Chapter 64
TCP BRJ: Enhanced TCP Congestion Control Based on Bandwidth Estimation and RTT Jitter for Heterogeneous Networks

Nan Ding, Rui-Qing Wu, and Hong Jie

Abstract In this paper, we propose an improved TCP scheme, TCP BRJ, which is capable of adjusting the initial slow-start threshold and congestion window in real time according to the bandwidth estimation in slow-start phase, dividing the network congestion grades based on round-trip time (RTT) jitter in congestion avoidance phase, distinguishing the random packet losses from the congestion packet losses, and reacting accordingly. Simulation results by NS-2 show that TCP BRJ provides more significant performance improvement in throughput, bandwidth utilization and fairness than TCP Reno and TCP Westwood in heterogeneous networks with high random bit-error rate (BER), and shows friendliness towards the widely used algorithm TCP Reno.

Keywords Bandwidth estimation • Round-trip time jitter • Wired/wireless networks • TCP Westwood • Loss differentiation

64.1 Introduction

With the rapid development of emerging wireless communication, a number of wired/wireless heterogeneous networks have been widely deployed nowadays. However, the traditional TCP protocol originally designed primarily for the wired networks has been unable to meet the performance requirements of increasingly complex wired/wireless heterogeneous networks. In wired networks, the random bit-error rate (BER) is negligible and the congestion is the main cause of packet loss, while in wireless networks, frequent random packet losses are unavoidable due to some non-negligible factors, such as high BER, fading and blackout [1]. The traditional TCP's reactive congestion control and avoidance mechanism taken the assumption that all packet losses are due to congestions become incapable of

N. Ding (✉) • R.-Q. Wu • H. Jie
School of Electronic Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China
e-mail: dingnan0807@qq.com

J. Mu et al. (eds.), *The Proceedings of the Third International Conference on Communications, Signal Processing, and Systems*, Lecture Notes in Electrical Engineering 322, DOI 10.1007/978-3-319-08991-1_64

differentiating the mixed packet losses. Therefore, TCP without modification suffers throughput degradations when used in heterogeneous networks.

In the past few years, many research efforts have been made to adapt TCP to heterogeneous networks. Such works can be classified into three main categories: split-connection schemes [2], localized link layer solution [3], and end-to-end schemes [4]. Among them, strictly end-to-end schemes at the transport layer have been paid much attention because they require no support from the network. As a representative end-to-end congestion control algorithm applied in wireless network nowadays, TCP Westwood [5], has evolved into many variants. When the packet loss occurs, it sets the slow-start threshold (ssthresh) and congestion window (cwnd) based on bandwidth estimation, which eliminates the influence of wireless random loss to some extent and provides significant throughput gains. However, the following shortages still exist in TCP Westwood when it is applied in heterogeneous networks with high BER. (1) It has no loss differentiation mechanism, which misinterprets random packet losses as congestion losses and results in an unnecessary congestion control. (2) It remains the slow-start phase intact as traditional TCP Reno [6] does, which initializes the ssthresh blindly with a fixed value leading to a decline of bandwidth utilization.

In order to overcome the above disadvantages, we propose a novel TCP BRJ scheme with loss differentiation mechanism and following improvements. (1) In slow-start phase, TCP BRJ adjusts the initial ssthresh according to bandwidth estimation in real time. (2) Every time after receiving an arrival acknowledgment (ACK) of new data segments, TCP BRJ calculates the RTT jitter and divide the network into five congestion grades based on the jitter in congestion avoidance phase. (3) When three duplicate ACKs occur, TCP BRJ differentiates the random packet losses from the congestion packet losses based on network congestion grades and reacts accordingly.

64.2 Overview of TCP Westwood

TCP Westwood (TCPW) is a sender-only modification of TCP NewReno [7]. The TCP sender determines bandwidth estimation (BWE) when each arrival ACK packet of new data is received. The estimation is based on information in the ACKs, and the rate at which the ACKs are received. After a packet loss indication, the sender uses BWE to set the cwnd and the ssthresh properly. Further details of BWE are presented in [5]. Here, we describe how the estimation is used to set the cwnd and ssthresh. Firstly, in TCPW, cwnd dynamics during slow start and congestion avoidance are unchanged, that is, they increase exponentially and linearly, respectively. A packet loss is indicated by (a) the reception of three duplicate ACKs, or (b) coarse timeout expiration. In case (a), TCPW sets cwnd and ssthresh as follows.

```
if (three DUPACKs are received) {        //packet loss
ssthresh = (BWE * RTTmin) /seg_size;
if (cwnd > ssthresh)    cwnd = ssthresh; } //congestion avoidance.
When coarse timeout expires, the cwnd and ssthresh are set as follows:
if (coarse timeout expires) {
cwnd = 1;
ssthresh = (BWE * RTTmin) / seg_size;
if (ssthresh < 2)    ssthresh = 2; }
```

64.3 RTT Jitter and Congestion Grade

A packet's round-trip time (RTT) consists of three parts: transmission delay, propagation delay and queuing delay. If all the packets pass through the same router and the packets are of the same size, the transmission delay and propagation delay are fixed. Then the changes of RTT are determined only by the queuing delay, which indicates the congestion degree of network. Therefore the network state can be characterized to a large extent by the RTT jitter, so TCP BRJ uses the adaptable RTT jitter to predict network states, and then divides the network congestion into different grades according to the previous forecast.

We use the following formula to calculate the RTT jitter based on RTT:

$$J_{sample}(k) = RTT_k - RTT_{k-1} \tag{64.1}$$

$$J_{estimate}(k) = \alpha * J_{estimate}(k-1) + (1-\alpha) * J_{sample}(k) \tag{64.2}$$

Where RTT_k and $J_{sample}(k)$ means respectively the RTT and RTT jitter measured when the kth ACK received at the sender. To improve the accuracy of RTT jitter, we use an exponentially-weighted moving average (EWMA) filter in formula (64.2) to calculate the estimation value of RTT jitter $J_{estimate}(k)$, where $\alpha \in [0, 1]$, and α is generally set to a fixed value equal to 0.125. From (64.2), we can see that if α is small, the value of $J_{estimate}(k)$ depends on $J_{sample}(k)$ and otherwise depends on $J_{estimate}(k-1)$. However, when the network state changes frequently, it is difficult to get accurate RTT jitter estimation adopting a fixed α. In TCP BRJ, we update the value of α dynamically according to the network states, which will be provided in following sections. If the network state changes quickly, α will be set to a smaller value, otherwise, if the network state is stable, we use a larger α. TCP BRJ obtains a more accurate RTT jitter estimation. The pseudo code is shown below.

```
if (network states change quickly) {
    $\alpha = | J_{sample}(k) - J_{estimate}(k-1) | / J_{estimate}(k-1)$;
    if ($\alpha > 1$)  $\alpha = 1/\alpha$;
} else  $\alpha = 0.875$;        // network states are stable
```

Table 64.1 Division of congestion grade

NCF	Congestion grade	Congestion degree
$(-\infty, -4)$	1	Serious under-load
$(-4, -1)$	2	Slight under-load
$[-1, 0.5]$	3	Normal network
$(0.5, 1)$	4	Slight congestion
$(1, +\infty)$	5	Serious congestion

This paper takes advantage of the network congestion flag (NCF) to realize a fine-grained classification of network congestion degree. The NCF is given by $\text{NCF} = (J_{\text{estimate}}(k) - T)/T$, namely the difference rate between RTT jitter estimation and a constant variable T. Where T is set to 0.5 ms, which is an experience value gained from simulation experiments. According to the value of NCF, the network congestion degree will be specifically divided into five grades as shown in Table 64.1.

64.4 Congestion Control of TCP BRJ

We note that in TCPW, the initial value of ssthresh is set to 1 artificially, which is hard to be adaptable to the network states on the condition of unknown bottleneck link bandwidth. And an inappropriate value of ssthresh will lead to a decline of bandwidth utilization. In this paper, we propose an improved slow-start phase, which adjusts the initial ssthresh and cwnd in real time according to BWE. During the detection period, if the current ssthresh is lower than BWE, the sender will set the ssthresh equal to BWE and increase the cwnd at a slightly higher speed than exponentially growth in TCPW, which will then make the slow-start process astringed fast. The pseudo code of improved slow-start phase is shown below.

```
if (ssthresh < $(BWE*RTT_{min})$/seg_size)
    ssthresh = $(BWE*RTT_{min})$/seg_size;
if (cwnd < ssthresh)   cwnd = cwnd + 1.5;
else  go to the congestion avoidance phase
```

The improved congestion avoidance phase of TCP BRJ realizes not only the division of congestion grade but also adopts a new cwnd adjustment strategy. The process of improved congestion avoidance phase is shown in Table 64.2.

Firstly, when the congestion grade is 1 and 2, the network utilization is very low, and the cwnd is increased by 1.1 and 1.2 respectively in order to make full use of the network resources as fast as possible. Secondly, the congestion grade equaled to 3 means a normal network state, and we increase the cwnd by 1. Thirdly, when the congestion grade grows to 4, which indicates a slight congestion, the sender will decrease the growth rate of cwnd appropriately. When the congestion grade is equal

Table 64.2 The adjustments in improved congestion avoidance phase

NCF range	Cwnd update	Grade	Remark
$(-\infty, -4]$	cwnd = cwnd + 1.2	1	Serious under-load network
$(-4, -1]$	cwnd = cwnd + 1.1	2	Slight under-load network
$(-1, 0.5]$	cwnd = cwnd + 1.0	3	Normal network
$(0.5, 1]$	cwnd = cwnd + 0.8	4	Slightly congestion network
$(1, -\infty)$	cwnd = cwnd + 0.6	5	Serious congestion network

to 5, the network is seriously congested, and then we increase the cwnd by a quite small value.

The proposed TCP BRJ also has a novel loss differentiation mechanism based on the bandwidth estimation and RTT jitter. When more than three duplicate ACKs occur, TCP BRJ differentiates the random packet losses from the congestion packet losses based on network congestion grades and reacts accordingly. If the congestion grade is larger than 3, TCP BRJ regards the packet loss as a congestion loss, otherwise, a random loss. More details are shown below.

```
More than three duplicate ACKs are received:
if (grade <= 3) {              //Wireless random loss
    ssthresh = int (cwnd * 4/5);
    cwnd = ssthresh + 3;
} else if (grade = 4) {          //Congestion loss
    sstemp = (BWE*RTT$_{min}$)/seg_size;
    ssthresh = int (sstemp);
    if (cwnd > sstemp)   cwnd = (cwnd + sstemp)/2;
} else {                     //Serious congestion loss
    ssthresh = int (sstemp);
    if (cwnd > sstemp)   cwnd = cwnd * 0.6;}
```

64.5 Performance Evaluations

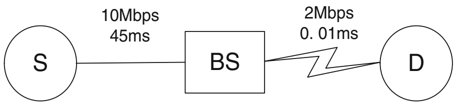
In this section, we simulate and analyze the performance improvements of TCP BRJ in throughput, bandwidth utilization, fairness and friendliness compared with TCP WestwoodNR (NS-2 modules of TCP Westwood with the NewReno feature) and TCP Reno in heterogeneous networks by the NS-2 simulator.

Throughput is an important feature of network performance, which is calculated by the effective amount of data delivered through the network based on received ACKs at the sender. The simulation topology is depicted in Fig. 64.1.

A single TCP connection running a long-live FTP application delivers data from 0 to 100 s. We run the simulation for BRJ, Reno and WestwoodNR, respectively. The BER at the wireless bottleneck link varies from 0.01 to 10 %.

The comparison of throughput is shown in Fig. 64.2. For random loss rate smaller than 0.1 %, all TCP schemes perform closely to each other. Besides that,

Fig. 64.1 Simulation topology for throughput



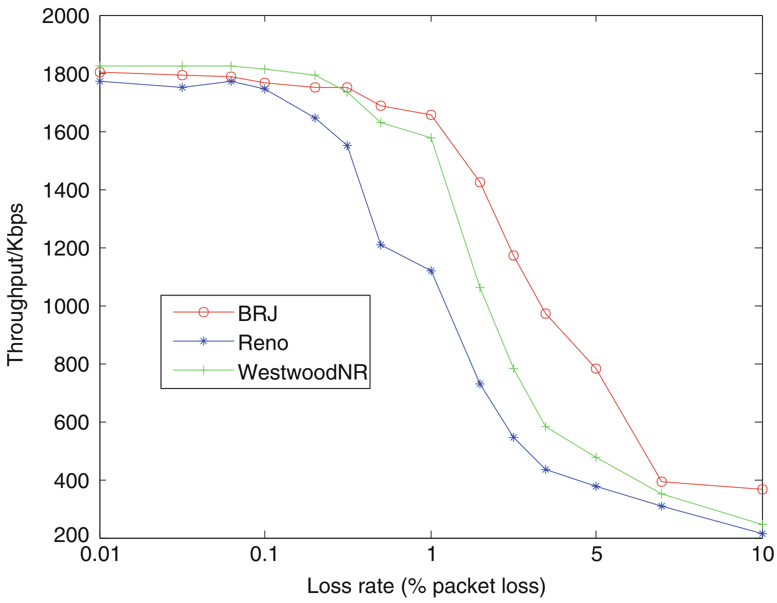


Fig. 64.2 Throughput comparison

BRJ starts to outperform the other TCP variants when loss rate is bigger than 0.1 %. What's more, the throughput of BRJ is especially superior to others when the loss rate varies from 1 to 5 %. At a very practical wireless loss rate, i.e., 2 %, BRJ outperforms WestwoodNR by 34.3 % and Reno by 95.3 %. Furthermore, the advantages of BRJ still exist in various bottleneck bandwidths as shown in Fig. 64.3.

We use the fairness index function proposed in [8] to evaluate the fairness of TCP schemes. Multiple connections of the same TCP scheme must interoperate and converge on their fair share. The simulation topology is shown in Fig. 64.4, where 10 same TCP flows share a 20 Mb/s bottleneck link with 1–5 % BER. We simulate different TCP schemes and the results are shown in Fig. 64.5. We can see that all TCP variants achieve satisfactory fairness index. What's more, from Fig. 64.6, which show the total throughput of 10 TCP flows, we can see that TCP BRJ greatly improves bandwidth utilization of bottleneck link compared to the others.

A friendly TCP scheme should be coexisted with other TCP variants and not cause them starvation. To verify the friendliness of TCP BRJ towards TCP Reno, we use the simulation topology in Fig. 64.4. There are 10 TCP flows running

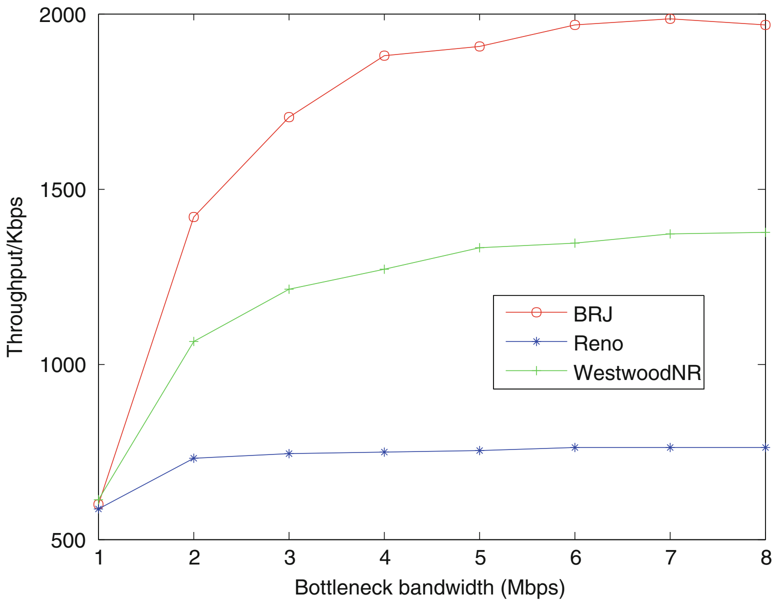


Fig. 64.3 Throughput in different bottleneck

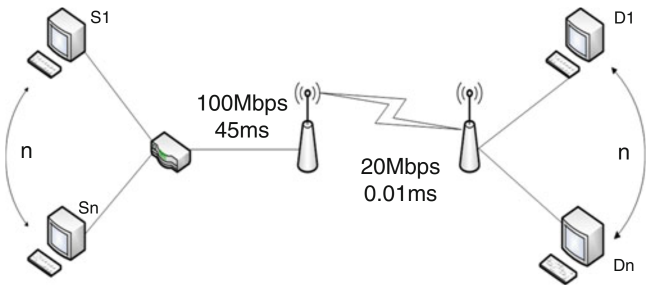


Fig. 64.4 Simulation topology for multiple flows

together through a bottleneck link with 0.1 % BER, where m TCP BRJ flows coexist with n TCP Reno flows. We vary the flows proportion of these two TCP schemes by adjusting the variables m and n. All ten connections are expected to share the bottleneck bandwidth equally, i.e., roughly 2 Mbps per connection.

The results are show in Fig. 64.7. It is observed that the bandwidth allocation of each TCP connection is close to its fair share at bottleneck link, except that TCP BRJ achieves a slightly higher throughput than Reno, but within a tolerable range.

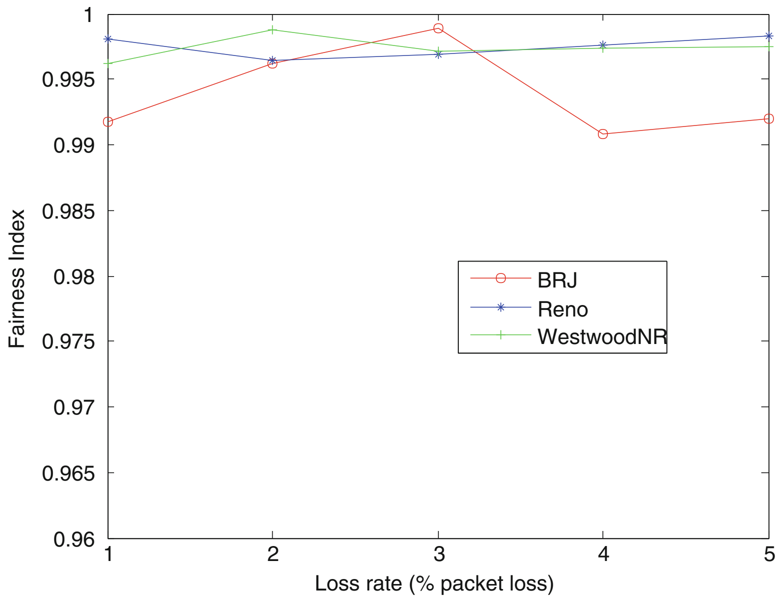


Fig. 64.5 Fairness index comparison

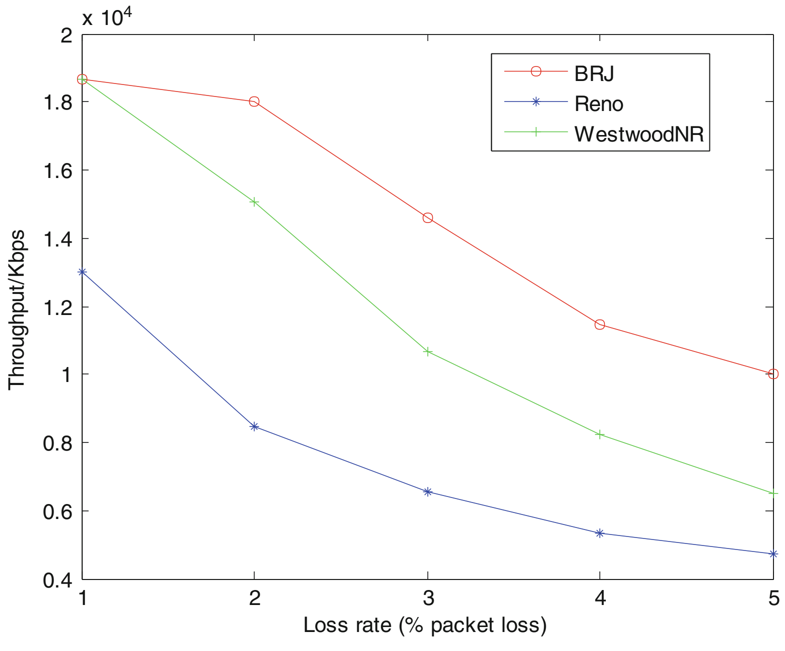


Fig. 64.6 Total throughput comparison

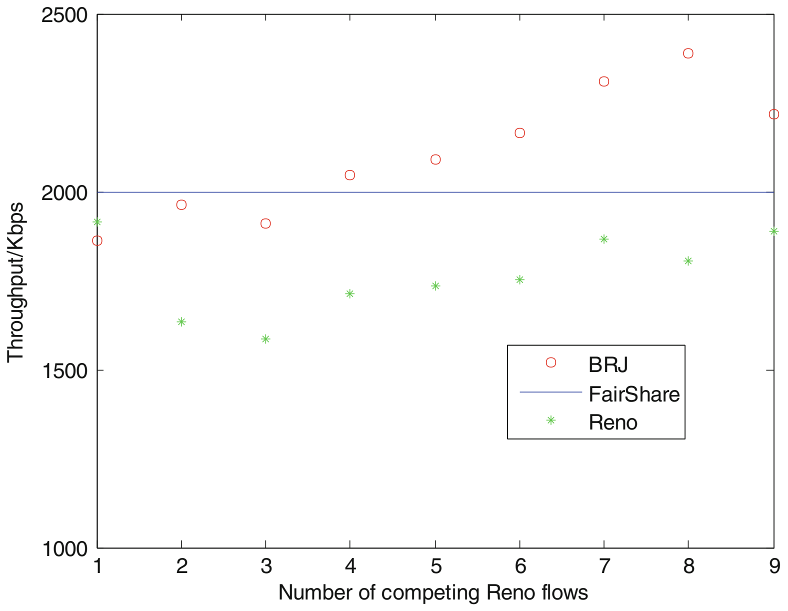


Fig. 64.7 Friendliness of BRJ towards Reno

Conclusion

In this paper, we propose a new TCP scheme, TCP BRJ, to improve the TCP performance in the wired/wireless heterogeneous networks. TCP BRJ has an improved slow-start phase and realizes a novel congestion avoidance phase. Moreover, it proposes a loss differentiation mechanism. The simulation results show that TCP BRJ provides a significant performance improvement in throughput, bandwidth utilization and fairness than others in heterogeneous networks with high BER, especially 1–2 %, a typical characteristic of wireless link. And for practical purpose, TCP BRJ is friendly towards TCP Reno.

Acknowledgment This work is supported by the Fundamental Research Funds for the Central Universities Project No. ZYGX2012J020.

References

1. Tsaoussidis V, Matta I (2002) Open issues on TCP for mobile computing. Wirel Commun Mob Comput 2:3–20
2. Byun HJ, Lim JT (2005) Explicit window adaptation algorithm over TCP wireless networks. IEE Proc Commun 152:691–696

3. Klemm F, Ye Z, Krishnamurthy SV et al (2005) Improving TCP performance in ad hoc networks using signal strength based link management. Ad Hoc Netw 3:175–191
4. Yanxiang Z, Fang S, Mingyan K (2010) A fuzzy packet loss differentiation algorithm based on Ack-timeout times ratio in heterogeneous network. In: IEEE international conference on Communications and Mobile Computing (CMC), vol 1, pp 453–457
5. Gerla M, Ng BKF, Sanadidi MY et al (2004) TCP Westwood with adaptive bandwidth estimation to improve efficiency/friendliness tradeoffs. Comput Commun 27:41–58
6. Stevens WR, Allman M, Paxson V (1999) TCP congestion control. Consultant
7. Henderson T, Floyd S, Gurtov A, Nishida Y (2012) The NewReno modification to TCP's fast recovery algorithm. RFC6582
8. Jain R, Chiu DM, Hawe WR (1984) A quantitative measure of fairness and discrimination for resource allocation in shared computer system. Digital Equipment Corporation, Eastern Research Laboratory, Hudson