TCPW BR: A Wireless Congestion Control Scheme Base on RTT

Liwei Tian[1, 2, *], Jinfeng Li[3], Longqing Zhang[2], Yu Sun[4] and Lei Yang[2]

Abstract: The wireless network is limited by the transmission medium, and the transmission process is subject to large interference and jitter. This jitter can cause sporadic loss and is mistaken for congestion by the congestion control mechanism. The TCP Westwood protocol (referred to as TCPW) is such that it cannot distinguish between congestion loss and wireless jitter loss, which makes the congestion mechanism too sensitive and reduces bandwidth utilization. Based on this, the TCPW protocol is modified based on the estimate of the Round-Trip Time (referred to as RTT) value-called TCPW BR. The algorithm uses the measured smooth RTT value and divides the congestion level according to the weighted average idea to determine congestion loss and wireless jitter loss. The simulation results show that the TCPW BR algorithm enhances the wireless network's ability to judge congestion and random errors.

Keywords: Congestion control, RTT, TCP Westwood.

1 Introduction

With the rapid development of computer networks, people's requirements for network resources are getting higher and higher. Especially in recent years, multimedia streams such as voice, image and video have emerged on the network, and the problem of network congestion [Kumar, Singh, Baghotia et al. (2013)] has become more and more serious. In general, the root cause of network congestion is that the load provided by the end system to the network is greater than the network resource capacity and processing capability, which is manifested by datagram delay, higher discard probability, and lower performance of upper-layer applications. In dealing with network congestion problems, two methods are generally adopted: one is to take measures to prevent network congestion, that is, to adopt flow control technology [Joseph and De Veciana (2011)]; the other is to take measures to reduce congestion at the time of congestion [Laflamme and Ossenbruggen (2017)], that is, to reduce congestion time and prevent congestion from spreading until the disappearance of congestion, that is, the use of congestion control technology. The task of congestion control is to ensure that the subnet can handle the

[1] College of Computer Science, Guangdong University of Science and Technology, Dongguan, 523083, China.

[2] College of Information Technology, Macau University of Science and Technology, Macau SAR.

[3] Department of Information Technology, Huizhou City College, Huizhou, 516025, China.

[4] College of Information Engineering, Guangdong Innovative Technical College, Dongguan, 523960, China.

[*] Corresponding Author: Liwei Tian. Email: 1809853gii30006@student.must.edu.mo.

CMC. doi:10.32604/cmc.2020.06135

traffic arriving by the bearer. This is a global problem involving all aspects of the behavior, including all hosts, routers and internal store-and-forward procedures.

In order to meet the development requirements of integrated networks, some congestion control protocols suitable for wireless networks and high-performance networks have been proposed at home and abroad, such as: NEW Reno [Floyd and Henderson (2012)], TCP Westwood [Bockenheimer, Fata and Possart (2008)] etc. among which TCP Westwood is more prominent in wireless congestion control, and comprehensive performance comparison in high performance networks Good is the H-TCP [Jin, Wang, Liu et al. (2009)] algorithm. Some better algorithms have emerged at the intermediate link nodes, such as Drop-Tail [Shorten, King and Wirth (2007)], RED [Feng, Huang, Xu et al. (2017)], BLUE [Wang, Zhang and Zhang (2010)], PI controller [Rivera-Vega, Ravi and Navathe (2010)], REM [Mise, Hasegawa , Satou et al. (2011)], etc. These algorithms are distributed over various levels in the transmission control, which greatly improves the congestion control environment of TCP and has great practical significance.

TCPW is an end-to-end congestion control mechanism protocol. TCPW eliminates the negative impact of random loss on network bandwidth utilization to a certain extent. It does not require the support of intermediate routers and fully adheres to the end-to-end TCP design principles. However, it is precisely because TCPW cannot distinguish between congestion loss and wireless loss, especially in wireless environments with high bit error rate and large delay. Because wireless network jitter and sporadic loss are often considered as network congestion, frequent network congestion processing is called. Mechanisms that reduce bandwidth utilization. Aiming at the shortcomings of TCPW that cannot distinguish between congestion loss and wireless self-loss, the TCPW BR algorithm is proposed. The improved RTT evaluation [Wang, Ren and Li (2014)] environment is used to estimate the RTT by using the timeout retransmission timer. Substituting the modified value of the weighted average mathematical expression, obtaining the congestion level parameter valued R, dividing the congestion level according to the value of R, making corresponding judgments on different reasons, improving the bandwidth utilization rate, maintaining good RTT fairness and TCP friendly.

2 TCPW scheme

TCPW adopts the idea of bandwidth estimation. The sender estimates the available bandwidth on the end-to-end link by observing the time interval of returning ACK. The ABSE (Adaptive Bandwidth Share Estimation) filtering mechanism makes its bandwidth estimation more accurate. The TCPW basic congestion window dynamic adjustment algorithm has not changed in the slow start and congestion avoidance phases. The basic idea is to use the bandwidth estimation BWE to adjust the congestion window and the slow start threshold when congestion occurs. The congestion control mechanism increases the AIAD (Additive Increase Adaptive Decrease): When the receiver detects three duplicate ACKs, Let Ssthresh=(BWE*RTT$_{min}$)/seg_size (instead of Reno's Ssthresh=Cwnd/2), if Cwnd>Ssthresh then Cwnd=Ssthresh; when RTO times out, let Ssthresh=(BWE*RTT$_{min}$)/seg_size (instead of Ssthresh=Cwnd/2), Cwnd=1 [GRIECO, A and MASCOLO (2005)]. Accurate estimation of bandwidth and threshold setting enable full utilization of network resources after retransmission, and TCPW largely eliminates the negative impact of random loss on network bandwidth utilization. It does not require the support of intermediate routers and fully adheres to the end-to-end TCP design principles. TCPW is very efficient in wired and wireless hybrid networks, with throughput up 550%.

The problem with TCPW is that it cannot distinguish between congestion and wireless packet loss, and research has found that it will overestimate the available bandwidth, which brings some unfairness. Below we use the network topology shown in Fig. 1 for experimental data analysis.

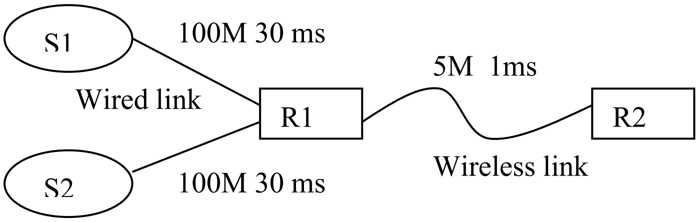


Figure 1: Network topology

The wired link bandwidth is 100 Mb/s, the time delay is 30 ms, the wireless link bandwidth is 5 Mb/s, the time delay is 1 ms, the packet size is set to 1000 bytes, and the link error rate is 1%, respectively. 2%, 3%, TCPW combines the following queue algorithms Drop-Tail, RED, BLUE, PI four, set WestTCP and Newreno two TCP data streams. When the BER of the wireless link is 1%, 2%, 3%, the average throughput and bandwidth utilization of the Westwood stream corresponding to the four queue algorithms are shown in Tab. 1.

Table 1: Bandwidth utilization at different bit error rates

Bit error rate	**Queue algorithm**			
	Drop-Tail	**RED**	**REM**	**PI**
0%	95.22%	90.94%	94.44%	95.32%
1%	70.44%	70.42%	72.72%	67.90%
2%	47.16%	44.10%	43.20%	42.04%
3%	34.24%	35.23%	32.08%	32.06%

It can be seen from the above analysis that TCPW combined with the common queue algorithm cannot distinguish between congestion loss and wireless packet loss when dealing with congestion. When the bit error rate is 0, the bandwidth utilization rate is above 90%, when the bit error rate is in the common 1% case, bandwidth utilization drops to around 70%. When the bit error rate reaches 3%, the bandwidth utilization is only 30% of the area, and a large amount of bandwidth is idle, resulting in waste of resources.

3 TCPW BR scheme

Based on this, we propose the TCPW BR scheme. The improvement of this algorithm is introduced below.

3.1 Congestion level division

The first step is to determine an accurate RTT estimation scheme. At present, there are many versions for detecting RTT values. This paper uses the method of timeout retransmission timer to predict the round trip delay [Leu, Jenq and Jiang (2011)]. SRTT + $(1-\beta)\times RTT_{NEW}\rightarrow\times SRTT$ where SRTT is a smooth RTT estimate and RTT_{NEW} represents the current RTT value (take $\beta=1/8$).

Substituting the measured RTT values into the following weighted average mathematical expressions (1) and (2) indirectly reflects network congestion. The specific practices are as follows:

$$F = RTT_{max}-RTT_{min} \tag{1}$$

$$R = (RTT-RTT_{min})/F \tag{2}$$

Here, F is the variation range of RTT, and RTT_{max} and RTT_{min} respectively represent the maximum and minimum values of the measured RTT during the transmission of the TCP data segment; RTT is the RTT value of the current time measured according to the current segment. $R\in[0,1]$ indicates the extent to which the currently confirmed data segment is used in the network transmission process. The smaller the R, the less time the data segment spends, and the network is idle; otherwise, the network is more congested. Divide R into 4 levels L, as shown in Tab. 2, where a higher level indicates a greater likelihood of congestion. The maximum value of the round trip delay is set to a value not greater than the timeout timer.

Table 2: Congestion level classification

R	[0,0.25]	(0.25,0.5]	(0.5,0.75]	(0.75,1]
L	1	2	3	4

3.2 TCPW BR algorithm

When the congestion level is equal to 1, it proves that the network condition is better and the congestion probability is small. If packet loss occurs at this time, it is considered that it is a large wireless packet loss, so there is no need to reduce the values of Cwnd and Ssthresh excessively. When the congestion level is 2, it is proved that there is slight congestion, and the transmission rate can be appropriately changed to reduce the value of the growth factor *P*. When the congestion level is greater than 3, the congestion is proved to be serious. The packet loss is considered to be congestion and packet loss. Call TCPW congestion control mechanism (p=0.5 or p=0.4), see Tab. 3.

Table 3: Growth factors corresponding to congestion levels

L	1	2	3	4
P	maintain	0.867	0.5	0.4

The algorithm is described as follows:

```
(1) Each time an ACK of a new data segment is received,
   If (congestion level=1||congestion level=2)//Think it is wireless packet loss, mild congestion
      Cwnd=Cwnd+1;
   If (Cwnd>Ssthresh)
      Cwnd=Cwnd+(1/Cwnd)*p;
(2) After receiving a duplicate ACK before timing out
   If (duplicate ACK=3&& congestion level=1)
      Fast retransmission;
      Quick recovery
   If (duplicate ACK=2&& (congestion level=3||congestion level=4))//Think it is a congestion packet
   Slow start or congestion avoidance;
     Cwnd=Cwnd*p;
     Ssthresh =(BWE*RTTmin)/seg_size;
   If (Cwnd>Ssthresh) then Cwnd=Ssthresh;
   If (duplicate ACK=3&& congestion level>2)
   Slow start or congestion avoidance;
     Cwnd=Cwnd*p;
     Ssthresh=(BWE*RTTmin)/seg_size;
   If (Cwnd>Ssthresh) then Cwnd=Ssthresh;
```

4 Simulation and analysis

4.1 Throughput

The TCP versions New Reno, TCPW and TCPW BR were simulated on the wireless link using the NS-2 [Weigle, Adurthi, Jeffay et al. (2006)] simulation platform, and their changes were recorded. Fig. 2 shows the network topology. The network-related parameters are: the bandwidth of the wired link is 100 Mb/s, and the one-way transmission time is 30 ms. The bottleneck link bandwidth is 100 Mb/s, the unidirectional transmission time is 10 ms. The wireless link bandwidth is 5 Mb/s, the unidirectional transmission time is 0.01 ms, and the transmission packet size is 1000 bytes.

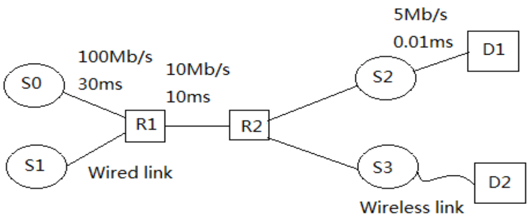
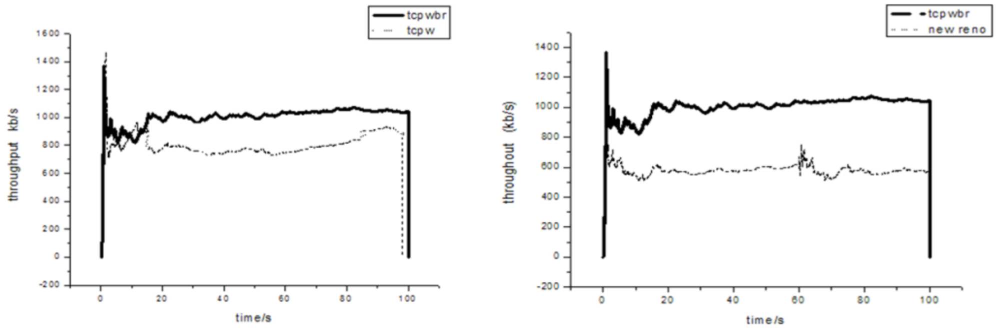


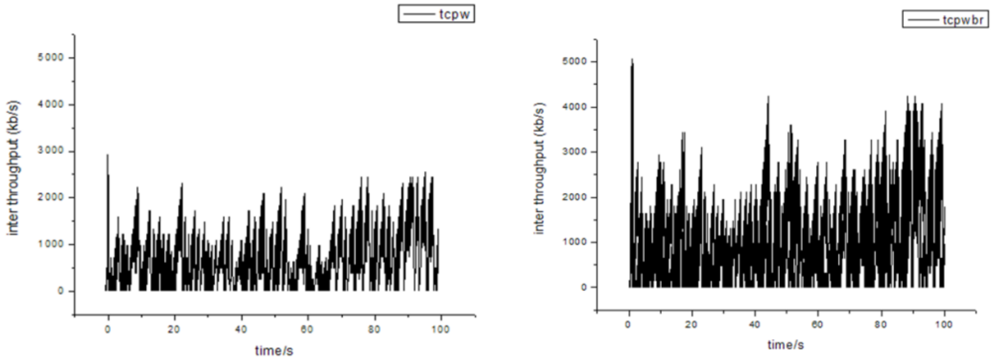
Figure 2: Network topology

During the simulation, the wireless link error rate was set as 3%. It is easy to see from Fig. 3 that the average throughput of TCPW BR is nearly 20% higher than TCPW, almost twice that of newreno. This is because TCP BR effectively distinguishes between wireless packet loss and congestion packet loss, avoids frequently calling congestion mechanism, and improves bandwidth utilization and throughput. Fig. 4 shows the interval throughput of the TCPW and TCPW BR algorithms. Obviously, TCPW BR is also much better than TCPW in terms of interval throughput, with higher bandwidth utilization and more stable throughput. Fig. 5 shows the bandwidth utilization of TCPW BR and Newreno at different link error rates.



(a) tcpw compared to tcpw br (b) tcpw br compared to newreno

Figure 3: Comparison average throughput when the link error rate is 3%



(a) interval throughput of tcpw (b) interval throughput of tcpw br

Figure 4: Comparison interval throughput when the link error rate is 3%

The algorithm analyzes the performance of the channel delay, setting the wireless link bandwidth as 5M and the link error rate as 1%. It is easy to get from Fig. 6. When the time delay changes from 10 ms to 200 ms, TCPW BR is slower than TCPW and Newreno. It is worth noting that when the channel delay is 100 ms, the throughput of TCPW BR reaches 950 Kbps, which is 2.2 times that of Newreno. As link latency increases, the benefits of improved algorithms become more apparent.

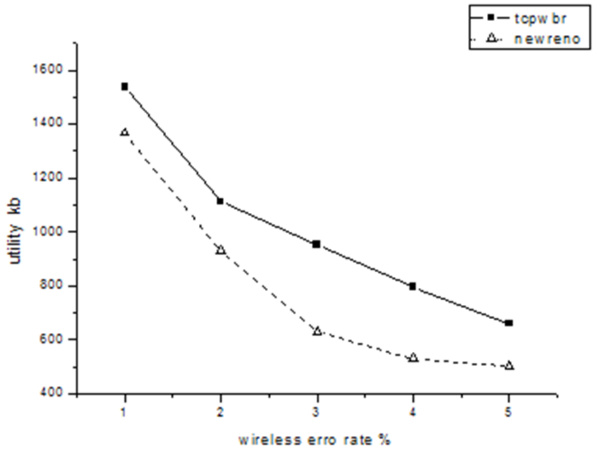


Figure 5: Bandwidth utilization of TCPW BR and Newreno at different bit error rates

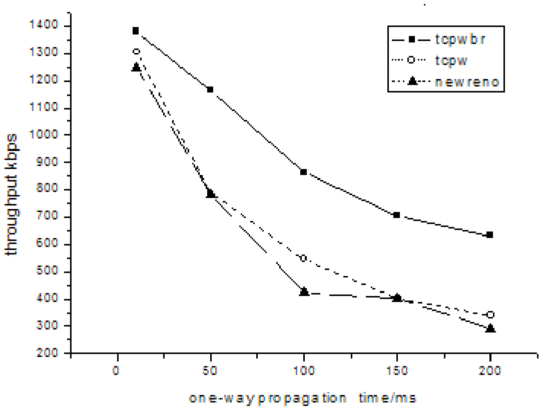


Figure 6: Newreno, TCPW, and TCPW BR bottleneck link throughput and one-way transmission time

Table 4: Comparison of packet loss ratios when the radio link error rate is 1%

TCP version	Newreno	TCPW	TCPW BR
Number of packets sent	7797	13305	13663
Number of lost packets	89	103	77
Packet loss rate	1.1415%	0.7741%	0.5636%

The simulation experiment also made a statistic on the packet loss rate corresponding to the three different algorithms. Since the TCPW BR is more flexible for the event handling mechanism, the packet loss rate is improved. As can be seen from Tab. 4, TCPW BR is 0.2 percentage points higher than TCPW when the number of transmitted packets is almost the same.

4.2 RTT fairness

For streams with different RTTs, the final response of the stream is quite different because it increases the window and reduces the period of the window. We use W_i and RTT_i to represent the average window and RTT of each RTT of the stream i, and use t to indicate the time interval of two consecutive packet drops after reaching steady state, and let pi be the packet loss rate, then the time sent in time t is known. The total window is 1/pi, so the average window sent by each RTT is:

$$w_i = \frac{1/p_i}{t/RTT_i} = \frac{RTT_i}{t \times p_i} \tag{3}$$

We know that for the ACMD type of partial congestion control algorithm, its response function can be written as follows:

$$R(p) = \frac{1}{RTT}\frac{c}{p^d} \tag{4}$$

where c and d are constants, so there is

$$\frac{w_i}{RTT_i} = \frac{1}{RTT}\frac{c}{p^d} \Rightarrow p_i = \sqrt[d]{c/w_i} \tag{5}$$

Substituting the above formula into (3), you can get:

$$w_i = \frac{RTT_i}{t \times \sqrt[d]{c/w_i}} \Rightarrow w_i = (\frac{RTT_i}{t \times \sqrt[d]{c}})^{\frac{d}{d-1}} \tag{6}$$

So we can figure out the ratio of the windows sent by the two streams in each RTT time:

$$\frac{(w_1/RTT_1)(1-p_1)}{(w_2/RTT_2)(1-p_2)} \approx \frac{w_1/RTT_1}{w_2/RTT_2} = (\frac{RTT_2}{RTT_1})^{\frac{1}{1-d}} \tag{7}$$

From the above Eqs. (3)-(7), we know that the RTT fairness [Abrantes, Araujo and Ricardo (2010)] of a protocol depends on the value of d in the response function of theprotocol. When d increases, RTT fairness will deteriorate.

We use the Fairness Index introduced to measure the fairness of the TCPW BR algorithm. Set the bottleneck bandwidth to 5M and the link-error-ratio to 1%. Set 3 senders and 3 receivers, send data at the same time, count the throughput of each stream, and substitute the formula $F(x) = [\Sigma x_i]^2 / n[\Sigma x_i{}^2]$, where x_1 , x_2 , ······ , x_n to represent the throughput of each competing data stream, when F is close to 1 When the best condition is reached. The obtained data is shown in Fig. 7. It is easy to see that the TCPW BR fairness index has been around 0.9 under different transmission delays, which has good RTT fairness.

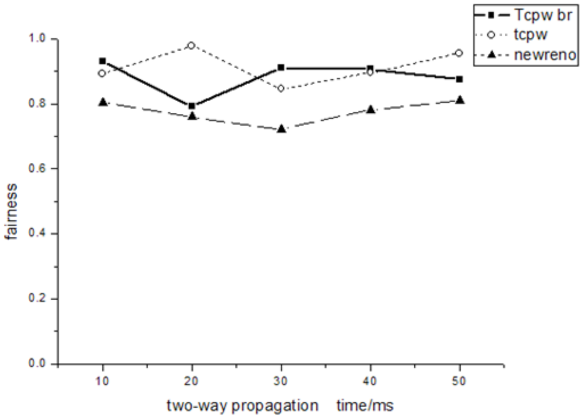


Figure 7: Fairness coefficients under different transmission delays

4.3 TCP-friendliness

The network is a very complex system. Although 95% of the current data streams use the TCP protocol, the real-time flow with non-responsive characteristics emerging in the network has a great impact on the traditional TCP service flow. The TCP stream will reduce the transmission rate when it is congested. The non-TCP stream will ignore the existence of congestion and continue to maintain the original transmission rate, resulting in the non-TCP stream acquiring more bandwidth. The TCP stream following the congestion control mechanism is getting more and more re- sources. Less, seriously impairing the performance of the TCP stream. Therefore, the congestion control algorithm needs to consider the fairness problem when non-TCP streams and TCP streams compete for resources, that is, TCP-friendliness [Aydin, Iyengar, Conrad et al. (2012)]. Fr can be measured by the TCP friendliness factor. The smaller Fr, the better the TCP friendliness of non-TCP flows.

Fr =link utilization of non-TCP flows/link utilization of TCP flows

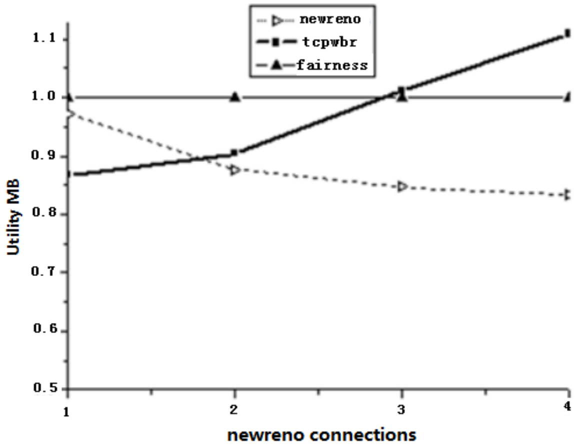


Figure 8: TCPW BR and Newreno’s friendliness

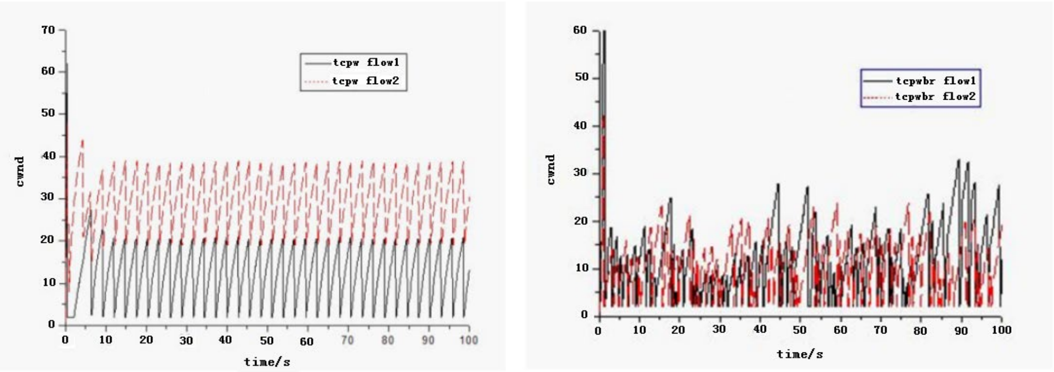
In the friendliness test part, we used 5 different TCP streams, set 5 senders and 5 receivers, and bind TCPW BR algorithm and Newreno algorithm respectively. During the experiment, the TCPW BR bound stream was increased from 1 to 4, and the Newreno bound stream was reduced from 4 to 1. During the simulation, it is guaranteed that 5 TCP streams compete for 5M bandwidth capacity. Set the link-error-ratio as 1%. As can be seen from Fig. 8, TCPW BR is friendly to Newreno.

4.4 TCP convergence

There are many factors affecting stability, such as topology, RTT, bandwidth, number of streams, etc. Good congestion control algorithms should maintain the stability of the network and avoid violent oscillations. There is no uniform standard for the stability of a system. If network delay is ignored, the global stability of using different control rules can be analyzed and determined in a general network topology. But when feedback delays are included in this problem, the analysis becomes complicated. Therefore, it is best to construct global, non-linear control rules through linearized analysis methods [Zhou, Zhang, Lin et al. (2016)].

We experimented with two TCPW streams and two TCPW BR streams, and recorded the size of the congestion window (Cwnd) during the experiment. The experimental parameters are configured as follows: the bottleneck link is 10M, the bit-error-ratio is 1%, and the simulation time is 100 s.

It is easy to see from Fig. 9 that the congestion window of the two data streams of TCPW appears synchronous and the convergence is very poor. The TCPW BR stream is more convergent than the TCPW stream.



(a)TCPW (b)TCPW BR

Figure 9: Comparison convergence when the bottleneck link is 10 M

5 Summary

In this paper, the timeout retransmission timer is used to smooth the RTT value, and the current congestion level is averaged according to the RTT change value to distinguish between congestion loss and wireless loss. The innovation of this algorithm is to use the method of timeout retransmission timer to calculate the smooth RTT value and the application of the weighted average formula. In the actual algorithm, other RTT estimation methods can also be tried to carry out the experiment. The difference in RTT valuation methods may have a certain impact on the classification of congestion levels. However, no matter which valuation method is adopted, as long as the bandwidth load is not significantly increased, combined with the original control mechanism, certain improvement effects will be achieved. Through the simulation experiment in this chapter, TCPW BR has a greater improvement in throughput and bandwidth utilization than TCPW, and maintains good fairness and friendliness.

Acknowledgement: The paper is funded by Guangdong Provincial Department of Education major scientific innovation project characteristics (natural sciences), project number: 2014KTSCX210. 2016 Guangdong University of Science and Technology Key Project: Smart Agricultural Greenhouse System Based on Internet of Things and Plant Growth Model.

References

Abrantes, F.; Araujo, J. T.; Ricardo, M. (2010): Explicit congestion control algorithms for time varying capacity media. *IEEE Transactions on Mobile Computing*, vol. 10, no. 1, pp. 81-93.

Aydin, I.; Iyengar, J.; Conrad, P.; Shen, C. C.; Amer, P. (2012): Evaluating tcp-friendliness in light of concurrent multipath transfer. *Computer Networks*, vol. 56, no. 7, pp. 1876-1892.

Bockenheimer, C.; Fata, D.; Possart, W. (2008): Analysis of TCP Westwood+ in high speed networks. *American Journal of Industrial Medicine*, vol. 31, no. 3, pp. 366-367.

Feng, C. W.; Huang, L. F.; Xu, C.; Chang, Y. C. (2017): Congestion control scheme performance analysis based on nonlinear red. *IEEE Systems Journal*, vol. 11, no. 4, PP. 2247-2254.

Floyd, S.; Henderson, T. (2012): The newreno modification to TCP's fast recovery algorithm. *RFC*, vol. 345, no. 2, pp. 414-418.

Grieco, L.; A.; Mascolo, S. (2005): Mathematical analysis of Westwood+TCP congestion control. *IEEE Proceedings-Control Theory and Applications*, vol. 152, no. 1, pp. 35-42.

Jin, Y.; Wang, J.; Liu, Q.; Luo, Y. (2009): An improved TCP with cross-layer congestion notification over wired/wireless hybrid networks. *Chinese High Technology Letters*, vol. 19, no. 4, pp. 368-373.

Joseph, V.; De Veciana, G. (2011): Stochastic networks with multipath flow control: impact of resource pools on flow-level performance and network congestion. *Acm Sigmetrics Performance Evaluation Review*, vol. 39, no. 1, pp. 61-72.

Kumar, D.; Singh, J.; Baghotia, A.; Kumar, S. (2013): Network congestion/resource allocation game. *Dissertations & Theses-Gradworks*, vol. 109, no. 2, pp. 3126-3127.

Laflamme, E. M.; Ossenbruggen, P. J. (2017): Effect of time-of-day and day-of-the-week on congestion duration and breakdown: a case study at a bottleneck in Salem, NH. *Journal of Traffic & Transportation Engineering*, vol. 4, no. 1, pp. 31-40.

Leu, F. Y.; Jenq, F. L.; Jiang, F. C. (2011): A path switching scheme for sctp based on round trip delays. *Computers & Mathematics with Applications*, vol. 62, no. 9, pp. 3504-3523.

Mise, Y.; Hasegawa, K.; Satou, S.; Aoki, T.; Beck, Y. et al. (2011): Venous reconstruction based on virtual liver resection to avoid congestion in the liver remnant. *British Journal of Surgery*, vol. 98, no. 12, pp. 1742-1751.

Rivera-Vega, P. I.; Ravi, V.; Navathe, S. B. (2010): Scheduling file transfers in fully connected networks. *Networks*, vol. 22, no. 6, pp. 563-588.

Shorten, R.; King, C.; Wirth, F.; Leith, D. (2007): Modelling TCP congestion control dynamics in drop-tail environments. *Automatica*, vol. 43, no. 3, pp. 441-449.

Wang, G.; Ren, Y.; Li, J. (2014): An effective approach to alleviating the challenges of transmission control protocol. *Iet Communications*, vol. 8, no. 6, pp. 860-869.

Wang, Y.; Zhang, T.; Zhang, J. (2010): Improved blue algorithm for fast stabilizing queue. *Computer Engineering*, vol. 36, no. 22, pp. 119-121.

Weigle, M. C.; Adurthi, P.; Jeffay, K.; Smith, F. D. (2006): Tmix: a tool for generating realistic tcp application workloads in NS-2. *ACM Sigcomm Computer Communication Review*, vol. 36, no. 3, pp. 65-76.

Zhou, B.; Zhang, F.; Lin, W.; Hou, C.; Anta, A. F. et al. (2016): Hdeer: a distributed routing scheme for energy-efficient networking. *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 5, pp. 1713-1727.