Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 9947883, 9 pages
https://doi.org/10.1155/2021/9947883

Research Article

An Improved Algorithm to Enhance the Performance of FAST TCP Congestion Control for Personalized Healthcare Systems

Jiyang Ye, Bowen Huang, and Xiaolong Chen

Jinhua Polytechnic, Jinhua 321017, China

Correspondence should be addressed to Jiyang Ye; 20050879@jhc.edu.cn

Received 9 March 2021; Revised 1 May 2021; Accepted 8 June 2021; Published 21 June 2021

Academic Editor: Arun K. Sangaiah

The development of personalized medical systems should be supported by a fast and stable network system. The FAST TCP network system is the appropriate support system for this purpose. However, when the FAST TCP is deployed, the static mapping selection method for protocol parameters is unable to guarantee the small queuing delay and fast convergence of the network simultaneously. By conducting theoretical analysis and simulation experiments, the relationships among FAST TCP protocol slow start condition, control law gain parameters, and FAST TCP system convergence rate were examined. To ensure the stability of the FAST TCP system and to select the smallest protocol parameters, an improved method to effectively accelerate the convergence velocity of the FAST TCP system is proposed in this study. In this method, the number of packets for staying in the buffer for FAST TCP connections was taken as the criterion of the slow start, and the gain parameter of the control law was dynamically adjusted according to the local information of each FAST TCP connection. Using this improved method, the FAST TCP system can achieve a stable and small queuing delay, whilst the FAST TCP system could converge quickly to the equilibrium point simultaneously.

1. Introduction

With the continuous advancement of the digital wave, more and more hospitals are actively rolling out personalized healthcare systems with the goal of achieving rapid development in the field of intelligent medical care. However, personalized healthcare system requires unilateral transmission of the lossless compression radiographic images and pathology slides, as well as networks with low latency and high reliability [1]. At the same time, remote control business scenarios such as remote ultrasound, remote surgery, and remote ward rounds also present the characteristics of unilateral transmission, large bandwidth, and low delay. Although operators are constantly increasing the bandwidth, the speed of the medical network did not change with the increase of bandwidth in the same proportion [1, 2]. In the application of a personalized healthcare system, the network environment is still not very satisfactory. In order to improve the network speed in the medical environment, increasing the investment of hardware is not the best choice. Therefore, it is necessary to design a new algorithm to improve the network utilization rate. FAST TCP [2, 3], a new type of transmission control protocol based on the delay calculation for the next generation of high-performance networks with high bandwidth, long delay, and large capacity, can support the requirements of personalized medical systems. FAST TCP uses unilateral acceleration technology, keeping compatible with traditional TCP flow control mechanism, whilst not making any changes to the TCP/IP header field. Therefore, FAST TCP can communicate with all the traditional TCP protocols, and in the application of personalized medical system, it only needs to be deployed on the server side to bring significant acceleration effect. The FAST TCP protocol features a congestion controller at each connection source end and the application of the estimated queue delay as the congestion feedback signal. Based on the design idea of the FAST TCP network equilibrium, its performance indexes, such as stability, fast response, throughput, and fairness, have been proven to be better than those of TCP RENO, HSTCP, and STCP [3, 4]. However, choosing the appropriate protocol parameters to ensure that the FAST TCP protocol shows consistent good performance remains an open problem [2].

Studies have shown that protocol parameter α, control law gain parameter γ, and slow start threshold m of FAST TCP must be initialised when establishing a FAST TCP connection [2, 5]. The protocol parameter α represents the number of packets expected to remain in the link buffer when FAST TCP reaches equilibrium. Thus, under the precondition of a stable network system, a smaller protocol parameter α leads to a shorter queuing delay of the link [2, 6, 7]. However, past studies [8–10] have reported that setting a smaller protocol parameter α would lead to slower convergence speed of the system to the equilibrium point and reduced utilization of the network. Therefore, the existing method of static mapping selection for FAST parameters could not guarantee that the system would obtain a smaller queue delay and a faster convergence rate at the same time [11, 12]. A study [13] used measurement distributed facilities in the network to monitor the backbone links and then applied fuzzy control technology to guide the source end of each FAST TCP connection to select the appropriate protocol parameter α according to the link performance, so as to stabilise queue delay. However, the algorithm did not consider the improvement of the convergence speed of the system.

Protocol parameter α, control law gain parameter γ, and slow start threshold m of FAST TCP are found to be the key factors affecting the system [14]. If the settings are not appropriate, they could seriously affect system performance [15]. Given that the protocol parameter α refers to the number of packets expected to remain in the link buffer when FAST TCP reaches equilibrium, therefore, under the precondition of system stability, the smaller the protocol parameter α, the smaller the queue delay of the link [16]. However, past studies [2, 3] have shown that choosing a smaller protocol parameter could lead to a slower convergence speed of the system to the equilibrium point and the reduced utilization of the network. Literature [13] used measurement facilities distributed in the network to monitor the backbone link state and then applied fuzzy control technology to guide the source end of each FAST TCP connection to select appropriate protocol parameters according to the link state to stabilise queue delay; however, their proposed system did not improve the convergence speed of FAST TCP.

To overcome these shortcomings, the relationships among FAST TCP slow start criteria, control law gain parameters, and FAST TCP system convergence speed were discussed through theoretical analysis and NS-2 simulation. To ensure the stability of the FAST TCP system and select the smallest possible protocol parameters, an improved method to enhance the convergence speed of the system is presented. The key finding of this method considered the number of packets for staying in the buffer for FAST TCP connections as the criterion of a slow start and adjusted the control law gain parameters according to the local information of each connection. The proposed method also ensured that the system was stable and had a small queue delay, thereby improving the speed at which the FAST TCP system converges to the balance point.

The main contribution of this paper is to overcome the above defects; in the case of ensuring FAST TCP system stability, the FAST TCP flow chooses the protocol parameters as small as possible, adaptive dynamic adjustment slow start threshold and gain parameters according to the local information source end. This improved method can ensure that the system is stable and has small queuing delay at the same time, improving the speed of convergence to equilibrium system.

The rest of the paper is organised into sections. Section 1 introduces the basic concepts and algorithms of the FAST TCP network model. Section 2 analyses the relationship between the slow start threshold and system convergence speed through simulation experiments. This section also presents and analyses the reasons, proposes the number of packets for staying in the buffer for FAST TCP connections as the criterion to determine the slow start threshold, and determines the improved algorithm of slow start threshold based on the parameters of each connection protocol. In Section 3, the relationship between gain parameters and system convergence rate is discussed theoretically. Under the condition of system stability, the operation state of the network system was evaluated according to the local information of each connected source end. When the system was far from the equilibrium point, larger dynamic gain parameters were selected to bring the system as close to the equilibrium point as possible in the least amount of time. When the system was near the equilibrium point, smaller gain parameters were selected to reduce the instability of the system. When the system tended to balance, choosing a smaller gain parameter reduced the system instability. Section 4 discusses the simulation results under the dynamic environment, and the final section presents the conclusion.

2. Related Research Works

2.1. Basic Symbols. For the convenience of reading, the relevant basic symbols are introduced at first.

As shown in Figure 1, we assumed that the dumbbell network system contained N FAST TCP connections with the same link attributes and that they shared the unique bottleneck link l. We also defined the connections set $I = \{1, 2, \cdots, N\}$ and assumed that the buffer at the link end was large enough. We adopted the tail-dropping queue management algorithm. For the FAST TCP system, the following performance indicators were defined:

δ: overshoot refers to the difference between the maximum transmission flow obtained by each connection and the steady-state value and the ratio of this difference to the steady-state value

st: convergence time refers to the time when the new connection converges to the equilibrium point

For any given i connection, we defined $i \in I$ and set the following parameters:

$W_i(t)$: size of sending window on the source side (packets)

d_i: propagation delay(s) (we assumed that all connections obtained accurate propagation delay and that there was no fairness problem [7])

$q_i(t)$: queuing delay(s)

$D_i(t)$: go and back delay, $D_i = d_i + q_i$ (s)

$x_i(t)$: transmission traffic (packets/s), where $x_i(t) = W_i(t)/D_i(t)$

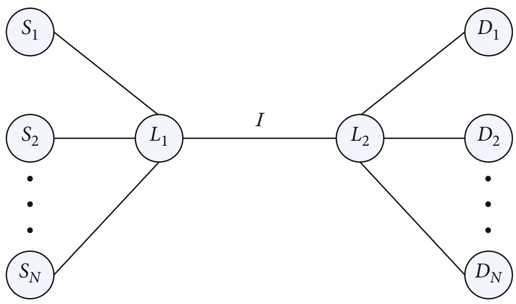

FIGURE 1: Dumbbell network topology.

$z_i(t) = x_i(t) \cdot q_i(t)$: the number of packets actually left in the link buffer [1, 6] (packets)

α_i: protocol parameters (packets), at equilibrium, the number of packets expected to stay in the link buffer

m_i: slow start threshold(s)

γ_i: gain parameters of control law

$e_i(t) = W_i(t)/d_i$: normalised window size, sum of packets left in buffer and packets left per unit time [1]

T: window update cycle(s)

For link l, we set the following:

C: the bandwidth is packets/s

$b_l(t)$: queue length is packets

$q_l(t)$: queuing delay, $q_l = b_l/c_l$ (s)

2.2. Research Works. The stability of the FAST TCP system is related to FAST TCP parameters and network parameters [2, 5]. The FAST TCP parameters include protocol parameter α representing the number of packets of data expected to remain in the link side buffer, control law gain parameter γ, and window update period T. The network parameters include propagation delay d and link bandwidth c. When the two sets of parameters do not cooperate, the stability of the system would be affected. Therefore, establishing the relationship between the two sets of parameters to guide the choice of protocol parameters is an effective solution to the disclosure problem [7–9].

Literature [2] proposed a guidance scheme for selecting protocol parameters based on link bandwidth by using a static mapping table. On the basis of experimental simulation, another study [8] pointed out that in a single-link network, the value should be greater than 0.0075 times of the link bandwidth. However, the above two schemes do not consider another network parameter d, which is an empirical value strategy, making it difficult to ensure the stability of the system under different network environments.

Assuming that window update cycle T is an RTT, Tan et al. [9] established a continuous congestion control model for a single-link single-connection FAST TCP network. The following conditions for the local asymptotic stability of the FAST TCP system are given by using the Routh–Hurwitz stability criterion [10].

Given network parameters c and d and protocol parameter γ, the protocol parameter α was selected to meet the following expression $\alpha > f(\gamma) \cdot c \cdot d$, where $f(\gamma) = ((\gamma^2 + 2\gamma - 4) + \gamma\sqrt{16\gamma + \gamma^2})/(2(\gamma^2 + \gamma + 2))$, thus indicating that the FAST TCP network congestion control models are asymptotically stable.

The stability condition connected the two sets of parameters and enabled the decoupling of the protocol parameters from other related parameters, thus guiding the selection of such parameters. However, the FAST TCP model used for stability analysis only considered that the window update period T is a special case of RTT and ignored the exponential smoothing filter used to estimate the congestion feedback signal at the source end. There are also some defects, such as conservative guidance scope and limited application scope. Thus, according to the stability conclusion found in the literature [11], the following guidance scheme is presented.

Under the constraint that parameter T is an RTT, $\gamma \in (0, 0.93]$, $c > 0$, and parameter d is bounded, we can guide the selection of protocol parameters α to ensure system stability within the scope of $[1, \infty)$.

The guidance scheme relaxed the selection range of the protocol parameters but restricted others [14, 15]. When other parameters do not meet the constraint conditions, the guidance scheme would be ineffective; thus, it can be said that the application range of the guidance scheme is limited.

According to the stability conclusion presented in past studies [8, 16], the following guiding scheme could be given: we can select $\alpha > 0$ or $\alpha > cd$ under constrained $\gamma d/T \in (0, 1)$ conditions. Similarly, this scheme is limited in terms of guiding the application scope.

3. FAST TCP Congestion Control Algorithms

In accordance with past studies [1, 2, 5] and analyses of the FAST TCP pseudocode, protocol parameter α_i was selected according to link bandwidth and slow start threshold $m_i = 0.00075$ (s) and control law gain parameter $\gamma_i = 0.5$ when the FAST TCP connection was established. Then, the sending window was updated according to Algorithm 1 [1].

The description of Algorithm 1 showed that the convergence speed of the FAST TCP protocol was related to the parameters α_i, m_i, and γ_i [1], among which α_i can be used to determine the balance point location of the FAST TCP system. If the α_i size was smaller, the convergence rate was slower. Meanwhile, the parameter m_i can be used to determine when the source changes from a slow start algorithm to a FAST window update algorithm. Parameter γ_i can be used to determine the proportion of window changes each time the FAST TCP algorithm is executed. Hence, the selection of these two parameters can affect the convergence rate. Therefore, under the condition of ensuring system stability, a smaller parameter α_i (smaller queue delay) was selected to promote the existing static selection method of parameters m_i and γ_i and to adjust them dynamically using local information available at each connection source. In this way, we can improve the FAST TCP protocol convergence speed and maintain a smaller queue delay.

```
Each acknowledgement frame is received {
    If connection establishment time is less than m_i,
    w_i(k' + 1) = w_i(k') + 1
      //Execute slow start window update algorithm
    Else if z_i(k') ≠ α_i
      Every window update cycle T///T is the window update cycle
      w_i (k + 1) = (1 − γ_i) · w_i(k) + γ_i (d_i/(d_i + q_i(k)) · w_i(k) + α_i)
      //Execute FAST TCP window update algorithm
}
```

ALGORITHM 1

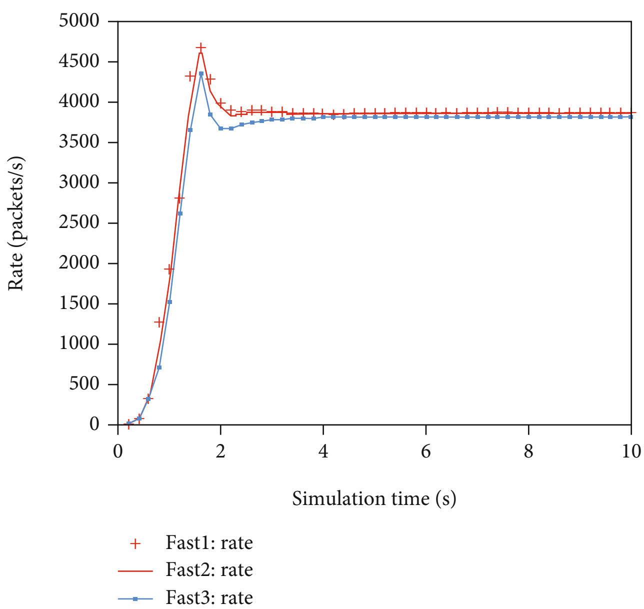

FIGURE 2: Each connection transmits traffic, connection start time, and overshoot when $C = 120000$ (packets/s), $\alpha_i = 200$.

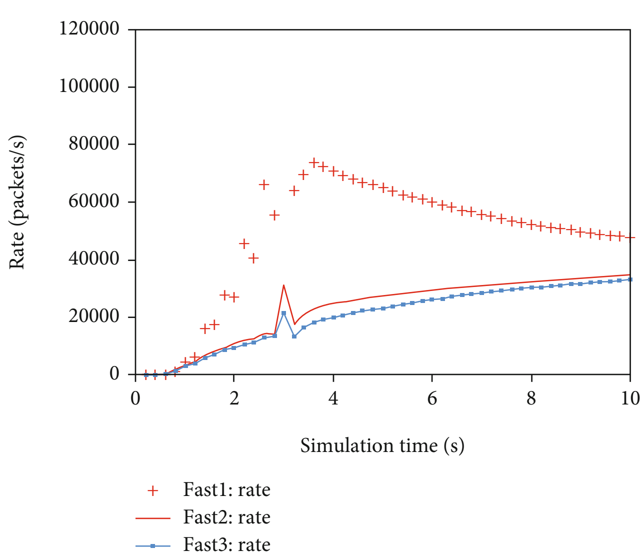

FIGURE 3: Each connection transmits traffic, connection start time, and overshoot when $C = 120000$ (packets/s), $\alpha_i = 200$.

4. Analysis and Improvement of Slow Start Threshold for FAST TCP Network Convergence Speed

4.1. Slow Start Queue Delay Ratio. As can be seen from Algorithm 1, the size of the slow start threshold was set to decide the time at which the slow start algorithm switched to the FAST TCP window in order to update the algorithm.

Firstly, we defined the meaning of each variable as shown below:

$T_1 = m_i$ is the execution time from the start of establishing the connection to the end of the slow start.

$T_2 = N \cdot \alpha_i / C$ is the queue delay for executing the FAST TCP window update algorithm when the system was balanced.

Queuing delay ratio was obtained from past studies [1, 5, 6]:

$$\frac{T_1}{T_2} = \frac{m_i}{N \cdot \alpha_i / C} = \frac{m_i C1}{N \cdot \alpha_i}. \quad (1)$$

Formula (1) represents the queue latency ratio of executing the slow start algorithm and executing the FAST TCP algorithm for a FAST TCP connection from the start of the establishment to the achievement of the equilibrium state.

We can see that the value obtained through Formula (1) changes when the relevant parameters N, α_i, m_i, and C change, thus indicating that the relative time of slow start algorithm execution also changes.

4.2. Simulations to Verify the Relationship between Queue Delay Ratio of Slow Start and Convergence Speed of FAST TCP System

Experiment 1. The number of connection N was set to 3, the simulation time was set to 10 (s), and the size of each packet was set to 1000 bytes. Next, we set the link transmission bandwidth C to 96 Mb/s ($C = 12000$ packets/s). FAST TCP sets the static mapping parameter as follows: $\alpha_i = 200$, $m_i = 0.00075$ (s), and $\gamma_i = 0.5$. The transmission delay at the FAST TCP source side was set to $d_i = 0.1$ (s). We tracked the data sending rate of FAST TCP connections. The simulation results are shown in Figure 2 and $T_1/T_2 = 0.015$ were obtained by using Formula (1). As shown in Figure 2, the convergence time of all connections (fast 1, fast 2, and fast 3, representing the flow rates of the three connections of the system, respectively) is 2

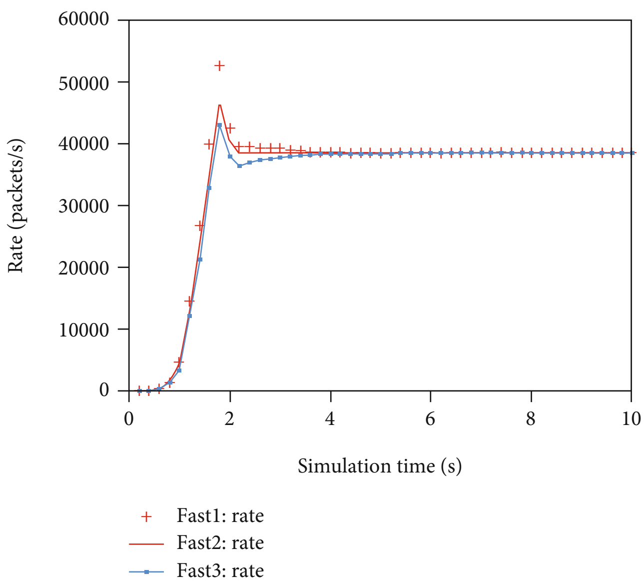

Figure 4: Each connection transmits traffic, connection start time, and overshoot, when $C = 120000$ (packets/s), $\alpha_i = 2000$.

Table 1: Relationships among slow start threshold, system overshoot, and convergence rate.

SN	α_i	m_i	C (packet/s)	N	d (s)	T_1/T_2	δ	st (s)
1	200	0.00075	12000	2	0.1	0.0225	12.4%	2.2
2	100	0.0075	12000	4	0.1	0.225	61.5%	2.2
3	200	0.0015	12000	2	0.1	0.045	14.5%	2.1
4	100	0.003	12000	4	0.1	0.09	8%	1.7
5	400	0.004	12000	1	0.1	0.12	27.5%	1.9
6	200	0.006	12000	2	0.1	0.18	43.4%	2.0
7	200	0.00075	12000	4	0.1	0.01125	41.6%	1.9
8	200	0.0075	12000	4	0.1	0.1225	71.5%	2.2
9	400	0.0015	12000	2	0.1	0.0230	23.5%	1.95
10	400	0.003	12000	2	0.1	0.047	27.6%	1.85
11	200	0.005	12000	4	0.1	0.081	21.3%	1.8
12	200	0.006	12000	4	0.1	0.099	20%	1.75
13	200	0.007	12000	4	0.1	0.105	66%	2.1
14	250	0.00075	12000	4	0.15	0.009	21.5%	3.05
15	250	0.0075	12000	4	0.15	0.09	20.5%	2.75
16	500	0.005	12000	2	0.15	0.06	33.3%	3.3
17	500	0.009	12000	2	0.15	0.108	29.1%	2.8

(s), and the overshoot is 25. Thus, the system had a faster convergence speed and a smaller overshoot.

Experiment 2. The link transmission bandwidth C was set to 960 Mb/s ($C = 120000$ packets/s). The other parameters were the same as in Experiment 1. The simulation results are shown in Figure 3.
The value of $T_1/T_2 = 0.15$ obtained from Formula (1) indicated that each connection accounted for a large proportion in the slow start phase compared with Experiment 1. Therefore, in Figure 3, we can see that the convergence time of the system is st > 10 (s), and the overshoot is $\delta > 200\%$.

Using Formula (1), we determined that when the bandwidth increased, the (1) ratio can be kept constant by increasing the protocol parameter α_i.

Experiment 3. Here, we increased the protocol parameter $\alpha_i = 2000$, but the other parameters were still the same as in Experiment 2. The simulation results are shown in Figure 4. The value of $T_1/T_2 = 0.015$ obtained from Formula (1) was the same as that in Experiment 1. Therefore, in Figure 3, it can be seen that the convergence time of the system is st = 2 (s) and the overshoot $\delta = 25\%$, which are the same as in Experiment 1. The performance of the proposed method is better in Experiment 1 compared to Experiment 2. However, it is known from the literature [1] that the queuing delay of link in Experiment 2 is 0.005 s in a balanced state, whilst the bottleneck queuing delay in Experiment 3 increased to 0.05 s when the value of α_i increased to 2000.

Experiment 4. Table 1 presents the simulation values of system overshoot and convergence time when the network is guaranteed to be in a stable state. The relevant parameters of the network and FAST TCP protocol are different from those in Table 1.

The experimental results of Experiments 1–4 indicate the following:

(1) When network parameters changed, the queue delay ratio of the slow startup queue of (1) changed, which in turn affected the performance of the FAST TCP protocol (overshoot and convergence time)

(2) Increasing the protocol parameter α_i under the same conditions improved the convergence speed of the system; however, doing so also increased the link queuing delay

(3) When different slow start thresholds were chosen, the networks exhibited different levels of performance. Comparing the data of groups 1, 7, and 14 with those in 4, 12, and 15 in Table 1, we can see that the default value of the slow start threshold currently used was not optimal. From the data of groups 4, 12, and 15, the network showed smaller overshoot and faster convergence speed when the corresponding slow start threshold was set at $T_1/T_2 \in (0.09, 0.1)$

4.3. FAST TCP Window Update Algorithm for Improving Slow Start Conditions. On the basis of the above experiments and on Formula (1), we used the current queuing delay parameter as the criterion of the slow start threshold. When the network-related parameters changed, the value T_1/T_2 also changed along with the performance of the FAST TCP protocol. Therefore, we defined the parameter number $l_i = \alpha_i/G$ ($G = 10$) of packets for each connection left in the buffer, which was used as the slow start threshold for each connection. In this way, even if the bandwidth changes along with the number of connections and other parameters, the value will be kept within the range of (0.09, 0.1), thus ensuring good

$l_i = \alpha_i/10$.
Each acknowledgement frame is received {
If (it was the first time a connection is made and $z_i(k') < l_i$)
//$z_i(k')$: the number of data packets actually left in the link buffer at the time of connection i at $k \cdot T$ [8]
$w_i(k'+1) = w_i(k') + 1$//execute slow start window update algorithm
Else if $(z_i(k) \neq \alpha_i)$
Every slot T {
$w_i(k+1) = (1-\gamma_i)\cdot w_i(k) + \gamma_i (d_i/(d_i + q_i(k))\cdot w_i(k) + \alpha_i)$
//Execute FAST TCP window update algorithm
}

ALGORITHM 2

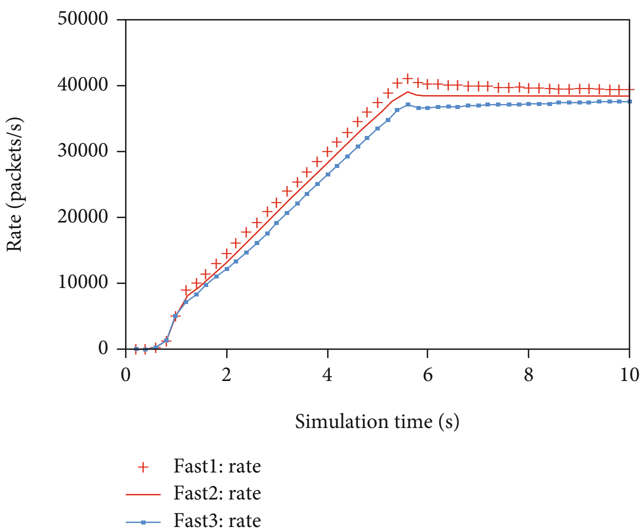

FIGURE 5: Each connection transmits traffic, connection start time, and overshoot based on improved Algorithm 2 when $c = 120000$ (packets/s), $\alpha_i = 200$.

network performance. The specific improved algorithm is presented below.

The validity of Algorithm 2 is verified below.

Experiment 5. In this experiment, the parameter settings were the same as those in Experiment 2, but the source side used the improved Algorithm 2. After tracking the data sending rate of the FAST TCP streams, the simulation results indicated that the convergence time of the FAST TCP system was st = 6 (s), as shown in Figure 5. The overshoot δ was less than 6%. Compared to Experiment 2, Algorithm 2 significantly reduced the overshoot and improved the convergence speed of the system without increasing the protocol parameter α_i (i.e., queuing delay).

5. Analysis and Improvement of Control Law Gain Parameters on the Protocol Convergence Rate

5.1. Theoretical Analysis. Previous studies [2, 3] have introduced a more accurate FAST protocol model, including self-synchronization and integral link, and the Nyquist stability criterion was applied to show that the system was asymptotically stable for bounded round-trip delay in a single bottleneck link when the value range of control rate gain parameter γ was (0,0.93).

Firstly, we analysed the influence of gain parameter γ_i of control law on the convergence rate of FAST TCP system when $\gamma_i \in (0,0.93)$.

We divide both sides of Equation (2) by propagation delay d_i, thus arriving at

$$S_i(t+T) = (1-\gamma_i)\cdot S_i(t) + \gamma_i\left(\frac{1}{d_i + q_i(t)}\cdot S_i(t) + \frac{\alpha_i}{d_i}\right), \quad (2)$$

which we organised as follows:

$$S_i(t+T) = \left(1 - \frac{\gamma_i \cdot q(t)}{d_i + q(t)}\right)\cdot S_i(t) + \gamma_i \cdot \widehat{\alpha}_i, \text{ which } \widehat{\alpha}_i = \frac{\alpha_i}{d_i}. \quad (3)$$

Theorem 1. *We also assumed that the γ_i values of all FAST TCP connections were equal and satisfied $\gamma_i \in (0, 0.93)$, that $S(t) = \sum_i S_i(t)$, and that the sizes of the FAST TCP system at exit slow start and equilibrium point were $S(0)$ and S^*, respectively. Thus, for any small positive number ε, the convergence time st satisfies the following:*

$$st \geq \frac{\ln(\varepsilon/|S(0) - S^*|)}{\ln(1-\gamma_i)}, \quad so\ ||S(st) - S^*| < \varepsilon, \quad which\ S^* = c + \sum_i \widehat{\alpha}_i.. \quad (4)$$

Proof. From (3), we arrive at the following:

$$S_i(t+T) = (1-\gamma_i)\cdot S_i(t) + \gamma_i\left(\frac{d_i(t)\cdot S_i(t)}{d_i + q(t)} + \widehat{\alpha}_i\right). \quad (5)$$

According to a past study [1], $\sum_i (d_i(t)\cdot S_i(t))/(d_i + q(t)) = C$, and this can be obtained by summing up the source terminals of Equation (5) as follows:

$$S(t+T) = (1-\gamma_i)\cdot S(t) + \gamma_i \cdot \left(c + \widehat{\alpha}\right), \quad (6)$$

where $\widehat{\alpha} = \sum_i \widehat{\alpha}_i$. □

FAST TCP window update algorithm with an improved slow start condition and gain parameters: $l_i = \alpha_i/10$.
Each acknowledgement frame is received {
    If (it was the first time a connection is made and $z_i(k') < l_i$)
        $w_i(k'+1) = w_i(k') + 1$//execute slow start window update algorithm
    Else if $(z_i(k) \neq \alpha_i)$
        Every slot T {
            {Flag = 1;
    $\eta_i = ABS(x_i(t) \cdot q_i(t) - \alpha_i)$;
    $\mathrm{diff}_i = \min\,(\eta_i/\alpha_i, 1)$
            $\gamma_i = 0.5 + 0.5 \cdot ((e^{\kappa \cdot \mathrm{diff}_i} - 1)/(e^{\kappa} - 1))\ \kappa < -1$
        $w_i(k+1) = (1-\gamma_i)w_i(k) + \gamma_i((d_i/(d_i + q_i(k)))w_i(k) + \alpha_i)$
        }
    }

Algorithm 3: The algorithm is described as follows.

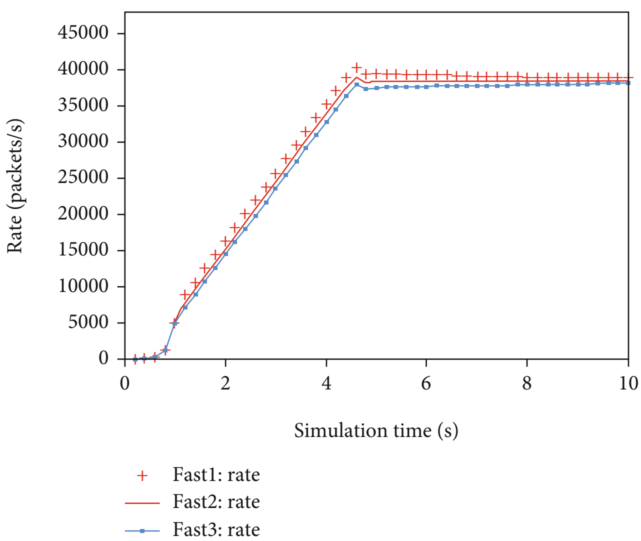

Figure 6: When $C = 120000$ (packets/s) and $\alpha_i = 2000$, based on the improved Algorithm 3 of each connection transmission flow, connection start time and overshoot.

Then, we can transform Equation (6) into

$$S(t+T) - \left(c + \widehat{\alpha}\right) = (1-\gamma_i) \cdot \left(S(t) - \left(c + \widehat{\alpha}\right)\right). \quad (7)$$

Equation (7) means an equal ratio sequence wherein the first term is $S(0) - (c + \widehat{\alpha})$, and the common ratio is $1 - \gamma_i$ for any $\mathrm{st} = k \cdot T$. Thus, we have

$$S(\mathrm{st}) = c + \widehat{\alpha} + \left(Y(0) - \left(c + \widehat{\alpha}\right)\right) \cdot (1-\gamma_i)^{\mathrm{st}}. \quad (8)$$

From Equation (8), we know that $S(\mathrm{st})$ converges to equilibrium point S^*, and S^* satisfies

$$S^* = c + \widehat{\alpha}. \quad (9)$$

Therefore, Equation (8) can be expressed as follows:

$$S(\mathrm{st}) = S^* + \left(S(0) - S^*\right) \cdot (1-\gamma_i)^{\mathrm{st}}. \quad (10)$$

From Equation (10), we know that for any small positive number ε, when the convergence time is

$$\mathrm{st} \geq \frac{\ln\,(\varepsilon/|S(0) - S^*|)}{\ln\,(1-\gamma_i)}, \quad (11)$$

we have $|S(\mathrm{st}) - S^*| \leq \varepsilon$.

The numerator of Equation (11) is negative, which means that the larger the gain parameter γ_i of the control law, the shorter the time for the system to converge to the equilibrium point.

Each FAST TCP connection can calculate the number $z_i(t) = (w_i(t)/R_i(t))q_i(t)$ of the link buffers actually left by the connection according to the local information $(w_i(t), q_i(t), d_i, \alpha_i)$ obtained by the source. Then, we calculate the absolute position $\eta_i = |z_i(t) - \alpha_i|$ and relative position η_i/α_i of the actual number of link buffers, after which the expected equilibrium point and the relative position are normalised to $\mathrm{diff}_i = \min\,(\eta_i/\alpha_i, 1)$. Under the same conditions, the larger the absolute position η_i relative to the normalised position diff_i, the larger the convergence time st. As shown in Equation (11), the convergence time st decreases with the increase of the gain parameter γ_i of the control law. Therefore, the following exponential function formula can be used to ensure the stability of the system:

$$\gamma_i = 0.5 + 0.5 \cdot \frac{\left(e^{\kappa \cdot \mathrm{diff}_i} - 1\right)}{e^{\kappa} - 1} (\kappa < -1). \quad (12)$$

The gain parameter γ_i was selected according to the standardised relative position. When the system is far away from the equilibrium point (diff_i value is larger), the larger gain parameter γ_i is selected so that the system could quickly converge to the equilibrium point. When the system is near the equilibrium point (diff_i value is smaller), in order to reduce the overshoot and jitter, the smaller gain parameter γ_i could

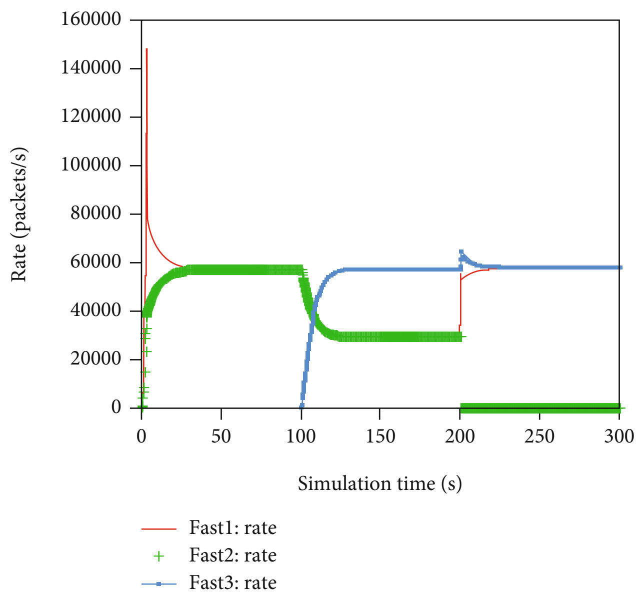

Figure 7: The convergene time and overshoot with the original FAST TCP algorithm.

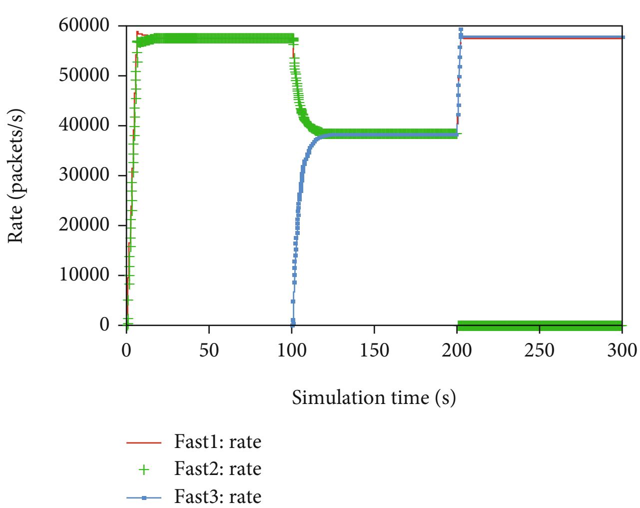

Figure 8: The convergence time and overshoot with the improved FAST TCP algorithm.

be selected to ensure that the system has less jitter near the equilibrium point.

5.2. FAST TCP Window Update Algorithm for Improving the Gain Control Law Gain Parameters. Algorithm 3 presents a FAST TCP window update algorithm, which can improve the slow start threshold and control law gain parameters simultaneously.

5.3. Simulation Verification. The effectiveness of Algorithm 3 is verified below.

Experiment 6. The parameter setting was the same as in Experiment 2, but the source side adopted improved Algorithm 3. The parameter of Algorithm 3 was $\kappa = -10$. The simulation results are shown in Figure 6. As can be seen in the figure, the convergence time of the system was $\mathrm{st} \le 4$ (s), and the overshoot was $\delta \le 5\%$. Compared with Experiments 2, 3, and 5, the improved Algorithm 3 reduces the overshoot of the system and improves the convergence speed of the system without increasing the protocol parameter α_i (i.e., queuing delay).

6. Dynamic Environment Simulation

When FAST TCP connections in the bottleneck link arrive and depart dynamically, the improved slow start threshold and control law gain parameter FAST TCP algorithm could also reduce the overshoot of the system and improve the convergence speed of the system without increasing the protocol parameters α_i (i.e., queuing delay).

Experiment 7. In this experiment, the link bandwidth C was 960 Mb/s, $\alpha_i = 200$, $d_i = 0.1$ (s), the fast 1 and fast 2 connections were established at the beginning, the fast 3 connection was established at 100 s, and the fast 2 connection was released at 200 s. If there was no improved algorithm, we take $m_i = 0.00075$ (s), $\gamma_i = 0.5$. The simulation results in Figure 7 revealed the convergence time of FAST TCP system $\mathrm{st} > 20$ (s) and overshoot $\delta > 200\%$ at the start of the system. When the new FAST TCP flows joined and departed, the convergence time of the FAST TCP system was $\mathrm{st} > 20$ (s). Under the same conditions cited above, the source side adopted the improved Algorithm 3 without setting parameters m_i, γ_i (assuming that each fast connection obtained accurate propagation delay and could allocate bandwidth fairly). The simulation results in Figure 8 revealed that the convergence time of the FAST TCP system was $\mathrm{st} = 4$ (s) and the overshoot was $\delta = 5\%$.

7. Conclusion

In this paper, the number of remaining link buffers was taken as the judgment condition of a slow start. This ensured that the distribution of queuing delay ratio was within the expected range and would not change with the change of bandwidth, the number of connections, and other parameters, thus improving the convergence speed of the system. At the same time, an improved method of dynamically adjusting the gain parameters of the controller was proposed. According to the local information (sending window, protocol parameters, RTT, and propagation delay), which can be obtained by each connection source, this method judged the network operation state and adjusted the gain parameters of control law according to the network operation state within the range of relevant protocol parameters to ensure the stability of FAST TCP system.

The simulation results showed that the two improved methods overcame the defects of static selection of protocol parameters in the original FAST TCP algorithm and ensured a stable and small queuing delay in the system. At the same time, the speed of convergence to the equilibrium point was improved, and high speed, large connection, and low delay were achieved for personalized healthcare systems.

In the future, each connection cannot communicate directly, so each connection cannot synchronously adjust the protocol parameters, and the convergence speed of the protocol parameter adjustment is slow. The next step is to discuss the synchronization of connection protocol parameters so as to provide fairness in bandwidth allocation.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This project is supported by the Zhejiang Provincial Natural Science Foundation of China under Grant LY18F010018 and the Scientific Research Fund of Zhejiang Provincial Education Department of China under Grant Y202043591.

References

[1] H. Sellak and M. T. Grobler, "Towards intelligent, multidimensional context-aware personalised healthcare and wellness support services," *Studies in Health Technology and Informatics*, vol. 266, pp. 168–173, 2019.

[2] D. X. Wei, C. Jin, S. H. Low, and S. Hegde, "FAST TCP: motivation, architecture, algorithms, performance," *IEEE/ACM Transactions on Networking*, vol. 14, no. 6, pp. 1246–1259, 2006.

[3] R. S. Prasad, C. Dovrolis, and M. Thottan, "Router buffer sizing for TCP traffic and the role of the output/input capacity ratio," *IEEE/ACM Transactions on Networking*, vol. 17, no. 5, pp. 1645–1658, 2009.

[4] A. K. Sangaiah, J. S. Ramamoorthi, J. J. P. C. Rodrigues, M. A. Rahman, G. Muhammad, and M. Alrashoud, "LACCVoV: linear adaptive congestion control with optimization of data dissemination model in vehicle-to-vehicle communication," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–10, 2020.

[5] H. Zhang and L. Chen, "CODA: toward automatically identifying and scheduling coflows in the dark," in *Proceedings of the 2016 ACM SIGCOMM Conference*, pp. 160–173, New York: ACM, 2016.

[6] M. Liu, G. Zhong, Y. He, K. Zhong, H. Chen, and M. Gao, "Fast HEVC inter-prediction algorithm based on matching block features," *International Journal of Mobile Computing and Multimedia Communications*, vol. 9, no. 1, pp. 40–61, 2018.

[7] A. K. Sangaiah, D. V. Medhane, T. Han, M. S. Hossain, and G. Muhammad, "Enforcing position-based confidentiality with machine learning paradigm through mobile edge computing in real-time industrial informatics," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 7, pp. 4189–4196, 2019.

[8] K. Jacobsson, L. L. H. Andrew, Ao Tang, S. H. Low, and H. Hjalmarsson, "An improved link model for window flow control and its application to FAST TCP," *IEEE Transactions on Automatic Control*, vol. 54, no. 3, pp. 551–564, 2009.

[9] L. Tan, W. Zhang, and C. Yuan, "On parameter tuning for FAST TCP," *IEEE Communications Letters*, vol. 11, no. 5, pp. 458–460, 2007.

[10] K. Koo, J.-Y. Choi, and J. S. Lee, "Parameter conditions for global stability of FAST TCP," *IEEE Communications Letters*, vol. 12, no. 2, pp. 155–157, 2008.

[11] Y. J. Choi, J. W. Ko, S. W. Yun, and P. Park, "Improved global stability conditions of the tuning parameter in FAST TCP," *IEEE Communications Letters*, vol. 13, no. 3, pp. 202–204, 2009.

[12] X. M. Huang, C. Lin, and F. Y. Ren, "Recent development of high speed transport protocols," *CHINESE JOURNAL OF COM PUTERS*, vol. 29, no. 11, pp. 1901–1908, 2006.

[13] L. W. Hai-Tao and C. Ming, "Congestion control scheme based on network measurement and fuzzy logic control technology," *Journal of South China University of Technology (Natural Science Edition)*, vol. 34, no. 6, pp. 89–95, 2006.

[14] P. V. Prakash, S. Tripathi, R. Pal, and A. Prakash, "A slotted multichannel MAC protocol for fair resource allocation in VANET," *International Journal of Mobile Computing and Multimedia Communications*, vol. 9, no. 3, pp. 45–59, 2018.

[15] S. Samreen, "Packet dropping counter measures in a MANET through reliable routing protocol leveraging a trust management framework," *International Journal of Mobile Computing and Multimedia Communications*, vol. 9, no. 3, pp. 60–75, 2018.

[16] A. K. Sangaiah, D. V. Medhane, G. B. Bian, A. Ghoneim, M. Alrashoud, and M. S. Hossain, "Energy-aware green adversary model for cyberphysical security in industrial system," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 5, pp. 3322–3329, 2020.