Available online at www.sciencedirect.com
SCIENCE DIRECT®
Computer Networks 47 (2005) 219–237
COMPUTER NETWORKS
www.elsevier.com/locate/comnet

Improving TCP performance in integrated wireless communications networks

Kai Xu, Ye Tian, Nirwan Ansari *

Advanced Networking Laboratory, Department of Electrical and Computer Engineering, New Jersey Institute of Technology, University Heights, Newark, NJ 07102, USA

Available online 6 August 2004

Abstract

Many analytical and simulation-based studies of TCP performance in wireless environments assume an error-free and congestion-free reverse channel that has the same capacity as the forward channel. Such an assumption does not hold in many real-world scenarios, particularly in the hybrid networks consisting of various wireless LAN (WLAN) and cellular technologies. In this paper, we first study, through extensive simulations, the performance characteristics of four representative TCP schemes, namely TCP New Reno, SACK, Veno, and Westwood, under the network conditions of asymmetric end-to-end link capacities, correlated wireless errors, and link congestion in both forward and reverse directions. We then propose a new TCP scheme, called TCP New Jersey, which is capable of distinguishing wireless packet losses from congestion packet losses, and reacting accordingly. TCP New Jersey consists of two key components, the timestamp-based available bandwidth estimation (TABE) algorithm and the congestion warning (CW) router configuration. TABE is a TCP-sender-side algorithm that continuously estimates the bandwidth available to the connection and guides the sender to adjust its transmission rate when the network becomes congested. TABE is immune to the ACK drops as well as ACK compression. CW is a configuration of network routers such that routers alert end stations by marking all packets when there is a sign of an incipient congestion. The marking of packets by the CW-configured routers helps the sender of the TCP connection to effectively differentiate packet losses caused by network congestion from those caused by wireless link errors. Our simulation results show that TCP New Jersey is able to accurately estimate the available bandwidth of the bottleneck link of an end-to-end path; and the TABE estimator is immune to link asymmetry, bi-directional congestion, and the relative position of the bottleneck link in the multi-hop end-to-end path. The proactive congestion avoidance control mechanism proposed in our scheme minimizes the network congestion, reduces the network volatility, and stabilizes the queue lengths while achieving more throughput than other TCP schemes.

* Corresponding author. Tel.: +1 973 596 3670; fax: +1 973 596 5680.
E-mail addresses: kx2@njit.edu (K. Xu), yt2@njit.edu (Y. Tian), nirwan.ansari@njit.edu (N. Ansari).

1389-1286/$ - see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.comnet.2004.07.006

Keywords: Wireless TCP; Bandwidth estimation; Explicit congestion notification; Congestion control; Loss differentiation

1. Introduction

Communication networks have evolved greatly in the last decade. Packet switching technologies have eventually merged the traditional voice networks and data networks together into a converged and integrated multimedia network. The horizon of the converged integrated network is extending further to incorporate wired, wireless, and cellular technologies. The all-IP wired and wireless hybrid network is becoming a reality and the wireless network is getting more involved in our daily communications. The revolutionary cellular technologies and the network integration of WAN, LAN, WLAN and cellular networks stretch the Internet beyond the limits of geography and terrain. In integrated WAN + LAN + 3G cellular systems, illustrated in Fig. 1, more data and multimedia communications are carried end-to-end over the existing Internet protocol infrastructure. TCP/IP is the dominant communication protocol suite in today's multimedia applications. Nowadays, most of the Internet traffic is carried by TCP, including traffic generated by Web accesses, e-mails, bulk data transfers, remote terminals, etc. Spurred by the demand of wireless Internet, TCP/IP needs to depart from its original wired network oriented design and evolve to meet the challenges introduced by the wireless portion of the network. Internet Protocol (IP) [1] is a connection-less, best-effort based variable length packet delivery network layer protocol that does not guarantee the reliable, timely and in-order delivery of packets between end stations. TCP, the transmission control protocol [2], is a layer-4 transport protocol that uses the basic IP services to provide applications with an end-to-end connection-oriented packet transport mechanism that ensures the reliable and ordered delivery of data.

Improving TCP's performance in wireless IP communications has been an active research area. The performance degradation of TCP in wireless and wired-wireless hybrid networks, as reported in much research [3,4], is mainly due to its lack of the ability to differentiate the packet losses caused by network congestion from the losses caused by wireless link errors. TCP was originally designed primarily for wired networks. In wired networks, the random bit error rate (BER) is negligible, and congestion is the main cause of packet losses. The TCP sender behavior of adjusting the sending rate of data packets is triggered by the self-clocking acknowledgement (ACK) sent by the corresponding receiver after successfully receiving the data packet. When packet loss occurs at a congested link due to buffer overflow at the intermediate router, either the sender receives duplicate ACKs (DUPACK) or the sender's retransmission timeout (RTO) timer expires. These events activate the sender's congestion control mechanism by which the sender reduces the size of its transmission window, or congestion window (*cwnd*) in TCP terminology, resulting in a lower transmission rate to relieve the link congestion. This is a reactive congestion control scheme, in which the action is triggered by the sender's self-induced congestion. Such TCP sender behavior works fairly well in the wired networks where packet losses are almost always caused by link congestion; and packet losses due to bit errors are usually negligible or, if any, not exceeding one packet loss per *cwnd*. However, in wired/wireless heterogeneous networks, high BER, fading and blackout become non-negligible factors for packet losses. Standard TCP's congestion control and congestion avoidance mechanisms based on the assumption that all packet losses are due to congestion become incapable of handling the mixed packet losses. TCP without modification exhibits throughput degradations when used in wired/wireless heterogeneous networks.

Many analytical and simulation-based studies of TCP performance in wireless environments assume an error-free and congestion-free reverse channel that has the same capacity as the forward channel. Such an assumption does not hold in many real-world scenarios, particularly in the hybrid networks constituted by various wireless

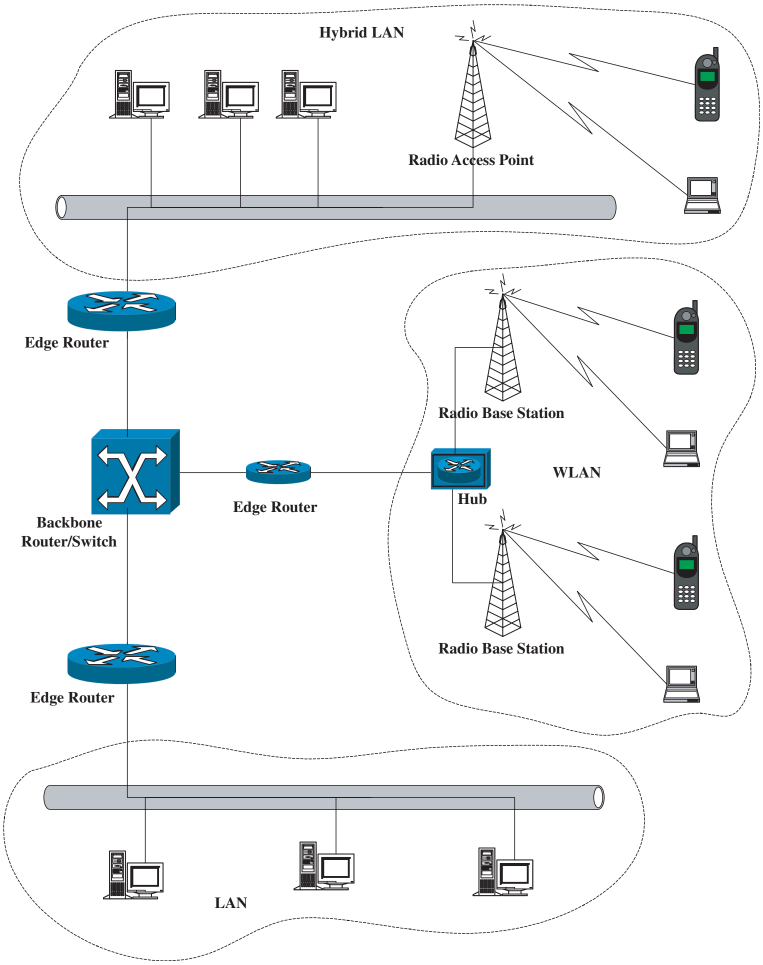

Fig. 1. A sketch of integrated networks.

LAN (WLAN) and cellular technologies. For instance, in a wired-to-wireless data connection, where the wireless portion is implemented through 3G cellular systems, the forward downlink from the base station to the wireless device such as a 3G cellular phone or a PDA is usually provisioned with higher link capacity than the reverse uplink [5]. Second, the ACKs traveling on the wired portion of the end-to-end path could very well face congestion since the wired portion of the path is usually shared by several other nodes and competing flows. Furthermore, the high BER also applies to the wireless link from the cellular device to the base station and, therefore, causes loss of ACK packets.

In this paper, we first study, through extensive simulations, the performance characteristics of four representative TCP schemes, namely TCP New Reno, SACK, Veno, and Westwood, under the network conditions of asymmetric end-to-end link capacities, correlated wireless errors, and link congestion in both forward and reverse directions. We then propose a new TCP scheme, called TCP New Jersey, which is capable of distinguishing wireless packet losses from congestion packet losses, and reacting accordingly. TCP New Jersey is an improvement to TCP Jersey [6] which we previously proposed for the mixed wired and wireless networks. The rest of the paper is organized as follows: In Section 2, we summarize the challenges

TCP faces in wireless environments. Section 3 briefly reviews the related work in improving the TCP performance for wireless IP communications. Section 4 presents the TCP New Jersey scheme. We then present in Section 5 various simulation results under different network conditions. We conclude the paper in Section 6 with a summary of the results and highlights of future work.

2. Challenges of wireless TCP

TCP schemes suffer from performance degradation when used in heterogeneous networks. Some factors that affect TCP's performance are summarized in this section.

2.1. Correlated errors

Unlike the bit errors in wired networks, in which they are considered as random and independent, the error process in radio transmission is caused by different phenomena [7], including path loss, fast fading, slow fading, and noise and interference from other devices like microwave ovens. The relationship between distance and the BER is not linear. Instead, the mean BER often remains constant up to a certain distance threshold, and then degrades rapidly. This is due to the behavior on the receiver side, where all signals below a threshold are discarded. A simple and widely used model that characterizes the bit errors of the wireless channel between two stations is the Gilbert–Elliot model, also known as the two-state Markov model, depicted in Fig. 2, where the process can be in either one of the two states, namely the good state or the bad state, and each state is associated with its BER, say, e_G and e_B. Within each state, bit errors occur independently of each other. The transition between the two states is modeled as a continuous-time Markov chain as depicted in the figure, where P_{GB} and P_{BG} are the transition rates from the good state to the bad state and vice versa, respectively.

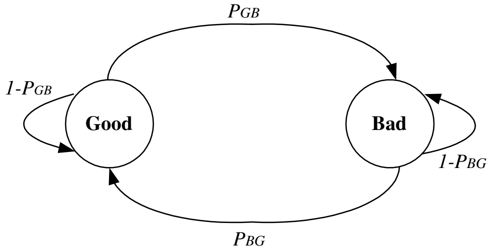

Fig. 2. Two-state Markov error model.

The mean sojourn time of the channel staying in one state can be approximated by $T_G = 1/P_{GB}$ and $T_B = 1/P_{BG}$, where T_G and T_B are exponentially distributed. Let $e = [e_G, e_B]$, and $\pi = [\pi_G, \pi_B]$ be the steady state probabilities of the channel being in the good and bad states. The mean BER of the wireless channel under this model is then $\bar{e} = \pi e'$. Often, the long-term fraction of the time the channel spends in the good (or bad) state is used to represent the state probabilities, which is defined as $f_G = E[T_G]/(E[T_G] + E[T_B])$, and $f_B = E[T_B]/(E[T_G] + E[T_B])$, respectively.

2.2. Burstiness of ACKs

TCP flow control is self-clocking. In essence, the arrivals of the ACKs at the sender trigger the sending of new packets and advances of the congestion window. However, ACK compression [8,9], a phenomenon caused by the queuing in the reverse path of a TCP flow can cause the almost instantaneous arrival of bursts of ACKs at the sender. This can break TCP's self-clocking and cause long bursts. Suppose several ACKs have been lost between two successfully received ACKs, the sender will send out a number of packets, indicated by the newly received ACKs, back to back. This burdens the forward path with an abrupt load increase and exacerbates the forward path condition. The effect of this problem can be devastating to some rate-based TCP schemes. Usually such schemes assume that short intervals between ACKs received at the sender imply a large available bandwidth in the forward path, since it is assumed that the ACK stream on the reverse path preserves the inter-packet dispersion on the forward path. The TCP self-clocking mechanism would therefore be influenced. The sender would make incorrect judgments of the forward path.

2.3. Link asymmetry

In networks with asymmetric links, TCP's performance in terms of the throughput also largely depends on the characteristics of the reverse link. Asymmetric links can be results of different physical designs of the forward and backward communication channels, or imbalanced network traffics in both directions. In 3G cellular networks, multimedia data transmission is carried through the wireless channel, and so is conventional voice data. The traffic on the downlink from the base station to the mobile device tends to consume more bandwidth than the traffic on the uplink. Also the transmission power of a base station is much higher than a mobile device. 3G wireless networks consider these facts in its design and can possibly control the link asymmetry, e.g., via pricing policy. The asymmetric link problem results in rate and delay variation. In a data path, the reverse channel may suffer from packet loss even when the condition on the forward channel is moderate. The effect of this problem can be ravaging to the design of some TCP schemes since most TCP schemes assume the packet loss occurs in the forward channel. The TCP self-clocking mechanism based on the ACK feedbacks would therefore be influenced and even be dominated by a poor condition on the reverse channel. The sender would make incorrect judgments of the forward channel condition, thus exacerbating the adverse effect on the reverse channel.

3. Related work

Standard TCP schemes such as Reno are simple and effective for reliable data transfer in wired networks, where packet loss due to bit errors is rare. As a transport layer protocol, standard TCP probes the available bandwidth of the network by continuously increasing the window size until the network is congested, and then decreases the window size multiplicatively, e.g., the additive-increase-multiplicative-decrease (AIMD) algorithm [10]. The congestion is presumably indicated by packet losses. The TCP sender backs off its congestion window whenever it is experiencing a packet loss to alleviate the congestion. In all-IP heterogeneous networks, where wired network interconnects with other wireless networks, e.g., cellular, satellite, and ad hoc networks, congestion is no longer the only cause of packet loss. Transmission errors, such as, random bit errors on the wireless link, signal fading, and mobile handoff process, also contribute a significant amount to packet loss. The challenges of wireless networks include high BER, limited bandwidth, unexpected disconnections, and link asymmetry [11]. If the packet losses due to these causes are misinterpreted by the sender as congestion, the congestion window would be reduced unnecessarily. In these circumstances, the standard Reno scheme experiences significant performance degradation.

TCP Reno [12] is incapable of handling multiple packet losses within one transmission window, which is a very likely situation in wireless links [13,14]. TCP New Reno [15] and TCP SACK [16] are modifications to the original Reno scheme, both of which handle the problem of multiple losses within one window. These schemes respond to packet losses reactively. TCP Veno [17], Westwood [18], and TCP Jersey are proactive schemes that intend to improve the flow control and avoid packet losses based on the estimation of certain network parameters at the sender side. In the following, we briefly discuss the characteristics of these TCP modifications.

3.1. TCP New Reno

In the standard Reno scheme, the fast recovery algorithm takes care of a single packet drop from one window. After one lost packet is recovered, Reno terminates the fast recovery algorithm. Therefore, multiple dropped packets would force the Reno scheme to invoke fast recovery again and again, and slows down the recovery of the lost packet.

New Reno modifies the fast recovery algorithm of Reno to cope with multiple losses from a single window, which is one of the characteristics of wireless networks where a fading channel may cause contiguous packet loss. In New Reno, the fast recovery algorithm does not terminate until multiple losses, indicated by the receipt of partial acknowledgements from one window, are all recovered. The limitation of New Reno is, however, that it cannot distinguish the cause of the packet loss, and thus a more effective fast recovery algorithm cannot be implemented.

3.2. TCP SACK

TCP SACK is a selective acknowledgement option for TCP, targeting at the same problem that New Reno tries to tackle. While the feedbacks of Reno and New Reno are based on the cumulative acknowledgements, SACK employs the selective repeat retransmission policy. It uses information fields called SACK blocks to indicate discontinuous blocks of data that have been successfully received and queued at the receiver buffer. The SACK blocks are carried back to the sender via the ACK packets. Hence, the sender has better, or, finer-grain knowledge about the exact packets that have lost rather than the limited knowledge about the loss at the left edge of the window only as in the standard ACK scheme. SACK implementation requires modification at the receiver side.

SACK blocks are encoded in the TCP option field, which inherently limits the number of SACK blocks that one ACK can carry. Moreover, as in Reno, SACK reactively responds to packet losses, and therefore has limited ability of congestion avoidance.

3.3. TCP Veno

TCP Veno estimates the backlogged packets in the buffer of the bottleneck link, a technique first developed by TCP Vegas [19], and uses this estimation to differentiate the random error losses from the congestive losses. If the number of backlogged packets is below a threshold, the loss is considered to be random. Otherwise, the loss is said to be congestive. If the loss is considered congestive, Veno adopts the standard Reno scheme. For the loss due to a random error, it increases the congestion window in a conservative manner, i.e., sending one packet on receiving every other ACK.

It is reasonable to treat the number of backlogged packets as an indication of the network congestion level. However, under some circumstances, this indication could be irrelevant.

3.4. TCP Westwood

TCP Westwood employs an end-to-end bandwidth estimator at the sender side. The sender derives the network bandwidth by exploiting the rate and pattern of the returning ACK stream. Upon experiencing a packet loss, the sender adjusts the congestion window dynamically according to the estimation. It is rather intuitive that the sender should send packets at a rate that utilizes the network resource most efficiently. The problem is how accurate this estimated rate could be. TCP Westwood claims improved performance over TCP Reno and SACK while achieving fairness and friendliness.

3.5. TCP Jersey

TCP Jersey develops two key components in its scheme, congestion warning (CW) and available bandwidth estimation (ABE), respectively. CW is a packet marking scheme that is different from explicit congestion notification (ECN) [20] in the following ways: First, ECN marks packets probabilistically when the average queue length lies between min_{th} and max_{th}, whereas, CW marks all the packets when the average queue length exceeds a threshold. This non-probabilistic marking scheme leaves the TCP sender, which receives the marks, to decide its window adjustment strategy rather than being influenced by the probabilistic marking of the packet in ECN. Second, CW inherits the same information bits used in the original ECN implementation but with simpler parameter settings. So, CW is not as sensitive as ECN to parameter settings [21]. The purpose of CW is to convey a simple image of the bottlenecked queue to the sender. Jersey also develops an ABE module, which is comparable and functions similarly to Westwood's rate estimator but with a different implementation. This will be described in detail in Section 4.1.

TCP Jersey adopts slow start and congestion avoidance from New Reno, but implements the rate-based congestion window control procedure based on ABE. It operates as follows. If an ACK is received without the CW mark, it proceeds as New Reno. If the received ACK or the third DUPACK is marked with the CW bit, it calls the rate control procedure to adjust the window size. When the third DUPACK is received without the CW mark, Jersey concludes that the packet drop is caused by a random error, and therefore it enters the fast retransmit without adjusting the window size.

TCP Jersey combines ABE and CW so that the TCP sender could set its congestion window to a more sensible value when congestion is detected. Also, such a combination improves TCP's ability to differentiate random wireless packet losses from losses caused by congestion.

4. TCP New Jersey

TCP Jersey was previously developed to improve TCP performance in the heterogeneous networks. In this study, we have made improvements to the algorithms used in the original Jersey to address the inaccuracy in bandwidth estimation caused by the ACK burstiness, and to further minimize the volatility of the flow load by stabilizing the transmission rate. We thus call the improved scheme TCP New Jersey (the new TCP Jersey scheme).

4.1. Timestamp-based available bandwidth estimation (TABE)

The original TCP Jersey employs an available bandwidth estimator (ABE) at the sender side to probe the end-to-end path bandwidth available to the connection. The sender therefore adjusts the congestion window proactively to better utilize the network resource and avoid the congestion caused by flows contending for the limited bandwidth. ABE operates based on the returning rate of the ACK packets on the reverse link. The gap between the consecutive ACKs conveys the information of the network utilization. Suppose the reverse link for the ACK is ideal, i.e., symmetric, congestion free and error free, the ACK pattern resembles the packet arrival pattern at the receiver side. The faster the packets arrive, the more the network resource can be utilized. ABE estimates the available bandwidth according to Eq. (1):

$$R_n = \frac{\mathrm{RTT} \times R_{n-1} + L_n}{(t_n - t_{n-1}) + \mathrm{RTT}}, \tag{1}$$

where R_n is the estimated bandwidth when the nth ACK arrives at time t_n, t_{n-1} is the previous ACK arrival time, L_n is the size of data that the nth ACK acknowledges, and RTT is the TCP's estimation of the end-to-end round trip time delay at time t_n.

The sender then interprets this rate as the optimal congestion window (ownd) in unit of segment by Eq. (2):

$$\mathrm{ownd}_n = \frac{\mathrm{RTT} \times R_n}{\mathrm{seg\_size}}, \tag{2}$$

where seg_size is the fixed segment size.

This simple yet effective rate estimator provides a good guideline for the sending rate when packet drop mostly occurs at the forward link and the reverse link has relatively moderate error rates for ACKs to get through. The result from the estimator reflects the desirable sending rate for the forward channel. In practice, however, the reverse link conditions are not always as good as expected. For instance, when the network transmits bi-directional traffic of a similar amount, or when the reverse link is a wireless channel, which suffers from non-negligible transmission errors, the reliability and accuracy of the result from the ABE module is compromised.

In TCP New Jersey, we propose a remedy for the above weakness. The reliability of the available bandwidth estimation comes from the strong resemblance between the forward traffic pattern and the reverse ACK pattern, an assumption which hardly holds when the reverse channel also experiences congestion and errors. The consequences, such as ACK compression, ACK delay, and ACK losses, can have a significant impact on ABE's integrity, and further on the overall performance of the TCP scheme. The TCP timestamp option proposed in RFC 1323 [22] provides a readily available solution to overcome this problem. In New Jersey, instead of using the ACK arrival time, the packet arrival time, which is stamped by the receiver and echoed back by the ACK according to

RFC 1323, is used in the bandwidth estimation. Therefore, in Eq. (1), t_n becomes the arrival time of the nth packet at the receiver. This is equivalent to having the estimation done at the receiver but without the need for the receiver to explicitly feed back the estimation. Since RFC 1323 timestamp option is widely implemented in most of the TCP protocols, this solution incurs no additional overhead to the overall protocol implementation. We call the improved ABE algorithm the timestamp-based ABE, or TABE. The timestamps, which are delivered by ACKs and received at the sender side, closely reveal the arrival traffic pattern at the receiver side. The estimated available bandwidth by TABE is thus less affected by the reverse link conditions and immune to the ACK loss on the reverse link.

4.2. Congestion-less slow start threshold adjustment

Almost all the TCP variants clearly divide the congestion control process into two distinctive phases, namely the slow start (SS) and the congestion avoidance (CA). The division is determined by the relationship between two TCP state variables, the slow start threshold (*ssthresh*) and the congestion window (*cwnd*). When *cwnd* is less than *ssthresh*, the process is in the SS phase and *cwnd* is doubled for every RTT; when *cwnd* exceeds *ssthresh*, it is in the CA phase and *cwnd* is increased by one for every RTT. The *ssthresh* is only adjusted when there is a packet loss detected either by receiving triple duplicated ACKs (TD) or an expiry of the retransmission timer (RTO). Upon such loss indications, *ssthresh* is set to half of the current *cwnd*, with *cwnd* set either to half of its current value when TD occurs, or 1 when RTO occurs. This is the best TCP can do to coarsely probe the available bandwidth in lieu of a more accurate measurement. This algorithm is called additive-increase-multiplicative-decrease, or AIMD. Clearly, it is a reactive approach to congestion control. Hoe and Wang et al. [23,24] improved the speed of the initial probe by estimating the bandwidth based on the inter-arrival times of the first several ACK packets and setting the *ssthresh* accordingly.

Taking advantages of New Jersey's ability to accurately estimate the available network bandwidth on the fly, the slow start threshold can therefore be adjusted in a more effective and dynamic way. Basically what the estimation result implies is the amount of bandwidth that can be utilized without causing network congestion. In New Jersey, upon receiving an ACK that acknowledges the delivery of a new packet, the sender sets *ssthresh* to *ownd* computed by Eq. (2). Therefore, *ssthresh* closely follows the dynamics of the available bandwidth of the path, by which the process enters and leaves the SS and CA phases proactively, and is adaptive to changing network conditions. By doing this, the sending rate would have a fast response to every changing network condition and converges to the optimal rate faster and more accurately without self-induced congestion, as the reactive approaches would otherwise cause.

5. Simulation results

With the increasing complexity and a wide variety of the applications that a network carries, no TCP scheme is optimal in performance in terms of data transfer, and there unlikely exists a universal solution to all the problems. In this section, we compare several TCP schemes in various network setups to reveal the problems in heterogeneous networks.

Both New Reno and SACK take multiple packet drops from one window into account, but both of them still have a limitation on distinguishing the cause of packet drops. These TCP schemes respond to packet loss reactively, and hence the ability of avoiding congestion is limited. TCP Veno, Westwood and Jersey are schemes that try to prevent network from congestion by proactively adjust the sending rate according to the network conditions. We simulate the aforementioned TCP schemes to demonstrate their abilities of responding to network congestion, link error, and network asymmetry. Throughput is an important metric for the performance comparison but is not the only one. In our simulations, we also pay particular attention to the congestion window, queue stabilities, and the ability to avoid congestion. Also, we specifically evaluate the TABE performance of New Jersey. Each simulation represents several runs, and the median values are taken. The simulations are conducted using the NS-2 network simulator [25].

5.1. Simulations on TABE performance

Both the Westwood and Jersey rate estimators exhibit good performance when the reverse link is congestion free and error free [6]. Hence, we test them under the condition that the reverse channel has contending traffic.

We test the two rate estimators used by Westwood and Jersey, respectively. Under the given network conditions shown in Fig. 3, a long-lived FTP application, whose packet size is 1000 byte, is running from S to D and the bottleneck link is a 1.5 Mbps link from S to R1. In this paper, the buffer size at each link takes the value of the bandwidth-delay product (BDP) of the link. If the BDP value is too small, the buffer size is set as 20 packets. Both estimators detect the correct bandwidth most of the time, but exhibit fluctuation and overshoot at some points, as illustrated in Fig. 4.

Constant bit rate (CBR) traffic is injected from C2 to C1, traversing the link R2 to R1, which is part of the reverse channel of the target traffic from S to D. The rate of the CBR traffic varies in time. From 1 to 5 s, the CBR traffic has a rate of 1 Mbps, which causes no significant congestion. Both estimators' outputs are close to the bottleneck bandwidth with a little over-estimation of Westwood and fluctuation of Jersey. From 5 to 10 s, when there is no reverse traffic, both estimators stay at the desirable rate. From 10 to 15 s, the reverse channel is totally jammed by the cross traffic. Westwood starts to exhibit over-estimates and Jersey shows under-estimates. From 15 to 20 s, the reverse channel is released by the cross traffic. Both estimators exhibit significant overshoot at the beginning, which is possibly caused by the ACK compression effect on the reverse link. From 20 to 30 s, the ACK stream on the reverse link experiences no congestions caused by the cross traffic, and both estimators produce satisfactory estimates that closely track the bottleneck bandwidth. The worst case of the rate estimation happens after the totally jammed link being released due to the ACK compression effect.

Under the same network conditions, we test TABE implemented by TCP New Jersey. By using the timestamp function in the Jersey estimator, TABE in New Jersey almost always produces the desirable estimation results. Some tolerable fluctuation occurs after dramatic change of the traffic load on the reverse link, as depicted in Fig. 5. This simulation shows that the traffic load on the reverse link has a minimal effect on TABE's accuracy.

A network setup, shown in Fig. 6, with more complicated traffic patterns and added wireless errors is used to further test the performance of TABE.

The wireless errors are generated using the two-state Markov error model introduced in

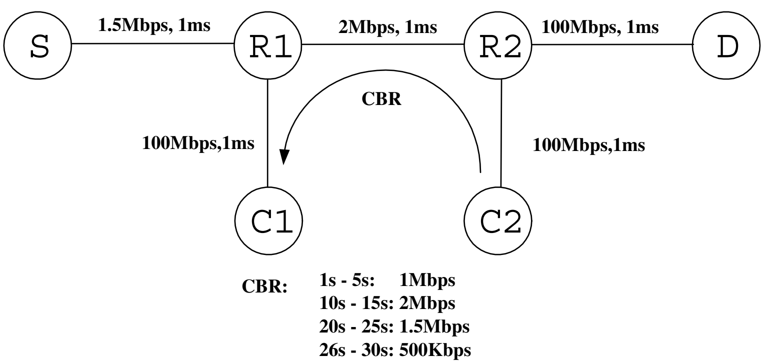

Fig. 3. Network setup for comparing the rate estimators of Westwood and New Jersey.

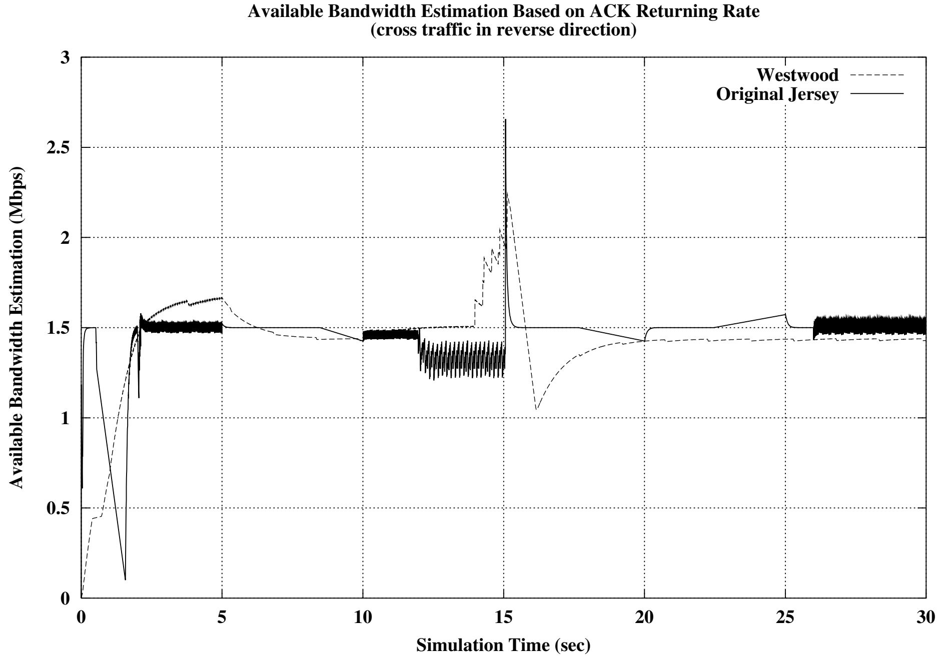

Fig. 4. Comparison results of the rate estimators of Westwood and Jersey.

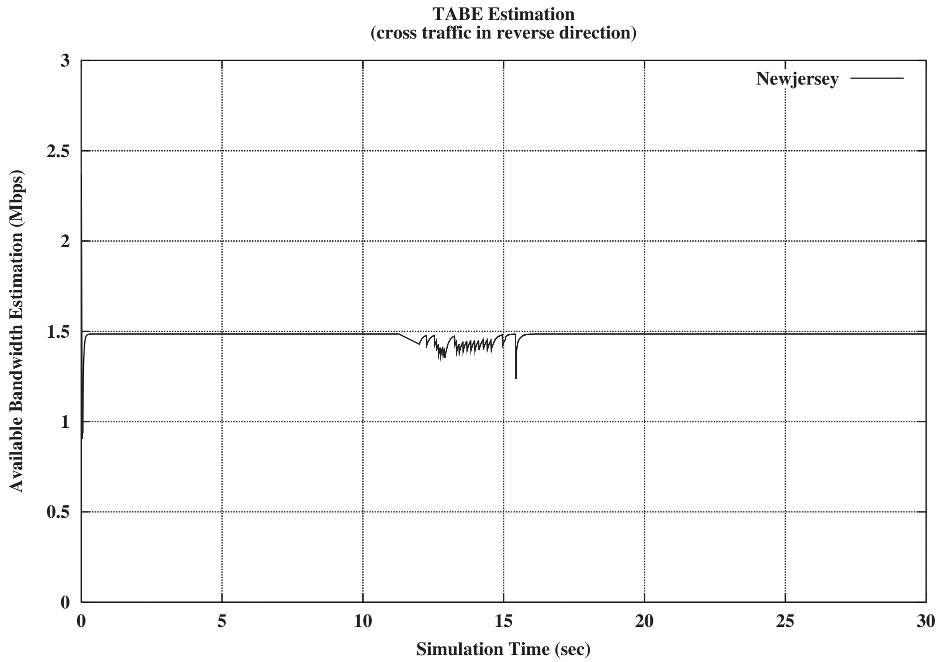

Fig. 5. Output of the rate estimator of New Jersey.

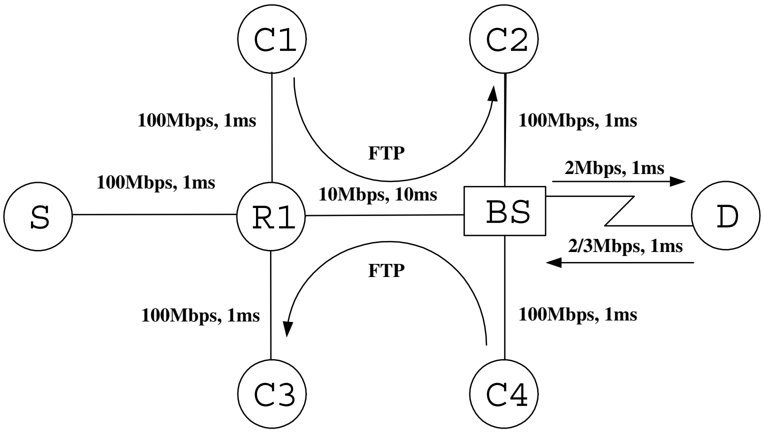

Fig. 6. Network setup for comparing New Jersey with other TCP schemes.

Section 2.1. This network setup resembles a possible structure of a cellular system integrated with a more reliable wired network in practice. The bottleneck link is at the last wireless hop, which incurs wireless losses as well. In the wired portion, flows from C1 to C2 and C4 to C3 are injected to simulate the bi-directional cross traffic in the real network. The cross traffic flows in both directions are short-lived FTP flows. All short-lived FTP sessions have independent exponentially distributed intersession gap with a mean of 1 s, and the session durations are drawn from independent exponential distributions with a mean of 3 s. These Poisson arrivals of cross traffic flows simulate the realistic model of traffic contending for bandwidth in the real world. The link from BS to D is the downlink between the base station and the mobile device. The link asymmetry is represented by the different bandwidth on the downlink and uplink. A long-lived FTP application is running from S to D throughout the simulations.

The result in Fig. 7 shows that New Jersey is able to capture the bottleneck bandwidth most of the time, while Westwood exhibits significant overshoots from time to time.

Using the same network setup as shown in Fig. 6, we further compare TCP New Jersey's performance to other TCP modification schemes. We investigate the sender's congestion window dynamic, the goodput, and the queue length of the buffer at the wireless link.

The simulations generate a vast amount of data. For illustrative purposes, we choose to show comparison results of Westwood, representing proactive TCP schemes, vs. New Jersey (Figs. 8–15); and SACK, representing reactive schemes, vs. New Jersey (Figs. 12–15). The comparison results of other schemes vs. New Jersey are omitted because similar results are produced as in these representative schemes. Nevertheless, we present a comprehensive comparison of the goodput of all aforementioned schemes.

Figs. 8 and 12 show that the congestion window dynamics of other schemes are rather oscillatory. Due to the congestion and errors, reactive schemes, such as SACK, experience packet losses and timeout from time to time, as reflected in the congestion window dynamics. New Jersey, on the other hand, exhibits a more stable congestion window dynamic with much less oscillations.

The goodput is basically the rate at which the data packets are successfully received in order and ACKed. Figs. 9 and 13 show that New Jersey leads all other schemes in terms of goodput.

The simulation also demonstrates that New Jersey has a more stable queue length at the buffer of the bottleneck link, i.e., the wireless link in this case (Figs. 10 and 14). The queuing dynamics can be viewed more clearly in the zoomed graphs (Figs. 11 and 15). A stable queuing process, i.e., a less volatile queue length, results in less delay

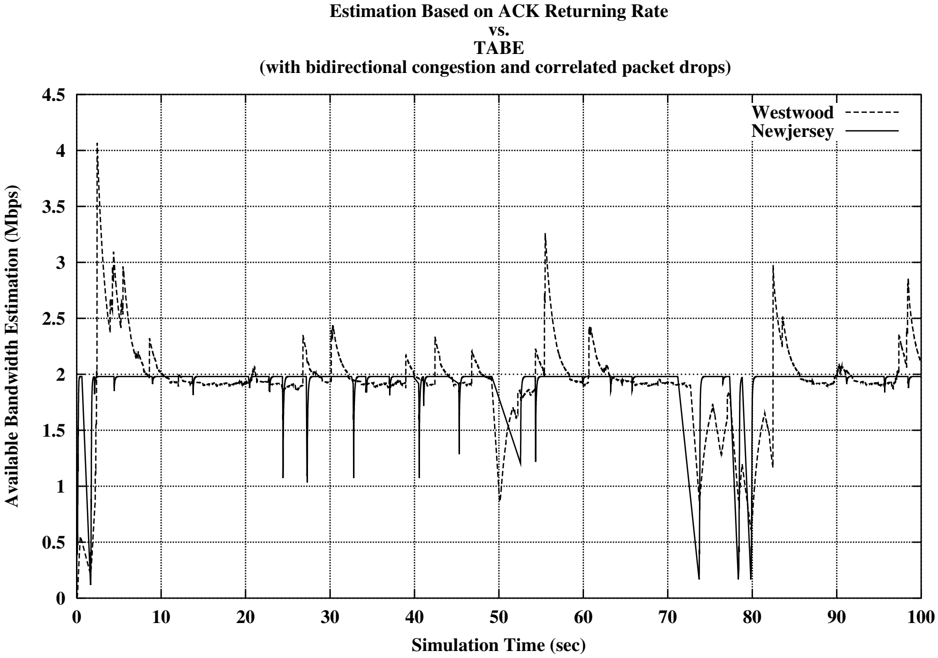

Fig. 7. Comparison of rate estimator of Westwood and New Jersey.

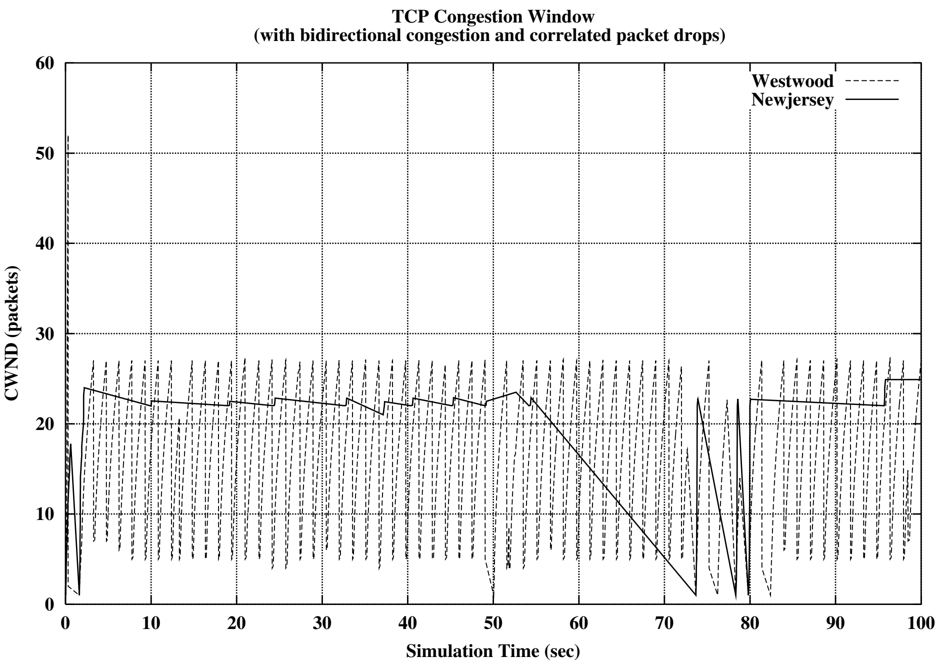

Fig. 8. TCP congestion window development: Westwood vs. New Jersey.

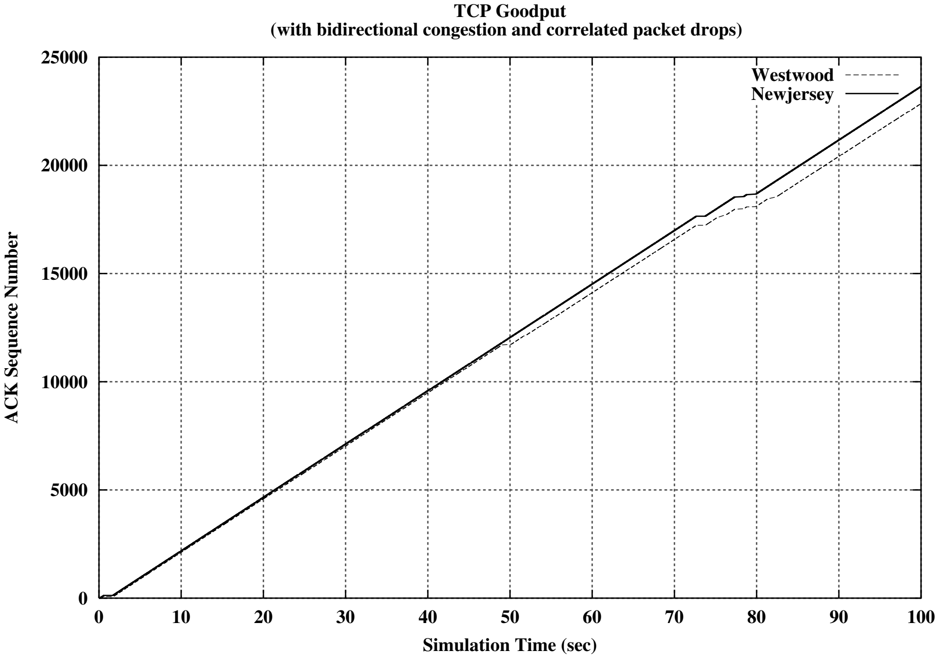

Fig. 9. Goodput comparison: Westwood vs. New Jersey.

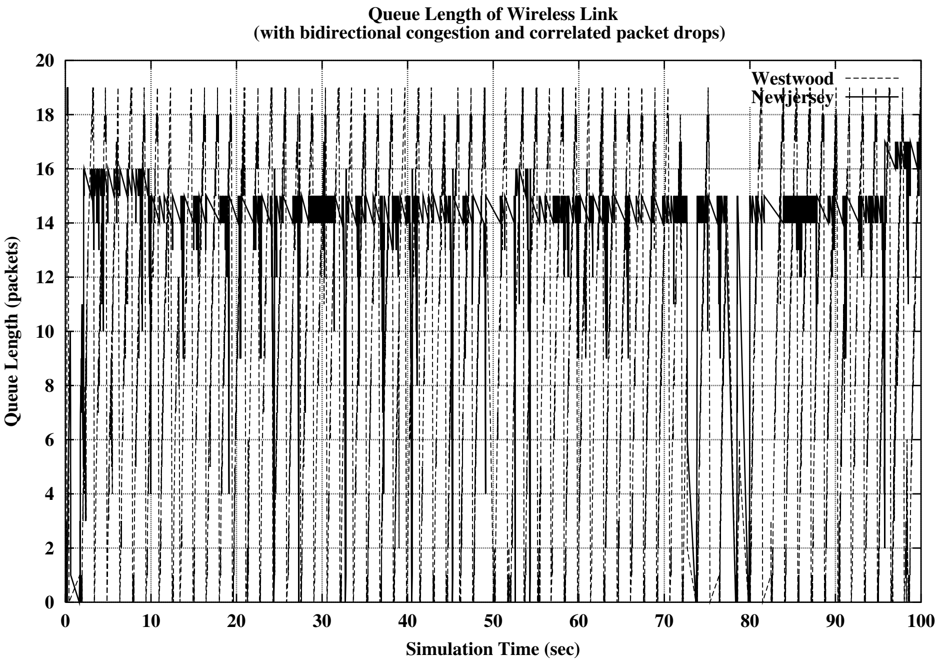

Fig. 10. Comparison of queue length at the bottleneck link: Westwood vs. New Jersey.

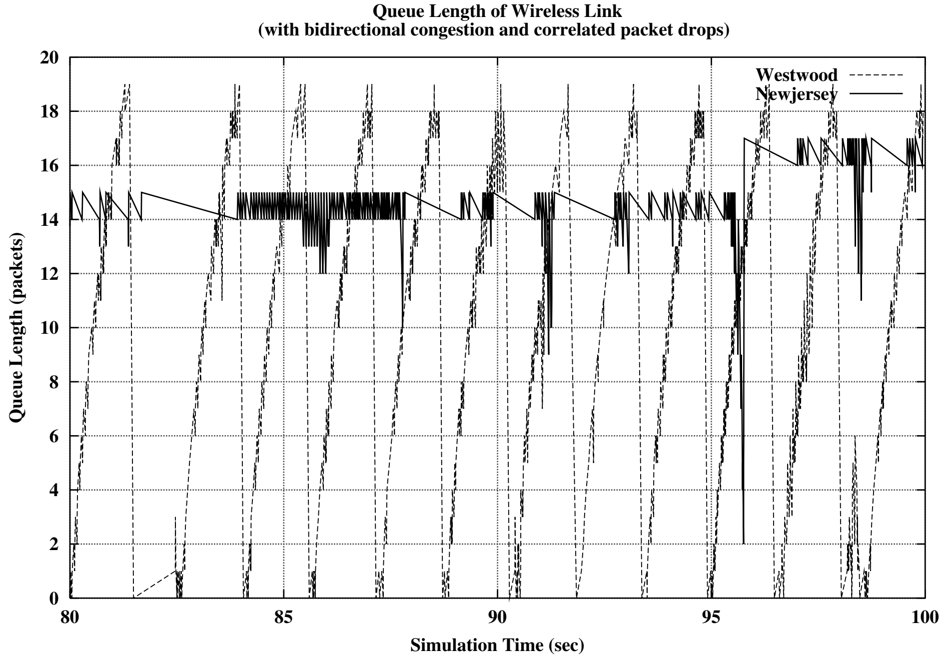

Fig. 11. Zoomed view of comparison of queue length at the bottleneck link: Westwood vs. New Jersey.

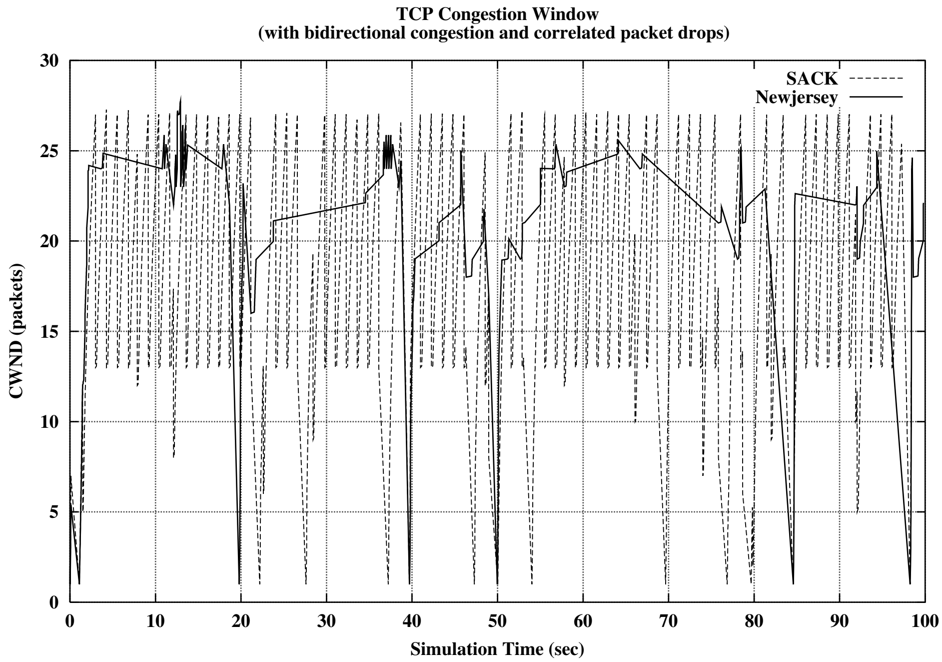

Fig. 12. TCP congestion window development: SACK vs. New Jersey.

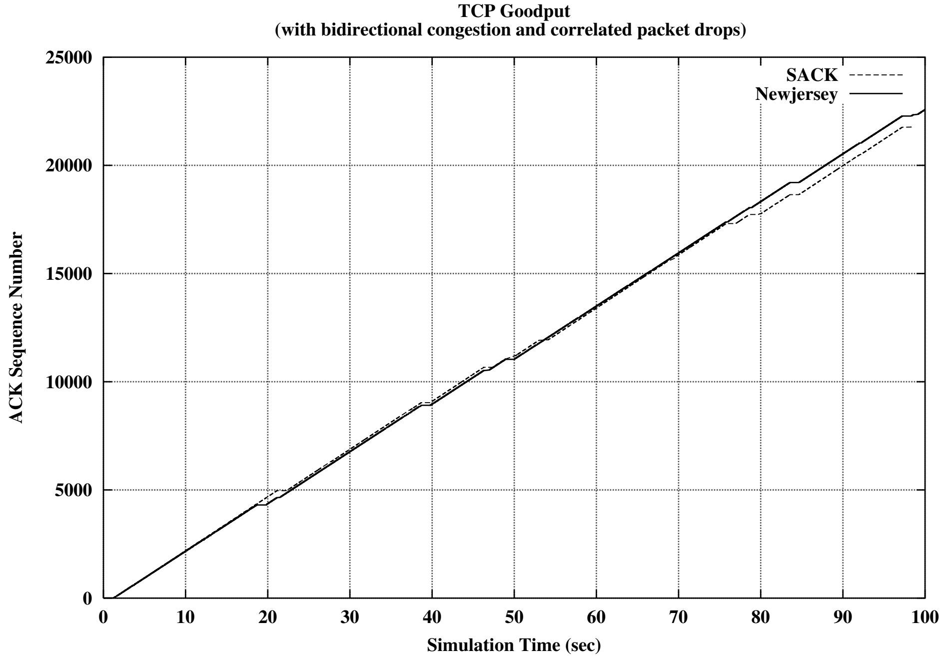

Fig. 13. Goodput comparison: SACK vs. New Jersey.

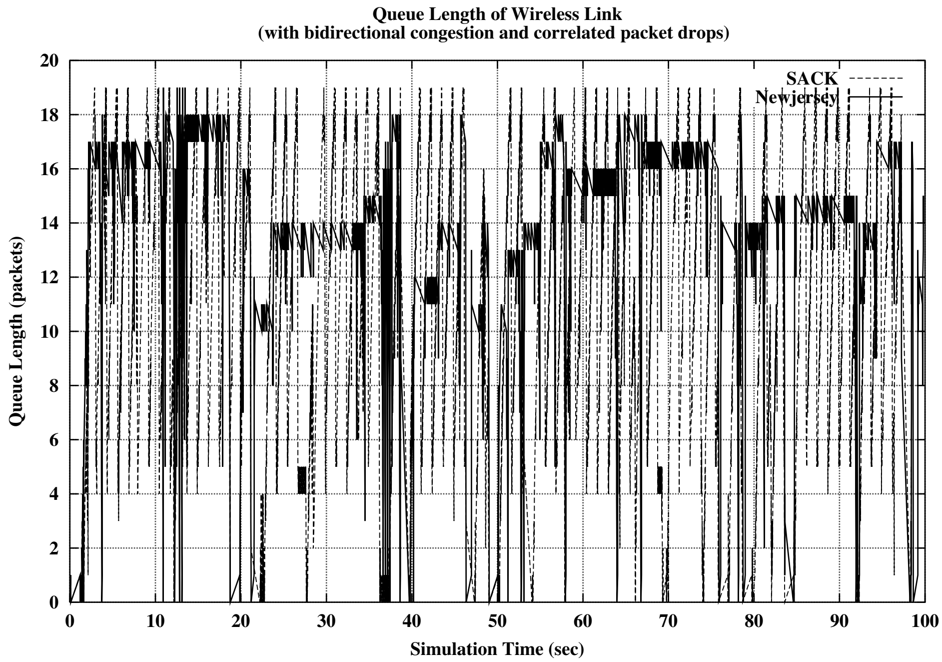

Fig. 14. Comparison of queue length at the bottleneck link: SACK vs. New Jersey.

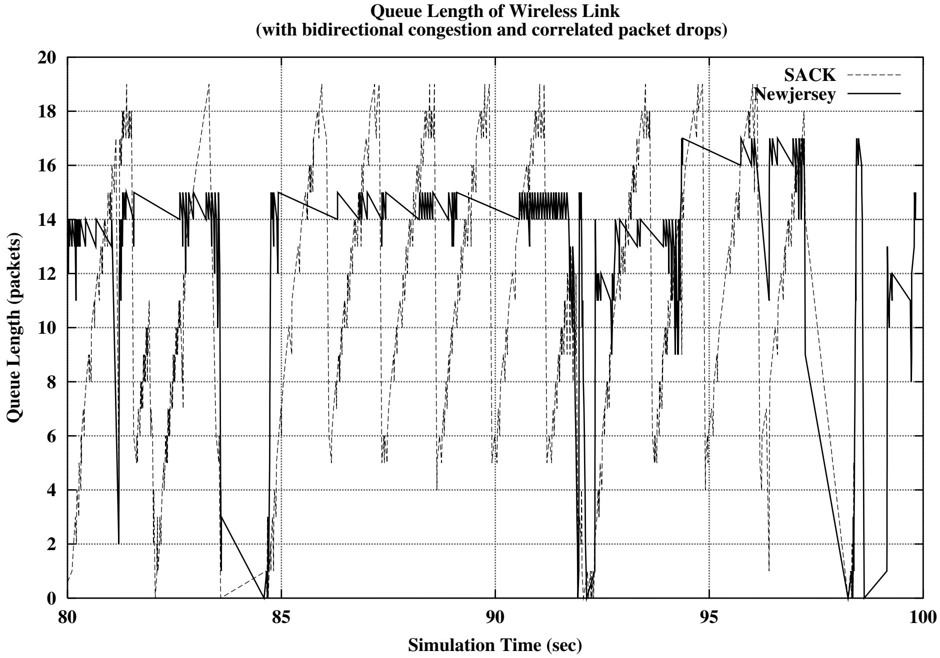

Fig. 15. Zoomed view of comparison of queue length at the bottleneck link: SACK vs. New Jersey.

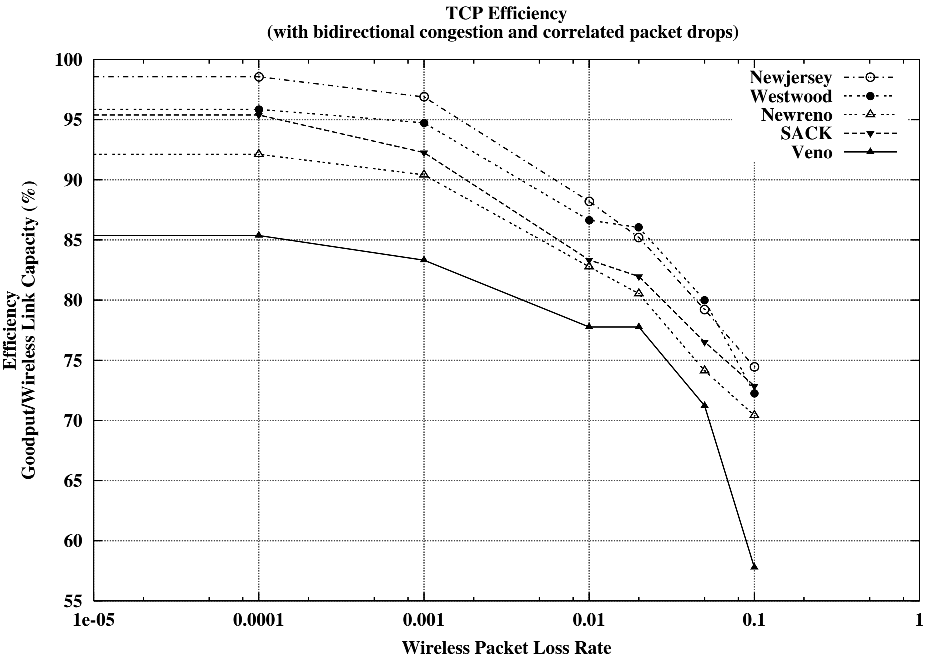

Fig. 16. Comparison of TCP schemes: New Jersey, Westwood, New Reno, SACK, and Veno, where the y-axis represents the link efficiency achieved by each TCP scheme. The link efficiency is defined as the ratio of the achieved goodput and the link capacity of the wireless link.

Table 1
Congestion occurrence comparison

Schemes	Wired link congestion loss (%)	Wireless link congestion loss (%)	Cross traffic congestion loss (%)
New Jersey	0.009	0.003	0.061
New Reno	0.155	0.199	0.048
Westwood	0.155	0.237	0.052
SACK	0.178	0.258	0.045
Veno	0.135	0.152	0.040

jitter, a desirable condition required by many multimedia applications.

A comprehensive comparison of TCP New Jersey vs. other schemes is shown in Fig. 16, where the link efficiency is defined as the ratio of the TCP's goodput and the capacity of the wireless link.

5.2. Comparison of congestion occurrence

One advantage of the TCP New Jersey scheme is its ability to avoiding congestion over other schemes. Given a good rate estimator, the network is expected to have less packet loss due to congestion. We compare the number of congestion occurrences at the links left of the BS (the wired link) and the one right of the BS (the wireless link), shown in Fig. 6, of all schemes. The results are presented in Table 1. New Jersey is almost congestion free on its forward link. We also examine the congestion occurrence of the cross traffic. New Jersey causes no more significant congestion to the cross traffic than other schemes.

6. Remarks and future work

In this paper, we have evaluated the performance characteristics of various TCP schemes under the combined network conditions with asymmetric end-to-end link capacities, correlated wireless errors, and link congestion in both forward and reverse directions. While this combined network condition is more realistic particularly to the end-to-end integrated wired and wireless networks involving 3G cellular technologies, to our knowledge, many analytical and simulation-based studies assumed an ideal congestion-free and error-free reverse channel.

As an improvement on our previous work, TCP New Jersey is presented and compared with other TCP schemes. Our simulation results show that TCP New Jersey is able to accurately estimate the available bandwidth of the bottleneck link of an end-to-end path; the TABE estimator is immune to link asymmetry, bi-directional congestion, and the relative position of the bottleneck link in the multi-hop end-to-end path; the proactive congestion avoidance control mechanism minimizes the network congestion, reduces the network volatility, and stabilizes the queue lengths while achieving superior throughput compared to other TCP schemes. The combination of the simple congestion warning router configuration and accurate estimation of the available network bandwidth using the TCP timestamp option distinguishes TCP New Jersey in differentiating the cause of packet losses.

Compared to other reactive congestion control approaches, TCP New Jersey proactively tunes the slow start threshold to the dynamics of the network without causing unnecessary network congestions. Thus, network devices require smaller I/O buffers.

Our future plans include analytical studies of TCP New Jersey's congestion control mechanism, and its influence and benefits to better active queue management.

Acknowledgment

This work has been supported in part by the New Jersey Commission on Higher Education via the NJI-TOWER project, and the New Jersey Commission on Science and Technology via the NJWINS.

References

[1] J. Postel, Internet protocol, RFC 791, 1981.
[2] J. Postel, Transmission control protocol, RFC 793, 1981.
[3] T.V. Lakshman, U. Madhow, The performance of TCP/IP for networks with high bandwidth-delay products and random loss, IEEE/ACM Transactions on Networking 5 (3) (1997) 336–350.
[4] F. Lefevre, G. Vivier, Understanding TCP's behavior over wireless links, in: Communications and Vehicular Technology 2000, 2000, pp. 123–130.
[5] http://www.cellular-news.com/story/10259.shtml.
[6] K. Xu, Y. Tian, N. Ansari, TCP-Jersey for wireless IP communications, IEEE Journal on Selected Areas in Communications 22 (4) (2004) 747–756.
[7] J.-P. Ebert, A. Willig, A Gilbert–Elliot bit error model and the efficient use in packet level simulation, Technical Report TKN-99-002, Technical University Berlin Telecommunications Network Group, March 1999.
[8] L. Zhang, S. Shenker, D.D. Clark, Observations and dynamics of a congestion control algorithm: the effects of two-way traffic, in: ACM SIGCOMM 1991, 1991, pp. 133–147.
[9] L. Kalampoukas, A. Varma, K.K. Ramakrishnan, Performance of two-way TCP traffic over asymmetric access links, in: Interop'97 Engineers' Conference, 1997.
[10] V. Jacobson, Congestion avoidance and control, in: ACM SIGCOMM 1988, vol. 18, August 1988, pp. 314–329.
[11] V. Tsaoussidis, I. Matta, Open issues on TCP for mobile computing, Wireless Communications and Mobile Computing 2 (1) (2002) 3–20.
[12] W. Stevens, TCP slow start, congestion avoidance, fast retransmit and fast recovery algorithms, RFC 2001, 1997.
[13] A.A. Abouzeid, S. Roy, M. Azizoglu, Comprehensive performance analysis of a TCP session over a wireless fading link with queuing, IEEE Transactions on Wireless Communications 2 (2) (2003) 344–356.
[14] M. Zorzi, A. Chockalingam, R. Rao, Throughput analysis of TCP on channels with memory, IEEE Journal on Selected Areas in Communications 18 (2000) 1289–1300.
[15] S. Floyd, T. Henderson, The New Reno modification to TCP'S fast recovery algorithm, RFC 2582, 1999.
[16] M. Mathis, J. Mahdavi, S. Floyd, A. Romanow, TCP selective acknowledgment options, RFC 2018, 1996.
[17] C.P. Fu, S.C. Liew, TCP Veno: TCP enhancement for transmission over wireless access networks, IEEE Journal on Selected Areas in Communications 21 (2) (2003) 216–228.
[18] C. Casetti, M. Gerla, S. Mascolo, M.Y. Sanadidi, R. Wang, TCP Westwood: bandwidth estimation for enhanced transport over wireless links, in: ACM/IEEE MobiCom 2001, 2001, pp. 287–297.
[19] L.S. Brakmo, L.L. Peterson, TCP Vegas: end to end congestion avoidance on a global Internet, IEEE Journal on Selected Areas in Communications 13 (8) (1995) 1465–1480.
[20] S. Floyd, TCP and explicit congestion notification, ACM Computer Communications Review 24 (5) (1994) 10–23.
[21] T. Bonald, M. May, J.C. Bolot, Analytic evaluation of red performance, in: IEEE INFOCOM 2000, vol. 3, 2000, pp. 1415–1424.
[22] V. Jacobson, R. Braden, D. Borman, TCP extensions for high performance, RFC 1323, 1992.
[23] J.C. Hoe, Improving the start-up behavior of a congestion control scheme for TCP, in: ACM SIGCOMM 1996, 1996, pp. 270–280.
[24] H. Wang, H. Xin, D.S. Reevew, K.G. Shin, A simple refinement of slow-start of TCP congestion control, in: ISCC 2000, 2000, pp. 99–105.
[25] Ucb/lbnl/vint network simulator, available from <http://www.isi.edu/nsnam/ns>.

Kai Xu received the B.S. degree in Electrical Engineering from Zhejiang University, Hangzhou, China in 1989 and M.S. degree in Electrical and Computer Engineering from Chiba University, Chiba, Japan in 1993. He is a Ph.D. candidate in Electrical and Computer Engineering, New Jersey Institute of Technology. His current research interests include Wireless IP, Internet Congestion Control, Active Queue Management, and Sensor Networks.

Ye Tian received the B.E. (Honors) degree in Electrical and Electronic Engineering from University of Canterbury, Christchurch, New Zealand in 1999 and M.S.E.E. degree in Electrical and Computer Engineering from NJIT, Newark, USA in 2002, respectively. He is a Ph.D. candidate in Electrical and Computer Engineering, NJIT. His current research interests include QoS routing, WLAN and IPSec.

Nirwan Ansari received the B.S.E.E. (summa cum laude), M.S.E.E., and Ph.D. from NJIT, University of Michigan and Purdue University in 1982, 1983, and 1988, respectively.

He joined the Department of Electrical and Computer Engineering, NJIT, as an assistant professor in 1988 and has been promoted to a full professor since 1997. He authored with E.S.H. Hou Computational Intelligence for Optimization (1997) and translated into Chinese in 2000, and edited with B. Yuhas Neural Networks in Telecommunications (1994), both published by Kluwer Academic Publishers. He is a technical editor of the IEEE Communications Magazine, the

ETRI Journal, as well as the Journal of Computing and Information Technology. His current research focuses on various aspects of high-speed networks and multimedia communications. He has contributed over 200 publications in journals, conferences, and edited books.

He organized (as the General Chair) the First IEEE International Conference on Information Technology: Research and Education (ITRE2003), was instrumental, while serving as its Chapter Chair, in rejuvenating the North Jersey Chapter of the IEEE Communications Society which received the 1996 Chapter of the Year Award and a 2003 Chapter Achievement Award, served as the Chair of the IEEE North Jersey Section and in the IEEE Region 1 Board of Governors during 2001–2002, and currently serves in various IEEE committees. He was the 1998 recipient of the NJIT Excellence Teaching Award in Graduate Instruction, and a 1999 IEEE Region 1 Award. He is a keynote speaker for the IEEE/ACM co-sponsored ICETE2004 (International Conference on E-Business and Telecommunication Networks).