An Improved TCP Congestion Control Algorithm of Based on Bandwidth Estimation in Heterogeneous Networks

Rui-Qing Wu, Hong Jie, and Nan Ding
University of Electronic Science and Technology of China, Chengdu 611731, China
Email: rqwu@uestc.edu.cn; skybird9003@gmail.com; dingnan11@163.com

Abstract —Due to the difficulty in discriminating the wireless loss from congestion loss in heterogeneous networks, TCP Westwood can not accurately estimate the available bandwidth, and cause low network utilization. In this paper, we propose an improved congestion control algorithm called TCP NewBR, which adaptively estimates bandwidth by using the bottleneck link utilization, and selects more precise time interval of ACK to improve the accuracy of available bandwidth estimation. In addition, we also modify the faster recovery and faster retransmission algorithms according to the queue length of bottleneck link. Simulation results in several scenarios show that TCP NewBR can get more accurate bandwidth estimation, and more significant throughput than traditional TCP Westwood, and fairer and friendlier than TCP Reno.

Index Terms—Heterogeneous networks, bandwidth estimation, throughput, TCP Westwood

I. Introduction

The traditional TCP was designed for the wired network, where the congestion accounts for most packet losses. The characteristics of bit error rate (BER) in heterogeneous networks often lead to a lot of packet losses and even the connection is interrupted. Because of the assumption that any packet loss is due to network congestion without the capability to distinguish wireless loss from congestion loss, the traditional TCP blindly decreases the congestion window (cwnd). The undesirable results include low bandwidth utilization, long latency, and low throughput. In the heterogeneous networks with the high BER, TCP performance deteriorates, narrow bandwidth, long delay and frequent handover or moving. Taking TCP Reno as an example, when one timeout or three duplicate acknowledgments occurs, TCP Reno detects a packet loss, and implements a multiplicative decrease algorithm, not taking into account the network status. In the past decades, many congestion control algorithms have been proposed to improve TCP performance in wired and wireless networks. These improved algorithms respectively focus on the end-to-end [1], cross-layer design [2], split-connection technique [3] and snoop mechanism [4]. The end-to-end improvement schemes, such as TCP NewReno, TCP SACK, TCP Vegas [5], and TCP Westwood [6], are primarily revised from the traditional TCP protocol, and don't depend on any intermediate network node, and take the advantages of simple implementation and end-to-end significance. Among these protocols variants TCP Westwood (TCPW) gives better performance in wireless network. TCPW uses the estimated bandwidth to properly set the cwnd and slow-start threshold (ssthresh), which improve the throughput and utilization. However, TCPW can not acquire the stable and accurate available bandwidth estimation in the networks with the asymmetric link or reverse flows existing in return path. Furthermore, TCPW is still unable to distinguish random loss from congestion loss which results in an inaccurate change of the cwnd and ssthresh, finally degrades performance. Researchers have proposed many improved schemes [7]-[10]. Among, Shimaa Hagag [8] proposes TCP Westwood New which enhances TCPW congestion avoidance mechanism based on the bandwidth ratio compared with a fixed value. But the fixed ratio can not correctly reflect the network status, resulting large fluctuations. Geethu Wilson [9] modifies bandwidth estimation using modified exponentially weighted moving average. Although the number of bandwidth samples increases, the result of bandwidth estimation is still inaccurate.

In view of the above problems, this paper proposes an improved algorithm called TCP NewBR, which not only takes the advantages of TCP BE and TCP RE [11], but also uses the utilization of bottleneck link to adjust bandwidth estimation in real-time so as to obtain better accuracy of bandwidth, and judges the degree of network congestion by buffer queue size during faster recovery and faster retransmission phase to acquire more significant throughput, fairness and friendliness.

II. TCP BE/RE Algorithm

The key idea of TCPW bandwidth estimate (BE) is to exploit the TCP acknowledgment to estimate the available bandwidth of a TCP connection. The bandwidth is estimated by measuring the rate of returned ACKs, i.e., the size of packets confirmed by the ACK is divided by the arrival time interval of confirmation at the sender side. The estimated bandwidth indicates the occupied bandwidth of data flow in the network, namely the available bandwidth. When three duplicate acknowledgements are received or one timeout occurs at the sender side, indicating a packet loss due to the

Manuscript received June 20, 2014; revised October 20, 2014.
This work was supported by the Fundamental Research Funds for the Central Universities Project No. ZYGX2012J020.
Corresponding author email: rqwu@uestc.edu.cn.
doi:10.12720/jcm.9.10.792-797

network congestion or random error, the estimation algorithm is then used to compute the new ssthresh and cwnd. Compared with the traditional TCP variants which blindly reduce the cwnd by half, the recovery mechanism of BE can get more accurate cwnd and higher bandwidth utilization in wireless network with high BER. However, since TCP traffic has a burst characteristic, BE is regarded as an instantaneous available bandwidth estimation of the bottleneck link, which is apt to cause an overestimate. In order to avoid BE overestimating, S. Mascolo proposed to extend the sample interval to RTT [12], considering the amount of data confirmed during the latest interval of RTT as a sample to obtain the sample bandwidth. The estimated bandwidth is actually the average sending rate in the time *T*, so this method is also known as the rate estimation algorithm (RE). RE can acquire the good performance under the condition of packet loss due to congestion, but RE underestimate bandwidth in the case of random loss, and results in poor link utilization and low throughput.

III. TCP NEWBR

A. Enhanced Bandwidth Estimation

A lot of works show that BE is more effective in networks with random error. On the other hand, RE method appears to be more appropriate when packets are lost due to congestion. Therefore the BE and RE algorithms have some limitations on the application. TCP NewBR takes the advantages of these two algorithms, and regards the bottleneck link utilization as a weighted linear factor to estimate available bandwidth. It can not only accurately estimate the available bandwidth in the case of non occurrence of packet loss, but also distinguish random loss from congestion loss. In other words, TCP NewBR can intelligently adjust the bandwidth estimation to improve the accuracy of estimation based on bottleneck link utilization. In addition, since the asymmetric link and the reverse flow affect the accurate estimation of available bandwidth, TCP NewBR extracts the time stamps from ACKs which reflect the received time of the corresponding packet confirmed by the ACK at the receiver. Because the received time is different from the arrival time of return ACK at the sender side, it is independent of the return path. TCP NewBR is to effectively avoid an overestimation of bandwidth as to ACK compression. Benefiting from the above two improvements, TCP NewBR can acquire more accurate bandwidth estimation and better throughput than BE and RE.

The underutilization denoted by u indicates the link surplus index. rtt is the current round-trip delay and rtt_{min} is set equal to the overall minimum round-trip delay measured over the duration of the connection (based on continuous monitoring of ACK rtts). rtt_{max} represents the maximum value of rtt using an exponential averaging of rtt measurements. The expression of rtt_{max} is as follows:

$$rtt_{\max} = \beta rtt_{(n\text{-}1)\max} + \frac{1-\beta}{2}(rtt_n + rtt_{n-1}) \tag{1}$$

where β ,$rtt_{(n\text{-}1)\max}$, rtt_n, and $rtt_{n\text{-}1}$ are the exponential smoothing factor, the (n-1)th rtt_{max}, the nth rtt and the (n-1)th rtt, respectively. We take $\beta = 19/21$.

We assume that when rtt is equal to rtt_{min}, bottleneck link is underutilized, and u is set to 1. In this case eligible rate estimation (ERE) is equal to BE. With the increment of rtt, u is exponentially decreased and the exponentially reduced bandwidth estimation is close to RE. Bandwidth estimation expression is as follows:

$$ERE = uBE + (1-u)RE \tag{2}$$

$$u = \exp\left(-\frac{d}{d_{\max}}\partial\right) \tag{3}$$

$$d = rtt - rtt_{\min} \tag{4}$$

$$d_{\max} = rtt_{\max} - rtt_{\min} \tag{5}$$

where d denotes the current queuing delay and $d_{\max}$ represents the max queuing delay. The ∂ is a constant largely enough to ensure that u approaches to 0 when rtt is close to $rtt_{\max}$, generally, ∂ is equal to 10.

B. Modified Faster Retransmission and Faster Recovery

Because of the increasing sending rate from slow-start stage to congestion avoidance phase of TCP congestion control, a congested router is the likely reason for packet loss. On the other hand, in the wireless portion, a noisy, fading radio channel is the more likely cause of packet loss. It is necessary to identify the predominant cause of packet loss, and appropriately set the ssthresh and cwnd to take full advantages of bandwidth resource. In this paper, we improve faster recovery and faster retransmission mechanism by using the queue length based on the estimated available bandwidth to choose more reasonable ways to set cwnd and sshtresh.

In the work [13], the approach based on the balance point of the window to set the cwnd is proposed, we refer to the principle of estimating the queue length of bottleneck link. Then the estimated queue length is used to determine the degree of congestion, and set the ssthresh and cwnd.

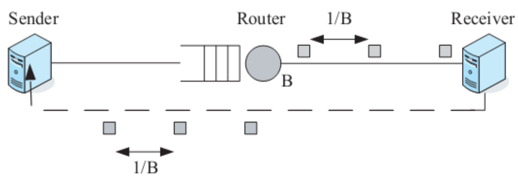


Fig. 1. Simulating the backward and forward path

Referring to the method in the literature [13], we can get the current round-trip delay:

$$rtt = T_P + \frac{1}{B} + \frac{q}{B} \tag{6}$$

where T_P is the propagation delay, q is the queue length, B is the bandwidth of the router and $1/B$ is the processing delay. The current round-trip delay consists of the three parts: propagation delay, processing delay and queuing delay. Hence, the queuing delay is:

$$rtt - rtt_{\min} = (T_p + \frac{1}{B} + \frac{q}{B}) - (T_p + \frac{1}{B}) = \frac{q}{B} \quad (7)$$

The queue length can be obtained by the expression (8):

$$q = B(rtt - rtt_{\min}) \quad (8)$$

Similar to TCP Vegas, which adjusts the size of the cwnd according to the difference between the expected rate and the actual rate, but this method is differed from the estimation of bottleneck queue length. TCP Vegas regards the difference as the queue length of bottleneck link. However, our estimation approach can better reflect the entire-network state, and get more accurate queue length by combining available bandwidth estimation and the round-trip delay difference to estimate the length of the buffer queue, rather than TCP Vegas by using cwnd to compute the expectations rate and the actually sending ratio. When the sender receives three duplicate ACKs, it indicates that one or more packets losses. We distinguish the reason of packet loss due to random error or the capacity of the network according to the bottleneck link queue length (qlen). If the qlen is less than ξ, the packet is loss due to random error. Accordingly, the cwnd should be reduced properly to avoid the waste of bandwidth. Conversely it indicates that network packet loss is as a reminder of the tendency to saturation, and sending rate should be adjusted to an appropriate value to ensure the full use of network bandwidth. When the retransmission timer expires, the processing method is similar to above. Finally, the improved faster retransmission and faster recovery mechanisms are as follows:

if (received three duplicate ACKs)
if (bottleneck queue length<ξ) {
$ssthresh = rtt_{avg} bw / seg\_size$
$cwnd = ssthresh$
}else if(bottleneck queue length >ξ) {
$ssthresh = rtt_{\min} bw / seg\_size$
$cwnd = ssthresh$
}
if (timeout)
if (bottleneck queue length <ξ) {
$ssthresh = rtt_{avg} bw / seg\_size$
$cwnd = 1$
}else {
$ssthresh = rtt_{\min} bw / seg\_size$
$cwnd = 1$
}

where *bw* denotes the estimated bandwidth, *seg_size* is the size of the packet, and rtt_{avg} represents the average value of *rtt* during the connection. A number of tests prove that the network has a better performance when ξ is 3.

In addition, since the performance of TCP significantly decreases under the network environment of multiple packet losses in the same window, we adopt the partial ACK to detect lost packets in the same window. In faster recovery phase, the sender also delivers a segment, and look forward to a cumulative ACK which confirms the sender had sent all the packets. After a RTT, if the sender receives the partial ACK, continuing to retransmit the next packet, rather than quitting the stage of faster retransmission. The improved faster retransmission and faster recovery mechanism can adjust the sending rate of network based on the network states, and improve the network performance.

IV. Simulation and Evaluation

In this section, we use NS-2 to simulate and evaluate the performance of the improved algorithm compared with BE and RE algorithms. The performance of improved algorithm focuses on the following: (i) the accuracy of available bandwidth estimation; (ii) algorithm throughput; (iii) fairness and friendliness. A mixed network topology is shown in Fig. 2. W0~W3 are four nodes, R0 and R1 are two routers, link bandwidth and one-way delay are shown in Fig. 2. MH is the mobile node connected to the wired network via the station BS. The buffer size is equal to the pipe size, and the packet size of TCP is set to 1400 bytes.

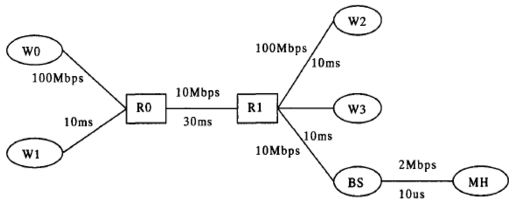


Fig. 2. Network topology

A. Accuracy of Bandwidth Estimation

It is essential to TCP congestion control algorithm based on bandwidth estimation to get an accurately available bandwidth. Three scenarios including competitive TCP flow, ACK compression, and non-adaptive UDP flow are set up to test the accuracy of bandwidth estimation of TCP NewBR compared with BE and RE. Simulation results show that TCP NewBR obtains higher degree of accuracy in bandwidth than BE and RE.

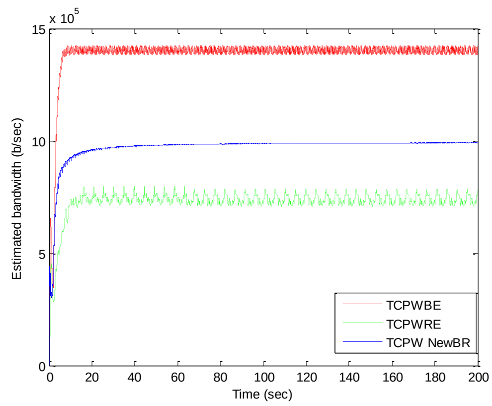


Fig. 3. TCPW with competitive flow

Fig. 3 illustrates the results for a link shared by two TCP NewBR (BE or RE) connections. The main parameters of network configuration are previously described in Fig. 2. Link bandwidth depends on the bottleneck link bandwidth, so bandwidth of each TCP connection is about 1 Mbps. As shown in Fig. 3, BE is estimated around 1.4Mbps, far higher than its 1 Mbps fair share. RE is estimated at 0.75Mbps fluctuation which is lower than 1 Mbps fair share, whereas the bandwidth estimation of TCP NewBR is in accordance with the fair share value.

The next simulation experiments include two TCP flows from W0 to MH, and a reverse flow from MH to W1 at 40s to 60s. From Fig. 4, we can see BE is rising rapidly in the period of occurrence of ACK compression, more than 6 times higher than fair share bandwidth, while TCP NewBR is almost unaffected. Simulation results illustrates TCP NewBR can accurately estimate the available bandwidth in the case of ACK compression to avoid BE overestimation. Fig. 5 illustrates the results for a link shared by one TCP NewBR and two non-adaptive UDP flows with a fixed rate of 2Mbps. The duration of the first UDP connection is from 30s to 60s and the other one is from 80s to 100s. We can see from Fig.5, when UDP exists, TCP NewBR can accurately estimate the available bandwidth of the network, and responses quickly to UDP flow.

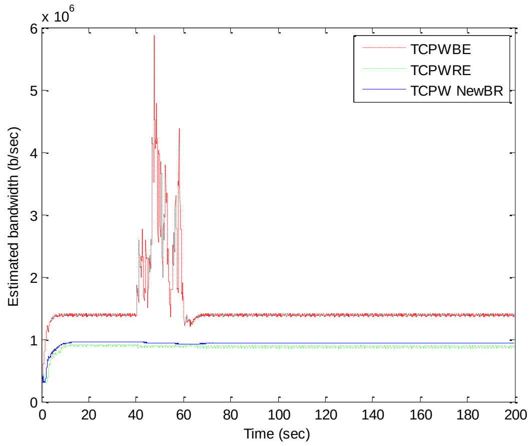


Fig. 4. TCPW with reverse flow

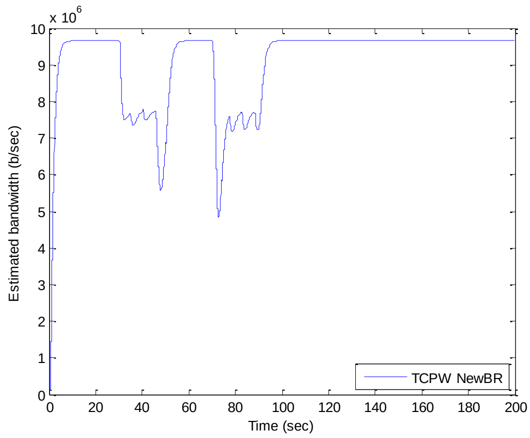


Fig. 5. TCPW NewBR with non-adaptive UDP flow

B. Throughputs with Loss Link

The main purpose of the improved algorithm is to enhance TCP throughput performance in the heterogeneous network. We evaluate the throughput of TCP NewBR compared with BE, RE or Reno. The configuration of network topology is shown in Fig. 6. The wired link has a capacity of 10Mbps and one-way propagation time of 45 ms. The wireless link is at 2Mbps point-to-point link connecting a mobile host to the base station.

We compare the performance of TCP NewBR with BE, RE and Reno under various error rates, bottleneck capacities and round trip propagation times. The simulation results are respectively shown in Fig. 7, Fig. 8, and Fig. 9.

In Fig. 7, we set up the BER of the wireless link from 0.0001 to 0.05, and TCP connections are established from the wired S to the wireless mobile node D. Simulation results show that the congestion control algorithm based on bandwidth estimation such as RE, BE, TCP NewBR is generally higher than the traditional TCP. The reason is that these algorithms consider in real time the current link status based on the link bandwidth estimation, and set dynamically the cwnd and ssthresh, rather than drop blindly sending rate, namely to keep adaptively the sending rate. In addition, the reason why TCP NewBR acquires a higher throughput performance with respect to RE or BE is that the accuracy of available bandwidth estimation and the process of distinguishing the main reason of packet loss can obtain more reasonable settings of the ssthresh and cwnd to improve the link utilization.

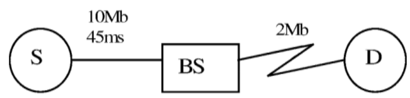


Fig. 6. One hop network topology

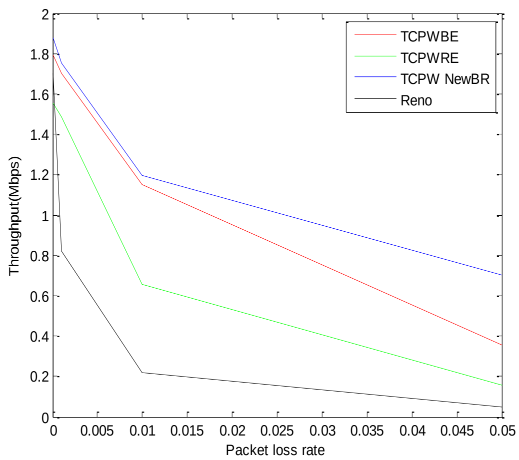


Fig. 7. Throughput versus packet error rate

In Fig. 8, the capacity of the wireless link varies from 1Mbps to 8Mbps with the error rate fixed at 0.005. Simulation results show that TCP NewBR can better utilize link capacity as the bottleneck link transmission rate increases, in other words, the link utilization is improved. RE underestimates bandwidth due to too long sample interval, resulting in low throughput. Besides TCP overestimating bandwidth leads to extra packets losses. While TCP NewBR obtains more significant throughput than both by real-time adjusting bandwidth estimation based on link utilization factor. In Fig. 9, the bottleneck propagation time varies from 0.01ms to 250ms with the error rate at 0.005. Simulation results show that TCP NewBR decays more slowly than other algorithms. At the same propagation time, throughput of TCP NewBR is significantly higher than other protocols, especially from 50ms to 100ms, 3-6 times higher throughput than the Reno.

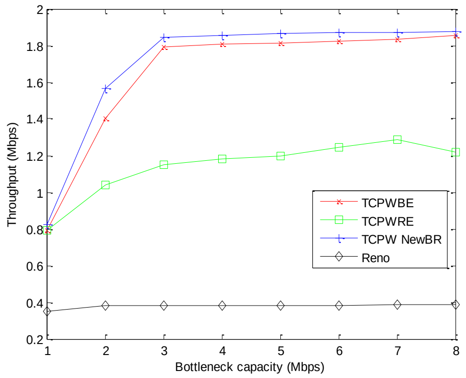


Fig. 8. Throughput versus bottleneck capacity

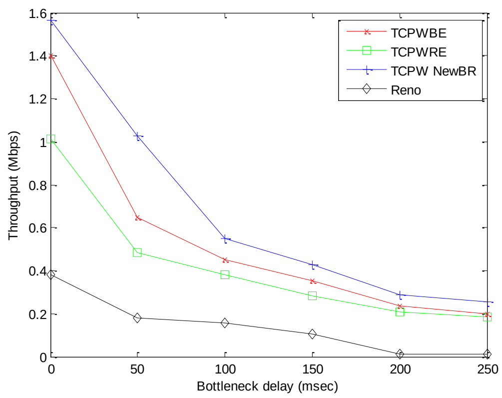


Fig. 9. Throughput versus bottleneck delay

C. Fairness and Friendliness

We use Fairness Index to measure the fairness of TCP NewBR algorithm, which is defined as $F(x) = (\sum X_i)^2 / (n \sum X_i^2)$, where n denotes the number of flow, X_i represents the average throughput of the ith connection. The closer to 1 F(x) is, the better fairness is. In Fig. 10, we set up 2~10 TCP flows sharing 10Mbps bottleneck bandwidth with the error rate fixed at 0.005 to compute fairness index. Simulation results show that fairness index of TCP NewBR is above 0.9, and higher than Reno. Finally, we simulate friendliness by setting up a total of 10 TCP flow (the number of TCP NewBR or BE decreased from 9 to 1, while the number of the corresponding Reno increased from 1 to 9) shared the bottleneck link. Simulation results show in Fig. 11 and Fig. 12 that TCP NewBR is friendlier to Reno than BE.

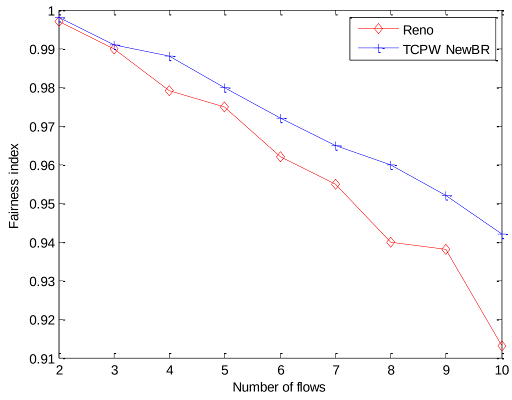


Fig. 10. Fairness of Reno versus NewBR

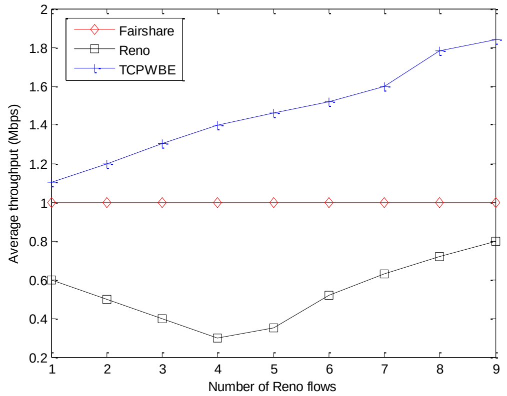


Fig. 11. Friendliness of BE

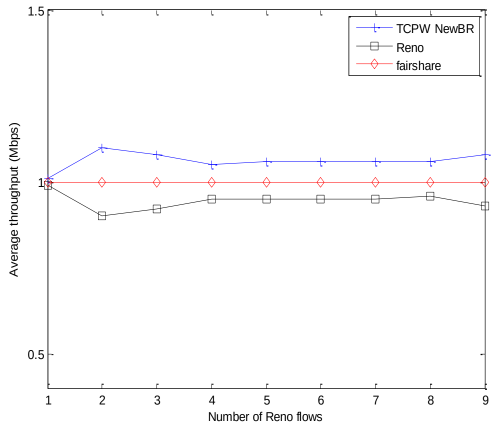


Fig. 12. Friendliness of NewBR

V. Conclusion

Congestions control algorithm based on the available bandwidth can avoid the influence of random error, and improves the performance of TCP stream in a wireless network. However, the bandwidth estimation of TCPW is not enough accurate in complex network environment [14]. So it degrades the performance in some extent. In this paper, we adaptively estimate bandwidth based on the bottleneck link utilization and use the estimated size of buffer queue to distinguish the congestion degree. The above simulation results show that TCP NewBR can accurately estimate the available network bandwidth with respect to the traditional TCPW and improve bandwidth utilization and throughput. Especially in the heterogeneous network environment with high error rate, TCP NewBR can obtain higher throughput than BE, RE and traditional TCP protocol, and provide well fairness and friendliness.

REFERENCES

[1] C. P. Fu and S. C. Liew, "TCP Veno: TCP enhancement for transmission over wireless access networks," *IEEE Journal on Selected Areas in Communications*, vol. 21, no. 2, pp. 216-228, 2003.

[2] D. Katabi, M. Handley, and C. Rohrs, "Congestion control for high bandwidth-delay product networks," *ACM SIGCOMM Computer Communication Review*, vol. 32, no. 4, pp. 89-102, 2002.

[3] A. Chockalingam, M. Zorzi, and V. Tralli, "Wireless TCP performance with link layer FEC/ARQ," in *Proc. IEEE International Conference on Communications*, vol. 2, 1999, pp. 1212-1216.

[4] A. Basalamah, "An FEC adaptive multicast MAC protocol for providing reliability in WLANs," *IEICE Transactions on Communications*, vol. 92, no. 5, pp. 1834-1840, 2009.

[5] L. S. Brakmo and L. L. Peterson, "TCP Vegas: End to end congestion avoidance on a global Internet," *IEEE Journal on Selected Areas in Communications*, vol. 13, no. 8, pp. 1465-1480, 1995.

[6] M. Gerla, M. Y. Sanadidi, R. Wang, *et al.*, "TCP Westwood: Congestion window control using bandwidth estimation," in *Proc. IEEE Global Telecommunications Conference*, vol. 3, 2001, pp. 1698-1702.

[7] S. Lar and X. Liao, "An initiative for a classified bibliography on TCP/IP congestion control," *Journal of Network and Computer Applications*, vol. 36, no. 1, pp. 126-133, 2013.

[8] S. Hagag and A. El-Sayed, "Enhanced TCP westwood congestion avoidance mechanism (TCP Westwood New)," *International Journal of Computer Applications*, vol. 45, no. 5, pp. 21-29, 2012.

[9] G. Wilson and R. Cyriac, "An enhancement to TCPW BBE by modifying the bandwidth estimation using modifieed EWMA international," *Journal of Advanced Research in Computer Science and Software Engineering*, vol. 2, pp. 278-281, 2012.

[10] D. G. Raimagia and C. N. Chanda, "A novel approach to enhance performance of Linux-TCP westwood on wireless link," in *Proc. Nirma University International Conference on Engineering*, IEEE, 2013.

[11] R. Wang, M. Valla, M. Y. Sanadidi, *et al.*, "Efficiency/friendliness tradeoffs in TCP Westwood," in *Proc. IEEE Seventh International Symposium on Computers and Communications*, 2002, pp. 304-311.

[12] L. A. Grieco and S. Mascolo, "TCP westwood and easy red to improve fairness in high-speed network," in *Proc. Seventh International Workshop Protocols for High Speed Networks*, Springer Berlin Heidelberg, 2002, pp. 130-146.

[13] H. T. Lee, F. L. Lian, and T. C. Fong, "Congestion control of transmission control protocol based on bandwidth estimation," *Journal of the Chinese Institute of Engineers*, vol. 33, no. 3, pp. 489-494, 2010.

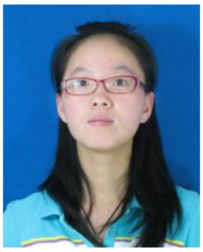
[14] P. Kaushika and M. Rathod Jagdish, "A survey on effectiveness of TCP westwood in mixed wired and wireless networks," *International Journal of Scientific & Engineering Research*, vol. 4, pp. 197-205, 2013.

[15] V. P. Reshamwala and K. D. Patel, "A review on Tcp westwood protocol for simulated and internet environment," *International Journal of Engineering and Computer Science*, vol. 2, no. 4, April, 2013.

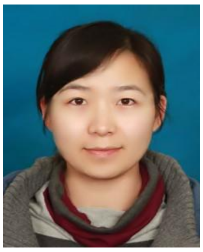
[16] X. Meng, J. Liu, and L. Wang, "Research on TCP improvement in hybrid network," in *Proc. IEEE International Conference on Computer Science and Automation Engineering*, vol. 1, 2011, pp. 650-653.

[17] K. Lan and N. Sha, "A CMT congestion window updates mechanism based on TCP Westwood," in *Proc. IEEE International Conference on Mechatronic Science, Electric Engineering and Computer*, 2011, pp. 2119-2122.

Rui-Qing Wu was born in Sichuan Province, China, in 1977. He received the M.S. degree in 2002 and the Ph.D. degree in 2010, both in electrical engineering from the University of Electronic Science and Technology of China (UESTC). He is with the School of Electronic Engineering, UESTC. His research interests include sensor network, control network optimization and multimedia communication.



Hong Jie was born in Jiangxi Province, China, in 1990. She received the B.S. degree in 2012 from Nanchang Hangkong University in electrical engineering. She is currently pursuing the M.S. degree with the School of Electronic Engineering, UESTC. Her research interests include control network optimization and network simulation.



Nan Ding was born in Hebei Province, China, in 1988. She received the B.S. degree in 2011 from Hebei Normal University in application electronic. She is currently pursuing the M.S. degree with the School of Electronic Engineering, UESTC. Her research interests include control network optimization and network simulation.

