0), (0,20), (10,20), (10,0) will converge to the equilibrium point (3.8237, 6.70).

From Figs. 7(a) and (b), we have the following observations: (i) The profile of stable points are quite irregular. It is hard to obtain a close form expression for the profile. (ii) With a very small variation of \( \Delta q \) (say, from \( \Delta q = 0.05 \) to \( \Delta q = 0.07 \)), stable region changes dramatically. (iii) A relative small \( \Delta q \) (for example, \( \Delta q = 0.1 \)) is enough to obtain global stability.

4.3. Discussion

From the numerical results, we can see that a quite small \( \Delta q \) is sufficient to make the whole box region stale (say \( \Delta q = 0.15 \)). In other words, the whole box region can be quickly filled as \( \Delta q \) increases. An intuitive interpretation of our results is that if the queue falls within the region with slope of \( \rho_1 \) (i.e., between \( q_1 \) and \( q_2 \)), by local stability the queue will converge to equilibrium point eventually; if the queue falls outside the region with slope of \( \rho_1 \), (say, the region with the slope of \( \rho_2 \), i.e., between \( q_2 \) and \( 2q_2 \)), the queue will fall back to the region with slope of \( \rho_1 \). This is because most packets will be marked and most of TCP flows will reduce their window size by half. Then the number of packets arrival to the router will be smaller and the queue-length will decrease to the range of \( \rho_1 \).

Our results show that a relative small \( \Delta q \) is enough for the system to be globally stable. In reality, the propose RED suggests enough margin is necessary in order to accommodate other traffic patterns such as burst traffic arrival, webmice traffic, and UDP traffic. Hence, in reality, the sufficient margin (i.e., \( \Delta q \)) is available. In most cases, if a system is configured to be locally stable, it is also globally stable.

More importantly, our numerical analysis is independent of the number of flows, which is based on the iteration.
of (1)–(4). This is especially important, when we study a system with a huge number of flows. According to [26], the core router is shared by over 10,000 flows. The simulator (say ns-2 [30]) cannot simulate such a system because of the state explosion and hence it is infeasible to verify global stability by the simulator. However, our numerical approach can easily be proceed and finish within a very short time (around several minutes), which means our approach has good scalability. Meanwhile, with more flows our results become more accurate.

5. Extension to heterogeneous RTTs

In reality, each flow goes through its own access network and then arrive at the bottleneck router. Thus, the round trip time (RTT) will be different for each flow. In order to capture RTT heterogeneity, we denote \( g(x) \) as the probability density function (pdf) of RTTs and discretize RTTs into \( m \) levels.\(^1\) For example, if we set the granularity of RTTs to be 20 ms, then \( \psi_1 \) is the percentage of RTT in \([0, 20)\) ms, \( \psi_2 \) is the percentage of RTT in \([20, 40)\) ms, and \( \psi_3 \) is for \([80, 100)\) ms, etc. If there are no flows with RTTs in \([60, 80)\) ms, then \( \psi_4 = 0 \). By this way, we can classify RTTs into \( m \) groups indexed by \( i \), and let \( \psi_i \) be the percentage of RTTs in \( i \)th level. We can obtain \( \psi_i, i = 1, 2, \ldots, m \) as follows:

\[
\psi_i = \int_{i-1}^{i} g(x) \, dx, \quad \text{where } \psi_1 + \psi_2 + \cdots + \psi_m = 1.
\]

Let \( N_i (i = 1, 2, \ldots, m) \) be the number of flows of \( i \)th level, \( N \) be the total number of flows, i.e., \( N = \sum_{i=1}^{m} N_i \). It is clear that \( \psi_i = N_i/N \). Assume that \( N_i \) is also sufficient large, as we shall see later that the influence of the small amount of flows will be averaged out by the large amount of others flows and become negligible.

Let \( Q^N(t) \) be the real queue length at the beginning of time slot \( t \). The time slot of 1st group of flows is set to be 1 and the time slot of 2nd group of flows is set to be 2 and so on. We define the normalized queue length \( Q(t) \) (over \( N \) flows) by \( Q(t) = Q^N(t)/N \). Let \( W(t) \) be the normalized window size over the total number of flows \( N \) at time slot \( t \). We denote \( W^N(t) \) as the window size of flow \( i \) at time slot \( t \) and again we define the normalized window size (over \( N_i \) flows) for the \( i \)th group \( W_i(t) := \sum_{i=1}^{N_i} W^N_i(t)/N_i \). Further, we denote \( f^N_i(Q^N(t)) \) as the marking probability according to the real queue length and the normalized marking probability (over \( N \) flows) \( f(Q(t)) := f^N_i(Q^N(t)) \).

5.1. Normalized model for heterogeneous RTTs

In this section, we generalize our normalized model for homogeneous RTTs to capture the system dynamics for heterogeneous RTTs. According to the specification of TCP-newReno [25], the multiple marks/drops within one RTT will incur only one reduction in window size. Then, the marking probability of the \( i \)th group of flows, \( p_i(t) \), is given by

\[
p_i(t) = 1 - (1 - f(Q(t)))^{W_i(t)}.
\]

5.1.1. Normalized window dynamics

Since \( W(t) \) be the normalized window size over the total number of flows \( N \) at time slot \( t \), it can be written as

\[
W(t) = \sum_{i=1}^{m} \psi_i W_i(t).
\]

where the normalized window size for each group, \( W_i(t) \) \( (i = 1, \ldots, m) \), is being updated at every \( i \)th RTT. For \( i \)th RTT group, during the first \( i - 1 \) time slots, the window size remains unchanged because it takes \( i \) time slots for the congestion signal to arrive at the TCP senders. At the \( i \)th time slot, the window evolves according to AIMD: some flows in the \( i \)th group will increase their window sizes with probability \( 1 - p_i(t) \) and some flows will decrease their window sizes by half with probability \( p_i(t) \). In order to describe such dynamics, we use the expression \( \lfloor \frac{\text{mod } N_i - 1}{2} \rfloor \) to generate \( \{0, 0, \ldots, 1, 0, \ldots, 0\} \) sequence, where all the elements are zero except every \( i \)th position being one. This representation corresponds to the indicator function used in [32].

Hence, the window dynamics follows:

\[
W_i(t+1) = \left\{ \begin{array}{l}
1 - \left[ \frac{\text{mod } i - 1}{i - 1} \right] W_i(t) + \left[ \frac{\text{mod } i}{i - 1} \right] \\
\times \left( W_i(t) + 1 \right)(1 - p_i(t)) + \frac{W_i(t)}{2} p_i(t) \end{array} \right\}, \quad \text{(16)}
\]

where \( p_i(t) \) given in (14).

Next, by substituting (16) into (15), we have the normalized window size \( W(t) \) dynamics with respect to \( W_i(t) \) given by

\[
W(t+1) = \sum_{i=1}^{m} \psi_i \left\{ \left[ 1 - \left[ \frac{\text{mod } i - 1}{i - 1} \right] \right] W_i(t) + \left[ \frac{\text{mod } i}{i - 1} \right] \right. \\
\times \left( W_i(t) + 1 \right)(1 - p_i(t)) + \frac{W_i(t)}{2} p_i(t) \right\}.
\]

Note that if we define \( \eta_i(t) = W_i(t)/W(t) \) for \( i = 1, 2, \ldots, m \), from (15), we can rewrite the above in terms of \( W(t) \) and \( \eta_i(t) \) as follows:

\(^1\) The value of ‘\( m \)’ can be determined according to the system requirement. The larger ‘\( m \)’ leads to fine RTT granularity at the expense of more cumbersome description for system dynamics.
know that near the equilibrium point, \( \eta_i(t) \rightarrow W_i(\infty) \). By solving equations in (1), (14)–(16), we have

\[
W(t+1) = \sum_{i=1}^{m} \psi_i(t) \left\{ \left[ 1 - \frac{t \mod \bar{f}}{\bar{f} - 1} \right] \eta_i(t) W(t) + \left[ \frac{t \mod \bar{f}}{\bar{f} - 1} \right] \right\} 
\times \left[ (\eta_i(t) W(t) + 1)(1 - p_i(t)) + \frac{\eta_i(t) W(t)}{t} p_i(t) \right].
\]

(17)

5.2. Equilibrium point for heterogeneous RTTs

**Lemma 3.** If the system in (1), (14)–(16) converges, it must converge to the following unique equilibrium point:

\[
\lim_{t \to \infty} W(t) = C, \quad \lim_{t \to \infty} W_i(t) = \tilde{C}, \quad \lim_{t \to \infty} Q(t) = \frac{1}{\rho_1 \tilde{C} (1 + \bar{f})},
\]

where \( \tilde{C} = C[\psi_1 + \frac{1}{2} \psi_2 + \cdots + \frac{1}{m} \psi_m]^{-1} \).

**Proof.** As \( t \to \infty \), \( W(t+1) = W(t) \) and \( Q(t+1) = Q(t) \). By solving equations in (1), (14)–(16), we have \( W(t) = C \). Since \( Q(t) \) goes to \( Q(\infty) \), the marking probability seen by each TCP flow remains unchanged and hence the average window size is the same for all the flows, i.e., \( W_1(\infty) = W_2(\infty) = \cdots = W_m(\infty) \) and \( p_i(\infty) = p_m(\infty) = \cdots = p_m(\infty) \) by (14). Then, from (19), we have

\[
W_i(\infty) = \tilde{C}[\psi_1 + \frac{1}{2} \psi_2 + \cdots + \frac{1}{m} \psi_m]^{-1} = \tilde{C}.
\]

This gives \( W(\infty) = C[\psi_1 + \frac{1}{2} \psi_2 + \cdots + \frac{1}{m} \psi_m]^{-1} = \tilde{C} \). Given \( W(\infty) \), by solving (16), we obtain \( p_1(\infty) = \frac{1}{1 + \bar{f}} \). Finally, similarly as in Theorem 1, by using a linearized version of (14) when \( \tilde{C} = W_i(\infty) > 2 \), we obtain \( Q(\infty) \), and the result follows. □

Lemma 3 suggests that when the system is stable, the normalized window sizes for different groups of flows \( W_i(t) \) are the same regardless of different RTTs. Intuitively speaking, the coordination among all the flows is because they share the common queue. Since the queue length converges to constant if stable, the marking probability within one RTT is the same for different groups. This results in the same normalized window size for different groups.

5.3. Stability condition for heterogeneous RTTs

Based on Lemma 3, we now show a criterion for the local stability.

**Theorem 2.** The system described by (1)–(15) is locally stable, if both \( \|z_1\| < 1 \) and \( \|z_2\| < 1 \), where

\[
\bar{\lambda}_{1,2} = \frac{1}{2} \left[ \sigma_{11} + \sigma_{22} \pm \sqrt{\sigma_{11}^2 + \sigma_{22}^2 - 2\sigma_{11} \sigma_{22} + 4\sigma_{12} \sigma_{21}} \right],
\]

and \( \sigma_{11}, \sigma_{12}, \sigma_{21}, \sigma_{22} \) are given by (19).

Since near the equilibrium point \( f(Q(t)) \) is quite small in general, we can approximate \( p_i(t) \) by \( p_i(t) = f(Q(t)) W_i(t) \).

**Proof of Theorem 2.** For simplicity, we denote (17) and (1) by \( F^h_i(W(t), Q(t)) \) and \( F^h_2(W(t), Q(t)) \), respectively. We
Proof of Theorem 3. We look at the system dynamics for the $i$th group of flows while keeping all the others at their equilibrium points. As before, during the first $i-1$ time slot, the window size of $i$th group remains the same. Hence, for $j = 0, 1, \ldots, i-1$, the Jacobian matrix for the $i$th group ($W_i(t), Q(t)$) at $j$th time slot is given by

$$A_{ij} = \begin{pmatrix}
\frac{\partial W_i(t+j)}{\partial W_i} & \frac{\partial W_i(t+j)}{\partial Q(t+j)} \\
\frac{\partial Q(t+j)}{\partial W_i} & \frac{\partial Q(t+j)}{\partial Q(t+j)}
\end{pmatrix}. $$

At the $i$th time slot (i.e., $j=i$), the TCP sender will receive the congestion signal and update the window size according to AIMD rule. Then, we have $W_i(t+i) = (W_i(t+i-1) + 1)(1 - p_i(t+i-1)) + \frac{w_i(t+i)}{C_0} p_i(t+i-1)$. Applying this relationship to Jacobian matrix, we get

$$\frac{\partial W_i(t+i)}{\partial W_i} = 1 - \frac{1}{1 + \frac{w_i}{C_0}} - \frac{1}{C(1 + \frac{w_i}{C_0})} = \xi_{11},$$

$$\frac{\partial W_i(t+i)}{\partial Q} = -p_i C \left(1 + \frac{w_i}{C_0}\right) = \xi_{12},$$

$$\frac{\partial Q(t+i)}{\partial W_i} = \frac{\xi_{11}}{1},$$

$$\frac{\partial Q(t+i)}{\partial Q} = 1 + \frac{\xi_{12}}{1}. $$

Then the Jacobian matrix at $i$th time slot $A_{ii}$ is given by

$$A_{ii} = \begin{pmatrix}
\xi_{11} & \xi_{12} \\
\xi_{12} & 1 + \frac{\xi_{12}}{1}
\end{pmatrix}. $$

Since the RTT for the $i$th group of flows is $i$ time slots, within the first $i-1$ time slots, the TCP sender does not receive any congestion signal and the window size for the $i$th group flows $W_i(t)$ remains unchanged. Then, we have $W_i(t+1) = W_i(t)$ for $t = 0, 1, \ldots, i-1$, and by applying this to the Jacobian matrix, we get

$$A_{ii} = \begin{pmatrix}
1 & 0 \\
0 & 1
\end{pmatrix}, \quad \text{for } j = 1, 2, \ldots, i-1.$$

The total Jacobian matrix over $i$ time slots $A_i$ is given by

$$A_i = A_{i,1} \times A_{i,2} \times \cdots \times A_{i,i-1} = \begin{pmatrix}
\xi_{11} + (i-1)\xi_{12} & \xi_{12} \\
\xi_{12} & \xi_{12} + (i-1)\xi_{12}
\end{pmatrix}.$$  

where $\xi_{11}$ and $\xi_{12}$ are from (20) and (21). Then, the local stability follows by enforcing all the solutions of det$\lambda I - A_i = 0$ to be inside a unit circle. □

5.5. Discussion

Lemma 3 shows that the equilibrium point is determined by the RTT and the associated weight $\psi_i$ for group $i$. This indicates that a single flow or a group of flows with small weight ($\psi_i$) does not contribute much to the equilibrium point and plays a minor role when there exist a large amount of other flows. The same observation can be applied to Theorems 2 and 3. The stability of the whole system will be largely unaffected by a single flow with long RTT, and this is in contrast to Theorem 2 in [2], where the RTT of each flow may affect the maximum RTT $\tau$ or minimum RTT $\tau$ and hence affect the stability of the system considered in [2]. Note that the model in [2] is more suitable for a small system with a small number of flows, while our results suggest that from a whole system point of view, we apply different stability criterion for a large system with many flows taking into account the proportion of flows with similar RTT. Specifically, Theorem 3 allows us to investigate how much each group of flows contributes to the whole system stability, telling us how the stability would change if we increase/decrease the number of a particular set of flows. Later in Section 6.2, we further demonstrate the utility of Theorems 2 and 3 via simulation results.

6. Simulation results

In the next section, we will provide the simulation results for the homogeneous RTTs and heterogeneous RTTs to justify Theorems 1 and 2, respectively.

6.1. Homogeneous RTTs

For homogeneous RTTs, we present ns-2 simulation results to show that (i) for a TCP-newReno/RED system with many flows, if it is stable, then it will converge to the unique equilibrium point given by (6) and (7); (ii) if the system is locally stable and $\Delta q$ is not too small, it is then also globally stable.

We simulate the system as in Fig. 1(a) with parameters listed in Table 2, Figs. 8(a) and (b) show the normalized queue-length and the average window size, respectively, where the number of flows ranges from 10 to 1000. The lower graphs display magnified versions of the upper graphs over [400, 500] (fine time scale).

By solving (13) with the parameters given in Table 2, we have $C \approx 6.70$. Then, from (5)-(7), we obtain $W(\infty) = 6.70$ and $Q(\infty) = 4.96$. Fig. 8 shows that $W(t)$ converges to 6.71 and normalized queue-length converges to 4.94, which are in good agreement with the analytical results. Fig. 8 also shows that as the number of flows increases, the variation becomes smaller, indicating that our model can effectively
capture the system behavior especially when the number of flows in the system is large.

Next, we provide experimental results for global stability. We use the same simulation parameters in Table 2, but different initial window size $W(0)$ and normalized queue size $Q(0)$. It is easy to verify that the configuration is locally stable by Theorem 1 and $\Delta q = q_2 - Q(\infty) = 4.06$. Fig. 9 shows that the average window size and the normalized queue-length all converge to their equilibrium points, regardless of their initial window sizes tested in our simulations. We have also run many other simulations using different initial queue-length and obtain similar results.

### 6.2. Heterogeneous RTTs

We here consider two groups of flows. The first group consists of 200 TCP flows with two-way propagation delay of 40 ms and the second group has 20 TCP flows with two-way propagation delay of 320 ms. They all pass through a common bottleneck router with capacity $200 \text{ kbps} \times (20 + 20) = 44 \text{ Mbps}$. In this setup, we obtain the system parameters as follows: the average queueing delay is around 100 ms. The RTT for the first group of flows is $40 + 100 = 140$ ms, and the RTT for the second group is $320 + 100 = 420$ ms. Then, we can set the length of each time slot to 140 ms, and one RTT can be discretized into three levels ($m = 3$). The first group of flows fall into the time slot 1 (RTT $\leq 140$ ms), the second group of flows are in time slot 2 (RTT $\in [140, 280]$), and there is no flow in time slot 2 (RTT $\in [280, 420]$). Hence, we have $\psi_1 = 200 / (200 + 20) = 10/11$, $\psi_2 = 0/220 = 0$, and $\psi_3 = 20/220 = 1/11$. In addition, according to Lemma 3, we have $Q(\infty) = 1.5873$ and $W(\infty) = 7.0$. Table 3 summarizes the parameter settings for heterogeneous RTT case.

From this parameter setting, we obtain the Jacobian matrix of Theorem 2 as follows:

![Diagram](image.png)

### Table 3

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>RTT of first group</td>
<td>140 ms</td>
</tr>
<tr>
<td>RTT of second group</td>
<td>420 ms</td>
</tr>
<tr>
<td>$m$</td>
<td>3</td>
</tr>
<tr>
<td>$\rho_1$</td>
<td>0.02</td>
</tr>
<tr>
<td>$\psi_1$</td>
<td>10/11</td>
</tr>
<tr>
<td>$\psi_2$</td>
<td>0</td>
</tr>
<tr>
<td>$\psi_3$</td>
<td>1/11</td>
</tr>
<tr>
<td>$C$</td>
<td>7.0 pkt/slot</td>
</tr>
<tr>
<td>$Q(\infty)$</td>
<td>1.5873</td>
</tr>
<tr>
<td>$W(\infty)$</td>
<td>7.0</td>
</tr>
<tr>
<td>$\eta$</td>
<td>1.0645</td>
</tr>
</tbody>
</table>

Fig. 8. The average window size and normalized queue-length under different number of flows $N$. (a) Average window size. (b) Normalized queue length.

Fig. 9. The average window size and instantaneous queue-length $Q^s(t)$ with different initial window size $W(0)$ and under fixed $Q(0) = 5$. 

Fig. 9. The average window size and instantaneous queue-length $Q^s(t)$ with different initial window size $W(0)$ and under fixed $Q(0) = 5$. 

\[ g = 1.0645 \]
\[
B = B_3 \times B_2 \times B_1 \n\]
\[
= \begin{pmatrix}
0.73169 & -0.66161 \\
0.73169 & 0.33839
\end{pmatrix} \times
\begin{pmatrix}
0.74035 & -0.64027 \\
0.74035 & 0.35973
\end{pmatrix} \times
\begin{pmatrix}
0.74035 & -0.64027 \\
0.74035 & 0.35973
\end{pmatrix} 
\]
\[
= \begin{pmatrix}
-0.4846 & -0.2874 \\
0.3298 & -0.6320
\end{pmatrix}
\]

where \( B_1, B_2 \) and \( B_3 \) are given by (18). We can calculate the eigenvalue of matrix \( B \) as \( \lambda_{1,2} = -0.5583 \pm 0.2989 \) and \( |\lambda_{1,2}| = 0.6333 < 1 \).

On the other hand, if we only consider the first group of flows in the system, the associated Jacobian matrix \( A_1 \) in Theorem 1 becomes
\[
A_1 = \begin{pmatrix}
0.8889 & -0.6300 \\
0.8889 & 0.3700
\end{pmatrix},
\]
whose eigenvalues are \( \lambda_{1,2} = 0.6294 \pm 0.7019 \) with \( |\lambda_{1,2}| = 0.9428 < 1 \). Similarly, we can express the associated Jacobian matrix \( A_2 \) by Theorem 1 as
\[
A_2 = \begin{pmatrix}
0.9565 & -4.8300 \\
0.9565 & -3.8300
\end{pmatrix},
\]
whose eigenvalues are \( \lambda_{1,2} = -0.3843, -2.4892 \) with \( |\lambda_{2}| = 2.4892 > 1 \).

Now, we are ready to compare the simulation results with Theorems 2 and 3. Fig. 10(a) shows that the mixed two groups of flows yield the stable queue dynamics, where the eigenvalue of \( B \) is inside a unit circle. From the eigenvalues of \( A_1 \) and \( A_2 \), we know that the first group is stable and the second group is unstable. Thus, we conduct the following two sets of simulations: (i) only the first group of flows passes through the bottleneck router with a ‘scaled-down’ capacity \( 200 \text{ kbps} \times 20 = 4 \text{ Mbps} \); (ii) only the second group of flows pass through the bottleneck router with capacity proportional to the number of flows, \( 200 \text{ kbps} \times 200 = 40 \text{ Mbps} \).

We plot the corresponding queue dynamics of each simulation setup in Figs. 10(b) and (c), respectively. We see that the second group displays wild fluctuation implying instability (the eigenvalue of \( A_2 \) is outside of a unit circle as predicted by Theorem 1), and the first group shows much less fluctuation (stable). We then increase the number of flows in the first group from 200

---

\(^2\) The use of a ‘scaled-down’ capacity is equivalent to replacing the other group of flows by their average window sizes (constant) under the original capacity \( 200 \text{ kbps} \times 220 = 44 \text{ Mbps} \).
to 1000 while the number of flows in the second group remains the same. Consequently, $\psi_1$ increases and $\psi_2$ decreases, which means the influence of the second group is diluted by the first group and the first group (stable) dominates the system stability in such a setting as shown in Fig. 10(d) (through similar approach, we can obtain the matrix $B$ whose eigenvalues are all inside a unit circle).

7. Conclusion

In this paper, we investigate the local and global stability of a TCP-newReno system with RED queue under many flows’ regime. The existing local stability literature is designed for TCP-Reno not for TCP-newReno. As to the global stability, traditional approaches rely on constructing a suitable Lyapunov function. However, for a typical TCP-newReno/RED system with many flows, it is generally infeasible to find the Lyapunov function as the entire system dynamics are too complicated to be tracked. In this paper, we present a normalized model for the TCP-newReno/RED with many flows. This normalized model is able to capture the essential system dynamics, and at the same time simple enough to allow fast numerical analysis. Based on the normalized model, we derive the local stability criterion and show how to make a locally stable TCP-newReno/RED system also globally stable by properly choosing some of its parameters. Our extensive numerical analysis also suggests that in most cases, any locally stable TCP/RED system is also globally stable, and this trend becomes clearer as the number of flows increases. Finally, we extend our model to the case of heterogeneous RTTs.

References

Xinbing Wang was born in Jilin, China. He received the B.S. degree (with honors) from the Department of Automation, Shanghai Jiaotong University, Shanghai, China, in 1998, and the M.S. degree from the Department of Computer Science and Technology, Tsinghua University, Beijing, China, in 2001. He received his Ph.D. degree major in the Department of Electrical and Computer Engineering, minor in Department of Mathematics, North Carolina State University, Raleigh, NC, USA in 2006. Currently, he is a faculty member in Department of Electronic Engineering, Shanghai Jiaotong University, Shanghai, China. His research interests include Resource Allocation and Management in Mobile and Wireless Networks; TCP Asymptotics Analysis; Wireless Capacity; Cross Layer Call Admission Control; Asymptotics Analysis of Hybrid Systems, and Congestion Control over Wireless Ad-hoc and Sensor Networks. He is a member of Technical Program Committee of IEEE ICC 2007, IEEE WCNC 2007, IEEE ICCCN 2007 and IEEE IPCCC 2007.

Do Young Eun received his B.S. and M.S. degree in Electrical Engineering from Korea Advanced Institute of Science and Technology (KAIST), Taejon, Korea, in 1995 and 1997, respectively, and Ph.D. degree from Purdue University, West Lafayette, IN, in 2003. Since August 2003, he has been an Assistant Professor with the Department of Electrical and Computer Engineering at North Carolina State University, Raleigh, NC. His research interests include network modeling and analysis, congestion control, resource allocation, and ad-hoc/sensor networks. He is a member of Technical Program Committee of IEEE INFOCOM 2005–2007, IEEE ICC 2005–2007 IEEE Globecom 2005, and IEEE IPCCC 2006. He received the Best Paper Awards in the IEEE ICCCN 2005 and IEEE IPCCC 2006, and the NSF CAREER Award 2006.